MONASH University

Enhancing Adversarial Robustness: Representation, Ensemble, and Distribution Approaches

Tuan-Anh Bui

Doctor of Philosophy

A Thesis Submitted for the Degree of Doctor of Philosophy at
Monash University in 2023
Faculty of Information Technology

Copyright notice

Abstract

Deep neural networks, despite their profound capabilities, are vulnerable to adversarial examples, hindering their application in certain fields. Numerous defense methods have been proposed to mitigate those attacks, with adversarial training emerging as the most effective across a spectrum of attacks. This research seeks to improve the adversarial training framework from three important strands of deep learning: representation learning, ensemble learning, and distributional robustness.

Representation learning lies at the heart of deep learning success where features are automatically learned from data. As such, it has an important role in achieving model robustness. To this end, we propose a unique adversarial training framework that possesses local and global compactness in the latent space. Local compactness is achieved by minimizing divergence between an input's latent representation and its adversarial equivalent, while global compactness is achieved by minimizing the divergence between latent representations of samples within a class and maximizing divergence between those of different classes. Building upon this, we introduce an advanced framework using the contrastive learning principle to refine the robust representation. This new method, employing an enhanced metric for measuring divergence between representations, yields superior adversarial robustness compared to its predecessor.

Attacks can be diverse, and ensemble learning offers a principled approach to examine the robustness problem. Here, we introduce a new concept of "transferring flow," designed to mitigate adversarial transferability and encourage model diversity within the ensemble. This process helps strengthening the ensemble's robustness to adversarial examples generated from the ensemble itself, while downplaying the robustness to adversarial examples originating from other models. Further, we introduce a multi-objective optimization framework capable of generating adversarial examples residing within the joint adversarial regions of the ensemble. Robustifying the ensemble with these adversarial examples leads to enhanced robustness surpassing the previous method.

Lastly, moving beyond adversarial training using pair of data samples and theirs adversarial counterparts, we examine the distributional robustness and present a unified framework that encapsulates existing adversarial training methods under our proposed Wasserstein distributional robustness framework. We introduce a new cost function involving the Wasserstein distance between empirical and worst-case distributions. This approach generalizes existing adversarial training methods and proposes a novel technique, outperforming prior methods in terms of robustness.

This thesis offers novel adversarial training frameworks to improve the robustness of deep neural networks while providing a deeper understanding of adversarial vulnerability

within the contexts of representation learning, ensemble learning, and distributional robustness. This enhanced understanding of adversarial vulnerability paves the way for the development of increasingly robust machine learning models in the future.

Thesis including published works declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

This thesis includes four original papers published in peer reviewed conferences and journals and one submitted publication. The core theme of the thesis is developing robust machine learning models against adversarial examples from three different perspectives including representation learning, ensemble learning, and distributional robustness. The ideas, development and writing up of all the papers in the thesis were the principal responsibility of myself, the student, working within the Faculty of Information Technology under the supervision of Dinh Phung and Trung Le.

The inclusion of co-authors reflects the fact that the work came from active collaboration between researchers and acknowledges input into team-based research.

In the case of Chapter 3-5 my contribution to the work involved the following:

Thesis Chapter	**Publication Title**	**Status**	**Nature and % of student contribution**	**Co-author name(s) Nature and % of co-author's contribution**	**Co-author(s) Monash student Y/N***
3	Improving Adversarial Robustness by Enforcing Local and Global Compactness	Accepted	Proposing research idea, implementing models, conducting experiments, and writing paper. 60%	1. Trung Le. Discussing idea and writing paper. 15%. 2. He Zhao. Discussing idea and writing paper. 5%. 3. Paul Montague. Discussing idea and writing paper. 5%. 4. Olivier deVel. Discussing idea and writing paper. 5%. 5. Tamas Abraham. Discussing idea and writing paper. 5%. 6. Dinh Phung. Discussing idea and writing paper. 5%.	N
3	Understanding and Achieving Efficient Robustness with Adversarial Supervised Contrastive Learning	Submitted	Proposing research idea, implementing models, conducting experiments, and writing paper. 65%	1. Trung Le. Discussing idea and writing paper. 15%. 2. He Zhao. Discussing idea and writing paper. 5%. 3. Paul Montague. Discussing idea and writing paper. 5%. 4. Seyit Camtepe. Discussing idea and writing paper. 5%. 5. Dinh Phung. Discussing idea and writing paper. 5%.	N

4	Improving Ensemble Robustness by Collaboratively Promoting and Demoting Adversarial Robustness	Accepted	Proposing research idea, implementing models, conducting experiments, and writing paper. 60%	1. Trung Le. Discussing idea and writing paper. 15%. 2. He Zhao. Discussing idea and writing paper. 5%. 3. Paul Montague. Discussing idea and writing paper. 5%. 4. Olivier deVel. Discussing idea and writing paper. 5%. 5. Tamas Abraham. Discussing idea and writing paper. 5%. 6. Dinh Phung. Discussing idea and writing paper. 5%.	N
4	Generating Adversarial Examples with Task Oriented Multi-Objective Optimization	Accepted	Proposing research idea, implementing models, conducting experiments, and writing paper. 65%	1. Trung Le. Discussing idea and writing paper. 15%. 2. He Zhao. Discussing idea and writing paper. 5%. 3. Quan Tran. Discussing idea and writing paper. 5%. 4. Paul Montague. Discussing idea and writing paper. 5%. 5. Dinh Phung. Discussing idea and writing paper. 5%.	N
5	A Unified Wasserstein Distributional Robustness Framework for Adversarial Training	Accepted	Proposing research idea, implementing models, conducting experiments, and writing paper. 60%	1. Trung Le. Discussing idea and writing paper. 25%. 2. Quan Tran. Discussing idea and writing paper. 5%. 3. He Zhao. Discussing idea and writing paper. 5%. 4. Dinh Phung. Discussing idea and writing paper. 5%.	N

I have renumbered sections of submitted or published papers in order to generate a consistent presentation within the thesis.

Student name: Tuan-Anh Bui

Student signature: **Date: 31/05/2023**

I hereby certify that the above declaration correctly reflects the nature and extent of the student's and co-authors' contributions to this work. In instances where I am not the responsible author I have consulted with the responsible author to agree on the respective contributions of the authors.

Main Supervisor name: Dinh Phung

Main Supervisor signature: **Date: 31/05/2023**

Publications during enrolment

The content of this thesis is based on the following publications:

1. **Anh Bui**, Trung Le, He Zhao, Paul Montague, Olivier de Vel, Tamas Abraham, Dinh Phung, *"Improving Adversarial Robustness by Enforcing Local and Global Compactness"*. In Proceedings of the European Conference on Computer Vision (ECCV) 2020.

2. **Anh Bui**, Trung Le, He Zhao, Paul Montague, S. Camtepe, Dinh Phung, *"Understanding and Achieving Efficient Robustness with Adversarial Supervised Contrastive Learning"*. Submitted to Neurocomputing, 2023.

3. **Anh Bui**, Trung Le, Quan Tran, He Zhao, Dinh Phung, *"A Unified Wasserstein Distributional Robustness Framework for Adversarial Training"*. In Proceedings of the International Conference on Learning Representation (ICLR), 2022.

4. **Anh Bui**, Trung Le, He Zhao, Paul Montague, Olivier de Vel, Tamas Abraham, Dinh Phung, *"Improving Ensemble Robustness by Collaboratively Promoting and Demoting Adversarial Robustness"*. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) 2021.

5. **Anh Bui**, Trung Le, He Zhao, Quan Tran, Paul Montague, Dinh Phung, *"Generating Adversarial Examples with Task Oriented Multi-Objective Optimization"*. Accepted to the Transactions on Machine Learning Research (TMLR), 2023.

Further contributions were made during the course of this research, but are not included in this thesis:

1. **Anh Bui***, Vy Vo*, Tung Pham, He Zhao, Dinh Phung, Trung Le, *"Diverse-Aware Agnostic Ensemble of Sharpness Minimizers"*. Preprint, 2023.

2. Trung Le, **Anh Bui**, Tue Le, He Zhao, Quan Tran, Paul Montague, Dinh Phung, *"On Global-view Based Defense via Adversarial Attack and Defense Risk Guaranteed Bounds"*. In Proceedings of The 25th International Conference on Artificial Intelligence and Statistics (AISTATS), 2022.

3. Hoang Phan, Trung Le, Trung Phung, **Anh Bui**, Nhat Ho, Dinh Phung, *"Global-Local Regularization Via Distributional Robustness"*. In Proceedings of The 25th International Conference on Artificial Intelligence and Statistics (AISTATS), 2022.

4. Ngoc Tran, **Anh Bui**, Dinh Phung, Trung Le, *"Multiple Perturbation Attack: Attack Pixelwise Under Mixed lp-norms For Better Adversarial Performance"*. arXiv preprint arXiv:2212.03069, 2022.

5. Ngoc Tran, Lam Tran, **Anh Bui**, Tung Pham, Toan Tran, Dinh Phung, Trung Le, *"Robust Contrastive Learning with Theory Guarantee".* Preprint, 2023.

6. Van-Anh Nguyen, Trung Le, **Anh Bui**, Toan Do, Dinh Phung, *"Optimal Transport Model Distributional Robustness".* Preprint, 2023.

Acknowledgements

This thesis represents the culmination of a three-and-a-half-year journey filled with a multitude of emotions and invaluable memories. I embarked on this Ph.D. adventure with excitement, eager to delve into research and embrace a new life with my family in the beautiful city of Melbourne. However, the onset of the Covid-19 pandemic presented itself as a formidable adversary, continuously challenging me from multiple angles.

During certain periods, I found myself isolated and overwhelmed, experiencing feelings of depression and guilt due to the separation from my son Minh-Khoi, who was stuck in Vietnam. In light of these challenges, I am immensely grateful to the remarkable individuals mentioned below, who served as my defenders and provided enormous support, enabling me to overcome the pandemic and complete this thesis.

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Dinh Phung and Dr. Trung Le. Their patient guidance and dedicated involvement at every step of the process were instrumental in the accomplishment of this thesis. Dinh introduced me to the world of research and provided me with endless support. Whenever I encountered obstacles, he selflessly stepped in to assist me. Our memories together, whether it be swimming, playing the guitar, shopping, or dining, will forever hold a special place in my heart. Dinh is not just a supervisor to me; he is a friend, a cherished member of my extended family. The friendship between our families, encompassing Dinh, Thi, Dylan, Boddie, and my own, has been a source of great joy and support.

I would also like to extend my gratitude to Trung, my co-supervisor, whose deep knowledge, passion, and enthusiasm served as an inspiration throughout my journey. Trung stood by my side during countless long nights, engaging in discussions and helping me solve problems. He, too, became a dear friend, introducing me to the world of kung fu and tai chi, and warmly welcoming me into his family, where we shared delightful moments over fun-filled and delectable meals.

Ethan deserves special mention as an unofficial guide whose contributions have been crucial in my research journey. His logical thinking and exceptional writing skills were a source of inspiration. I'm grateful for his support whenever I found myself stuck.

I extend my sincere thanks to my esteemed panel members, Prof. Jianfei Cai, Dr. Shirui Pan, Assoc. Prof. Arun Konagurthu, Dr. Abhinav Dhall, and Dr. Teresa Wang, for their valuable comments and advice, which greatly contributed to the improvement of this thesis. I would also like to thank the examiners, Dr. Ehsan Abbasnejad and A/Prof. Xingjun Ma, for their insightful comments and suggestions, from which this thesis has benefited greatly.

I am deeply grateful for the opportunity to collaborate with Paul Montague, Tamas Abraham, Olivier de Vel, Sayit Camtepe, and Quan Tran. Working alongside them has been a rewarding experience, and I appreciate their contributions.

I must express my gratitude to my ex-supervisors at SUTD, prof. Ngai-Man Cheung and Dr. Trung Tran, who gave me the opportunity to change my career path from electrical engineering to machine learning. Tu Nguyen's introduction to Prof. Dinh and the lab was also a defining moment.

I'm thankful to my friends at Monash University including Dai, Nhan, Quan, Viet, Binh, Thanh, Tuan Nguyen, Son, Van, Trang, Van-Anh, Vy, Long, Cuong for all the fun times we shared. I would like to express my deep appreciation to my close friends Van, Hai, Thanh Tong, Cong, Tuan, Dzung, Dat, Doanh, Quyen, Tien-Cuong, Cuong Tran, Giang, Thanh Nguyen, and Thu-Dzung, despite the vast distance of thousands of kilometers between us. Your friendship has been a constant source of strength and encouragement throughout this journey. I am also grateful to Trang Diep family, who supported me from the very beginning.

Lastly, I am profoundly grateful to my family for their consistent support and boundless patience throughout my entire life. My parents Tuan, Nhuan, and my sister Giang have been a constant source of unconditional love and encouragement. I owe them my very existence, and I will forever cherish their presence in my life. I would like to extend a special appreciation to my family-in-law Thuy, Viet-Anh, Lieu, and Bon, particularly my mother-in-law Lieu, who selflessly cares for my wife and sons during my absence and even traveled to Australia to support us. Their dedication and willingness to go the extra mile have been truly remarkable. My sons, Minh-Khoi and Minh-Dang, are the driving force behind my energy, motivation, and ultimate happiness. Their presence in my life has been an endless source of inspiration and joy.

And the most important, my beloved wife Thuy holds an immeasurable place in my heart and life. She is not only my best friend, but also my soulmate and lifelong partner. Throughout this challenging journey, she has been an unwavering pillar of strength, standing by my side, providing unyielding support, and constantly encouraging me to overcome any challenges that crossed our path. Embracing change and sacrifice, she relocated to a new country and stood by me through lockdowns, even bringing our second son into this world in the middle of the pandemic. She has been a steadfast companion in every step. It is to her, the anchor in my life, that I dedicate this thesis.

Contents

List of Figures

Abbreviations

AI	**A**rtificial **I**ntelligence
AT	**A**dversarial **T**raining
AML	**A**dversarial **M**achine **L**earning
CL	**C**ontrastive **L**earning
CW	**C**arlini-**W**agner
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etwork
DR	**D**istributional **R**obustness
EMD	**E**arth **M**over's **D**istance
FGSM	**F**ast **G**radient **S**ign **M**ethod
GAN	**G**enerative **A**dversarial **N**etwork
KL	**K**ullback-**L**eibler
MGDA	**M**ulti-**G**radient **D**escent **A**lgorithm
ML	**M**achine **L**earning
MOO	**M**ulti-**O**bjective **O**ptimization
PGD	**P**rojected **G**radient **D**escent
SCL	**S**upervised **C**ontrastive **L**earning
SSL	**S**elf-**S**upervised **L**earning
SSCL	**S**elf-**S**upervised **C**ontrastive **L**earning

Chapter 1

Introduction

1.1 From Adversarial Examples to Adversarial Training

1.1.1 The Rise of Adversarial Examples

Deep Neural Networks (DNNs) have shown remarkable achievements across a variety of disciplines (Goodfellow et al., 2016), including computer vision (He et al., 2016, Zagoruyko and Komodakis, 2016), natural language processing, and speech processing (Vaswani et al., 2017). The evolution of model architectures, simultaneously with the rapid growth of computational resources, has enabled DNNs to achieve or even surpass human-level accuracy in several tasks (Hesamian et al., 2019, Minaee et al., 2020), such as image classification (He et al., 2016) or natural language processing tasks (Ouyang et al., 2022).

Due to their superior performance, DNNs have found extensive real-world applications, from powering language translation in Google Translate and user recommendations on Amazon to facilitating autonomous driving technology. However, despite their success, DNNs, including state-of-the-art models, are known to be vulnerable to adversarial perturbations. These perturbations, while imperceptible to the human eye, can lead to incorrect or unexpected outcomes (Szegedy et al., 2014, Goodfellow et al., 2015).

Figure 1.1 provides an illustrative example of such adversarial examples, attacking the pre-trained ResNet50 model (He et al., 2016). The image on the left correctly predicted as a "koala" with 49.8% confidence, is compared to the identical-looking adversarial

image on the right. The latter has been subtly manipulated to induce the model to misclassify it as a "balloon" with 100% confidence.

Adversarial examples have been demonstrated to exist across a wide range of DNN applications (Akhtar and Mian, 2018), encompassing image classification (Goodfellow et al., 2015, Madry et al., 2018), image segmentation (Xie et al., 2017), graph-structured data (Dai et al., 2018), and speech-to-text systems (Carlini and Wagner, 2018). For example, Wu et al. (2020a), Song et al. (2018) demonstrated that adversarial perturbations can be printed on physical objects such as clothes or street signs causing Object Detectors such as YOLOv2 (Redmon and Farhadi, 2017) to misclassify them. This inherent vulnerability and instability of DNNs could pose serious risks to their real-world applications. Consequently, there is an urgent need to develop DNN models that are robust against various types of adversarial examples, thereby enhancing their security and reliability in real-world deployments.

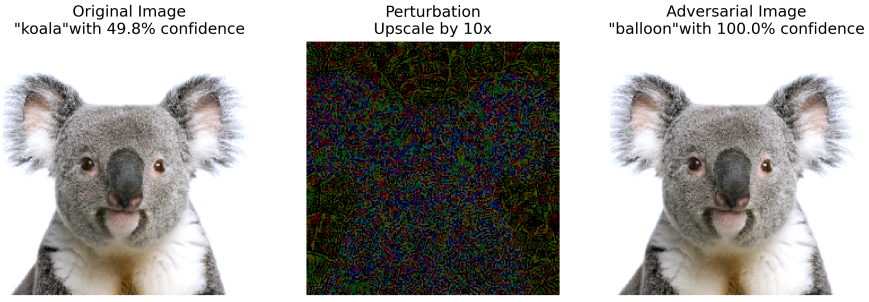


FIGURE 1.1: An example of an adversarial perturbation attack. The image on the left is the original image which was predicted as "koala" with 49.8% confidence, and the image on the right is the adversarial image, which has been modified to cause the model to misclassify it as "ballon" with 100% confidence. The middle image shows the perturbation that was added to the original image with 10x magnification. The code to generate this example is available at `https://github.com/tuananhbui89/demo_attack`.

1.1.2 Efforts to Tackle Adversarial Examples

Since proposed in Szegedy et al. (2014), adversarial examples have been the subject of extensive research in recent years, with the number of papers published on the topic increasing exponentially, as shown in Figure 1.2. On the one hand, various attack methods have been proposed to enhance effectiveness (Madry et al., 2018, Kurakin et al., 2016, Dong et al., 2018, Carlini and Wagner, 2017), computational efficiency (Zhang

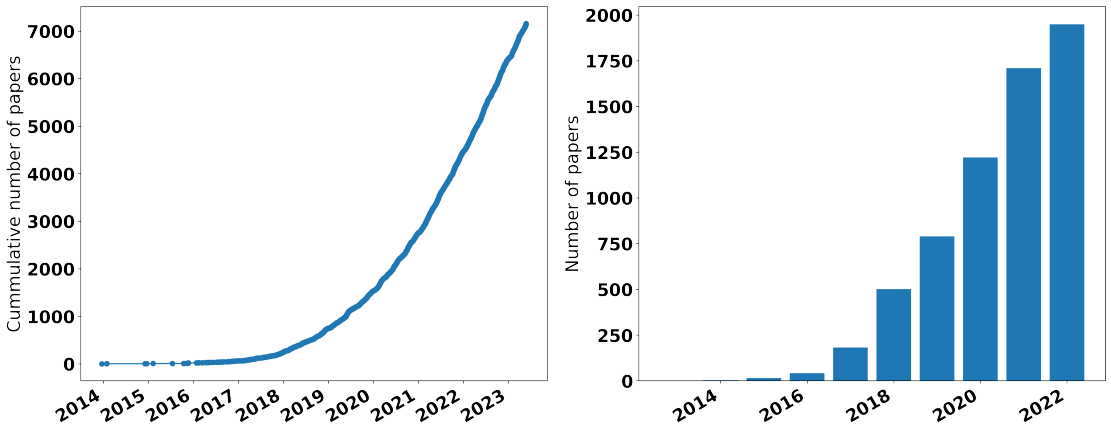


FIGURE 1.2: Number of adversarial examples papers published on arXiv from 2014 to May-2023. Data source from: `https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html`

et al., 2022, Wong et al., 2019a), transferability among inputs (Moosavi-Dezfooli et al., 2017) or among models (Papernot et al., 2016a). On the other hand, there is also an extremely large number of defense methods proposed to mitigate adversarial attacks (Akhtar et al., 2021, Bai et al., 2021), including advances in the architecture perspective such as ensemble (Tramèr et al., 2018), distillation (Papernot et al., 2016b), quantization (Gui et al., 2019), pruning (Dhillon et al., 2018), smooth activation functions (Xie et al., 2020). Pre-processing and post-processing approaches, such as transformations (Dziugaite et al., 2016, Guo et al., 2017) and detection methods (Metzen et al., 2017), have also been explored. Despite numerous defense strategies being proposed to counter adversarial attacks (Akhtar and Mian, 2018, Akhtar et al., 2021), no method has yet provided comprehensive protection or completely illuminated the vulnerabilities of DNNs. For example, many defense methods have been broken with carefully crafted attacks or with the invention of new attacks (Madry et al., 2018, Croce and Hein, 2019, 2020).

The failure of many defense methods can be attributed to several factors. Firstly, compared to classical machine learning methods such as decision trees or support vector machines, deep learning methods are more complex and harder to trace the decision-making process. DNNs, with their multi-layered, interconnected neuron networks, are not only more complex but also opaque. While these models can achieve high performance on a wide range of tasks, they are often difficult to interpret, as the relationships between the input features and the output predictions are highly nonlinear and not easily captured by a simple graphical representation.

Secondly, the difficulty in interpreting the decision-making process also makes it harder to understand and evaluate the vulnerability of DNNs (Carlini et al., 2019). For example, many defense methods have been proven of giving a false sense of security because of the obfuscated gradient (Athalye et al., 2018) in which the gradient of these models is hardly computed correctly. This phenomenon can be implanted by using non-differentiable operations such as rounding/quantization or stochastic operations or by using specific operations which cause vanishing or exploding gradients. These defenses first seem to be robust against gradient-based attacks but later be shown to be failed against black-box/transferred attacks or carefully fine-tune white-box attacks (Athalye et al., 2018). Therefore, careful evaluation and systematic vetting of defense methodologies are critical in ensuring robustness against adversarial examples.

1.1.3 Adversarial Training: Importance and Challenges

In light of the aforementioned issues, Adversarial Training (AT) stands out as the most resilient defense method against adversarial examples (Szegedy et al., 2014, Madry et al., 2018, Athalye et al., 2018). The premise of AT is straightforward: it involves the generation of adversarial examples, which are then integrated into the training set (Szegedy et al., 2014). The adversarial examples are typically generated using gradient-based attacks such as the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) or Projected Gradient Descent (PGD) (Madry et al., 2018). Despite its simplicity, AT has been demonstrated to be robust against a multitude of attacks and remains one of the most reliable defense methods (Madry et al., 2018, Athalye et al., 2018).

However, the practical deployment of AT comes with several challenges that limit its feasibility in real-world applications. Firstly, AT is computationally intensive, given that it necessitates the generation of adversarial examples for each training sample, repeatedly over multiple epochs, each of which requires multiple forward and backward processes (Madry et al., 2018). Moreover, AT's susceptibility to multi-step attacks when employed with a single-step attack has been documented (Tramèr et al., 2018). This susceptibility was later attributed to a phenomenon known as catastrophic overfitting (Rice et al., 2020). Another significant drawback of AT is the tradeoff between the robustness and accuracy of a model. Models that demonstrate high robustness following

AT often suffer reduced accuracy (Zhang et al., 2019). Consequently, efforts to refine and optimize AT for real-world applications are necessary to mitigate these challenges.

To this end, recent research has explored various avenues to enhance AT. One direction involves the integration of AT with techniques like sharpness-aware minimization (Foret et al., 2021), which promotes smoother behavior in both the input and model spaces. This integration has shown promise in striking a better balance between model robustness and accuracy (Wu et al., 2020b, Nguyen et al., 2023). Another active direction involves augmenting AT with data augmentation methods, such as diffusion models (Ho et al., 2020, Song et al., 2020), which increase the diversity of the training data. This augmentation has demonstrated significant improvements in both robustness and accuracy (Gowal et al., 2021, Rebuffi et al., 2021, Wang et al., 2023). Moreover, there are ongoing efforts to improve the efficiency of AT, including approximating multi-step attacks with single-step counterparts (Zhang et al., 2022, Wong et al., 2019a, Andriushchenko and Flammarion, 2020). These evolving approaches aim to advance AT's capabilities and address its practical limitations for real-world deployment.

1.2 Aims and Contributions

In this section, we delve into the contributions made in this thesis concerning the enhancement of adversarial robustness. Our focus lies on three distinct perspectives: representation learning, ensemble learning, and distributional robustness.

1.2.1 Representation Learning Approaches

Why Representation Learning? Representation learning constitutes a fundamental pillar of machine learning and deep learning (Bengio et al., 2013). It primarily concerns the extraction of efficient and meaningful representations of data, with these learned representations deployed for various downstream tasks, such as classification, regression, or clustering.

As discussed in Bengio et al. (2013), successful representation learning can reveal general priors about the observed data, crucial for the development of effective AI models.

These priors, while not task-specific, include characteristics such as smoothness, locality, and abstraction, which are typically beneficial for learning AI models. Numerous approaches have attempted to integrate these priors into the representation learning process, including methods like autoencoders (Kramer, 1991, Kingma and Welling, 2013) and self-supervised learning (Yarowsky, 1995, Balestriero et al., 2023).

However, in the context of adversarial machine learning, these representation learning methods are not sufficient to achieve the robustness property. That leads to the first central question that this thesis tries to answer: what are characteristics of a representation that benefit robustness?

Several early works attempt to improve adversarial robustness through the lend of representation learning, however, they are either limited intuition or not effective. The work Ilyas et al. (2019) made a hypothesis that adversarial vulnerability is the result of the sensitivity of deep learning models to well-generalizable but imperceptible-to-human features. This phenomenon was later analyzed as the phenomenon of learning a shortcut in deep models (Geirhos et al., 2020). Based on this hypothesis, the authors proposed an ideal framework that learns from useful and robust features only that can achieve both robustness and generalization. However, to achieve this goal, the authors proposed a method to disentangle the robust/non-robust features relying on a pre-trained robust model which is limited by the robustness of the pre-trained model.

Samangouei et al. (2018) proposed a GAN-based method to model the data manifold and then used the learned generator to approximate the input sample. In this way, the gradient-based attacks could not find the adversarial examples because of the non-differentiability of the generator. However, later Jalal et al. (2017) proposed an overpowered attack method to efficiently attack these kinds of non-differentiable-based defenses.

Stutz et al. (2019) found that the regular adversarial examples leave the manifold of benign data, which explains the drop of generalization when using these adversarial examples in adversarial training. The authors proposed a VAE-GAN architecture to approximate the data manifold and based on that, they proposed an on-manifold adversarial generation by using a pre-trained encoder-decoder. More specifically, they perturbed the latent representation of the benign input getting by the encoder and then

used the decoder to generate the adversarial example. However, this method solely relies on the assumption that the data manifold is well approximated by a VAE-GAN architecture which is not feasible in real-world applications.

These above lines of work can be classified into a category of direct robust representation learning methods where they tried to directly model the robust representation of the data. On the other hand, there are also other works that indirectly learn robust representations through additional regularizations that introduce inductive bias properties in the latent space. These kinds of methods are empirically proven to be more effective. For example, the work (Mao et al., 2019) was the pioneer that found the shift of representations of adversarial examples to a cluster of a false class. To mitigate this issue, they proposed to minimize the distance between the representations of adversarial examples and the representations of benign examples formulated by a triplet loss.

Contributions. In Chapter 3, we delve into our contributions, as outlined in two papers: Bui et al. (2020) and Bui et al. (2021a), which aim to address the aforementioned research question. These works revolve around harnessing both local information, such as adversarial/benign identification, and global information, such as class identity, to acquire a robust representation. Our ultimate goal is to develop an ideal feature extractor capable of exhibiting invariance to adversarial perturbations, leading to similar representations for both adversarial and benign examples. At the same time, this feature extractor aims to maintain discriminative power for class identity, resulting in distinct representations for different classes. By achieving such a robust feature extractor, we can establish a solid foundation for constructing a robust classifier, even with the use of a simple linear classifier on top of it.

The work **(Bui et al., 2020)** marks the initial stage in this series, where we introduce a regularization method that imposes local and global compactness properties on the latent space. In particular, we enforce the local compactness property by minimizing the divergence between the representations of adversarial and benign examples. Additionally, we propose minimizing the divergence between representations of samples from the same class while maximizing the divergence between representations from different classes to enforce the global compactness property. Through comprehensive experiments, we showcase that our proposed method enhances the model's robustness while preserving its generalization ability.

The work **(Bui et al., 2021a)** is a follow-up to the previous work, where we further investigate the impact of the distance metric used to measure the divergence between representations. While the previous work was the pioner work that proposed to leverage both local and global information to learn robust representations, the distance metric used to measure the divergence between representations is the Euclidean distance, which is not suitable for high-dimensional data. At the same time, contrastive learning (Chen et al., 2020a) gained popularity in the computer vision community as an effective self-supervised representation learning method. At its core, contrastive learning employs a contrastive loss that encourages the proximity of representations between an anchor example and its positive examples, while promoting distance between the anchor example and its negative examples. Although the principle aligns with our previous work, the contrastive loss considers the relative distance between representations in the latent space, providing a better interpretation of global information compared to the absolute distance in our earlier approach.

In our work (Bui et al., 2021a), we introduce a novel regularization method that utilizes the contrastive loss to enforce local and global compactness properties in the latent space. However, it is not a straightforward application of the contrastive loss, as we demonstrate that applying it directly yielded ineffective results. To gain a deeper understanding, we address three research questions in Bui et al. (2021a): why contrastive learning (CL) can improve adversarial robustness, how to integrate CL with adversarial training in the context of AML, and the key factors influencing CL's performance in AML.

To tackle these questions, we propose the Adversarial Supervised Contrastive Learning (ASCL) framework, which combines adversarial training and contrastive learning to learn robust representations. Furthermore, we develop a novel set of strategies for selecting positive and negative samples, carefully choosing the most relevant samples for the anchor to enhance adversarial robustness. Through empirical evaluation, we demonstrate that our ASCL framework significantly outperforms several adversarial training methods, including our previous work (Bui et al., 2020), and achieves comparable performance to state-of-the-art robust defenses in the literature.

1.2.2 Ensemble Learning Approaches

Why Ensemble Learning? While there are many adversarial training variants have been developed, however, most methods typically address the robustness within a single model (e.g., Madry et al. (2018), Papernot et al. (2016c), Moosavi-Dezfooli et al. (2016), Qin et al. (2019), Shafahi et al. (2019)). To cater for more diverse types of attacks, recent work, notably Tramèr et al. (2018), He et al. (2017), Strauss et al. (2017), Liu et al. (2018a), Pang et al. (2019), have shown that ensemble learning is a promising and potential direction to strengthen adversarial robustness of single models for obtaining stronger models.

In essence, an ensemble model's decision is an amalgamation of predictions made by its members. As a result, an attacker would need to deceive the majority, if not all, of the ensemble members to successfully fool the entire model. This poses a greater challenge for attackers. Moreover, the greater the diversity among ensemble members, the more difficult it becomes for attackers to launch successful attacks.

Conventional ensemble learning methods such as bagging (Breiman, 1996), boosting (Freund and Schapire, 1997), Bayesian averaging (Hoeting et al., 1999) have been proven that can significantly increase the diversity among ensemble members, subsequently, improve natural accuracy. Therefore, naturally, these conventional ensemble methods are expected to improve adversarial robustness.

However, recent work showed that a naive ensemble of weaker defenses is not a stronger classifier as expected (He et al., 2017). An adaptive attack which has full access to the ensemble model and adapts its attack strategy accordingly can still easily find the adversarial examples that can fool the ensemble model. It is also observed that adversarial examples crafted from one model can be transferred to fool other models (Papernot et al., 2016a,b). Therefore, key principles for ensemble-based adversarial training largely remain open. In particular, there are two research questions that we aim to answer in this thesis: "What are factors affecting the ensemble robustness?" and "What principles can be used to collaborate single models to a more robust ensemble model?".

Several ensemble-based adversarial training methods have been proposed in the literature, including notable contributions from Tramèr et al. (2018), Strauss et al. (2017), Liu

et al. (2018a), and Pang et al. (2019). Although these methods have shown initial success, the key principles to achieve ensemble robustness have yet to be fully understood. One critical challenge is to minimize the "transferability" among ensemble members in order to enhance the overall robustness of the ensemble model (Papernot et al., 2016a, Tramèr et al., 2018, Pang et al., 2019, Liu et al., 2016, Kariyappa and Qureshi, 2019).

In Kariyappa and Qureshi (2019), robustness was achieved by aligning the gradients of the ensemble members to be diametrically opposed, thereby reducing the shared adversarial spaces or transferability. However, this method was specifically designed for black-box attacks and remained vulnerable to white-box attacks. Moreover, attempting to achieve gradient alignment becomes unreliable when dealing with high-dimensional datasets and poses challenges when extending it to ensembles with more than two members.

More recently, Pang et al. (2019) proposed a method to promote the diversity of non-maximal predictions among ensemble members (i.e., the diversity among softmax probabilities excluding the highest ones) to mitigate adversarial transferability. However, the central concept of transferability has yet to be systematically addressed in a comprehensive manner.

Contributions. In Chapter 4, we present our contributions on improving adversarial robustness through ensemble learning perspective as outlined in two papers: Bui et al. (2021b) and Bui et al. (2023).

In the work **Bui et al. (2021b)**, we first make the concept of adversarial transferability concrete via the definitions of secure and insecure sets. Our goal is to reduce adversarial transferability and increase model diversity by minimizing the overlap between the insecure sets of committee models, thus reducing the similarity of their regions affected by adversarial examples.

However, we observe that solely reducing adversarial transferability is not enough to guarantee accurate predictions from the ensemble model. This is because a committee member that consistently made incorrect predictions could dominate the final decisions. To address this issue, we introduce a concept called "transferring flow" by combining robustness-promoting and demoting operations.

The key principle behind coordinating these operations is to promote the predictions of one model on a given adversarial example while demoting the predictions of another model on the same example. This approach aims to minimize the negative impact of incorrect predictions and ensure the correct predictions of the ensemble model.

Unlike previous works (Strauss et al., 2017, Pang et al., 2019, Kariyappa and Qureshi, 2019) that only focused on adversarial examples specifically crafted for the ensemble model, we expose the committee members in our ensemble to various divergent sets of adversarial examples. This exposure motivates the committee members to become increasingly diverse over time.

Through extensive experiments, we demonstrate that our proposed method significantly outperforms previous ensemble-based adversarial training methods. Interestingly, by strengthening the demoting operations, our method also facilitates better detection of adversarial examples. This capability to identify adversarial examples further enhances the overall robustness of the ensemble model.

In previous studies, generating adversarial examples that deceive all ensemble members has proven crucial for enhancing ensemble model robustness. Expanding on this, our work **(Bui et al., 2023)** introduces a novel method for generating transferable adversarial examples within the joint insecure region shared by all ensemble members. We formulate the adversarial generation task as a multi-objective optimization problem, aiming for Pareto optimality by maximizing multiple objectives simultaneously. However, directly applying multi-objective optimization proved unsatisfactory due to task dominance.

To address this, we propose the Task Oriented Multi-Objective Optimization (TA-MOO) framework, which prioritizes unsuccessful tasks while maintaining success using a novel geometry-based regularization term. Extensive experiments across three adversarial generation tasks and one adversarial training task showcase the effectiveness of our approach in generating stronger and more resilient adversarial examples. Combining our method with adversarial training further strengthens model security.

1.2.3 Distributional Robustness Approaches

Why Distributional Robustness? In the previous two perspectives, we have discussed about the importance of representation learning and ensemble learning on improving adversarial robustness. However, these methods, along with other AT-based methods, seek a pointwise adversary by independently perturbing each data sample.

Considering adversarial effects at a distributional level, on the other hand, may offer unexplored benefits. Unlike AT, distributional robustness seeks a worst-case distribution that generates adversarial examples from a known uncertainty set of distributions located in the ball centered around the data distribution. This approach is expected to have better generalization performance on unseen data.

Conceptually and theoretically, distributional robustness can be viewed as a generalization and better alternative to AT. Several attempts (Staib and Jegelka, 2017, Sinha et al., 2017) have shed light on connecting AT with DR. However, to the best of our knowledge, practical DR approaches that achieve comparable performance with state-of-the-art AT methods have yet to be developed.

Contributions. In Chapter 5, we present our contributions towards improving adversarial robustness through the lens of distributional robustness, as introduced in **Bui et al. (2022)**.

In particular, we propose a unified framework that connects Wasserstein distributional robustness with current state-of-the-art AT methods. We introduce a new cost function of the Wasserstein distance and propose a unified formulation of the risk function in WDR, with which, we can generalize and encompass the existing AT methods including SOTA ones in the distribution robustness setting.

Through extensive experiments, we demonstrate that with better generalization capacity of distributional robustness, the resulted AT methods in our framework can achieve better adversarial robustness than their standard AT counterparts.

1.3 Thesis Outline

The rest of the thesis is organized as follows. In Chapter 2, we provide the necessary background information for this study. We offer an overview of adversarial machine learning (AML), covering four main research directions: adversarial attacks, adversarial defenses, certified robustness, and real-world applications of AML. Additionally, we delve into the details of adversarial examples and adversarial training, which are the primary focuses of this thesis. Moreover, we discuss the background of representation learning, multi-objective optimization, and distributional robustness, as these techniques play a crucial role in our efforts to enhance adversarial robustness.

Chapter 3 presents our contributions to improving adversarial robustness from a representation learning perspective. Specifically, we introduce the Adversarial Divergence Reduction (ADR) method (Bui et al., 2020), a novel regularization approach for learning robust representations. Furthermore, we present the Adversarial Supervised Contrastive Learning (ASCL) framework (Bui et al., 2021a), an advanced version of ADR that exhibits improvements in methodology, performance, and understanding.

While the previous chapter focuses on enhancing adversarial robustness within a single model, Chapter 4 shifts the perspective to ensemble learning. We discuss our contributions in this area, beginning with the introduction of the Crossing Collaborative Ensemble method (Bui et al., 2021b), a novel collaboration strategy that enhances the ensemble model's adversarial robustness. Additionally, we present the Task-Oriented Multi-Objective Optimization (TA-MOO) framework (Bui et al., 2023), which employs a multi-task optimization approach to generate transferable adversarial examples. This work demonstrates that the generated adversarial examples can improve the ensemble model's adversarial robustness.

Chapter 5 focuses on improving adversarial robustness through the lens of distributional robustness. We introduce the Unified Distributional Robustness (UDR) framework (Bui et al., 2022), which establishes a connection between Wasserstein distributional robustness and current state-of-the-art adversarial training methods. While the previous two perspectives focus on improving adversarial robustness at a pointwise level, this framework considers adversarial effects at a distributional level, enhancing generalization performance on unseen data.

Finally, in Chapter 6, we conclude our thesis by summarizing the major findings and discussing the limitations of our work, as well as outlining potential future directions.

Chapter 2

Background

In this chapter, we delve into the background of adversarial machine learning, providing a detailed definition of adversarial attacks and defenses, as well as an overview of some of the most popular methods in each category. We then provide a background of representation learning, multi-objective optimization, and distributional robustness, which are the tool sets that we will use to develop our defense strategies in the following chapters.

2.1 Notions and Terminologies

We first establish the machine learning setting and terminologies throughout the thesis. We begin by introducing the base target model, also known as the victim model, which represents the model under attack. Subsequently, we introduce the ensemble model, consisting of a collection of models employed to defend against adversarial attacks. Lastly, we present the distance metric, a function that measures the distance between two samples, which has been used frequently in this thesis.

The Target Model. We consider a supervised learning problem, where we are given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ consisting of n samples, where $\mathbf{x}_i \in \mathcal{X}$ is the input and $y_i \in \mathcal{Y}$ is the corresponding label. The goal is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ that maps an input $\mathbf{x} \in \mathcal{X}$ to a label $y \in \mathcal{Y}$. The function f is modeled by a neural network with parameters θ, which is trained by minimizing the empirical risk on the training

dataset $\mathcal{D}$:

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\mathbf{x}_i;\theta), y_i), \tag{2.1}$$

where ℓ is the loss function that measures the discrepancy between the prediction $f(\mathbf{x}_i;\theta)$ and the ground-truth label y_i for the i-th sample.

The Loss Function. For classification problems, the most common loss function is the cross-entropy loss which is defined as follows:

$$\ell(f(\mathbf{x}_i;\theta), y_i) = -\sum_{j=1}^{k} y_{ij} \log \frac{\exp(f_j(\mathbf{x}_i;\theta))}{\sum_{l=1}^{k} \exp(f_l(\mathbf{x}_i;\theta))}, \tag{2.2}$$

where y_{ij} is the j-th element of the one-hot encoded label y_i and k is the number of classes. In this definition, the output of the model $f(\mathbf{x}_i;\theta)$ is interpreted as a logit vector.

The softmax function σ is commonly used to convert the logit vector to a probability vector, which is defined as follows:

$$\sigma(\mathbf{z})_j = \frac{\exp(z_j)}{\sum_{l=1}^{k} \exp(z_l)}. \tag{2.3}$$

The label-smoothing cross entropy loss (Szegedy et al., 2016) is also commonly used in the literature, where the soft labels are used instead of the one-hot encoded labels. The soft labels are defined as follows:

$$\tilde{y}_i = (1-\alpha)y_i + \frac{\alpha}{k}, \tag{2.4}$$

where α is a hyperparameter that controls the smoothness of the soft labels. If $\alpha = 0$, the soft labels are the same as the one-hot encoded labels, while $\alpha = 1$, the soft labels are uniform distribution over the classes.

It is worth noting that the loss function ℓ for training the target model is not necessarily the same as the loss function used for generating adversarial examples. For example, the Kullback-Leibler (KL) divergence loss is not used as the loss function for training the target model, but commonly used for generating adversarial examples (Zhang et al.,

2019). The KL divergence loss of two probability vectors p and q is defined as follows:

$$\ell_{KL}(p, q) = \sum_{j=1}^{k} p_j \log \frac{p_j}{q_j}. \tag{2.5}$$

Decomposed Model. The model f_θ can be decomposed into two parts: the feature extractor or the encoder $g_\theta : \mathcal{X} \to \mathcal{Z}$ and the classifier $h_\theta : \mathcal{Z} \to \mathcal{Y}$, where $\mathcal{Z}$ is the feature space. The decompositon is defined as follows:

$$f_\theta(\mathbf{x}) = h_\theta(g_\theta(\mathbf{x})). \tag{2.6}$$

Evaluating the Model. The quality of the model f_θ is measured by its performance on the test dataset $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, where m is the number of samples in the test dataset. The performance is measured by the test accuracy, which is defined as the proportion of correctly classified samples in the test dataset:

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\{h_\theta(g_\theta(\mathbf{x}_i))=y_i\}}. \tag{2.7}$$

The performance of the model also can be measured by the test loss, which is defined as the average loss on the test dataset:

$$\frac{1}{m} \sum_{i=1}^{m} \ell(f_\theta(\mathbf{x}_i), y_i). \tag{2.8}$$

When evaluating robustness of the model, the test robust accuracy and the test robust loss are measured on the adversarial test dataset $\mathcal{D}_{\text{adv}} = \{(\mathbf{x}_i^a, y_i)\}_{i=1}^m$, where the adversarial examples are generated using the adversarial attack algorithm $\mathcal{A}$.

Ensemble Model. An ensemble model is a collection of K models $f_{\theta_1}, \ldots, f_{\theta_K}$, where each model is parameterized by θ_k and has the same goal as the target model. The model θ_k can be called as a base model/classifier or an ensemble member, depending on the context.

Ensemble mechanism is the process of combining the predictions of the ensemble members to make the final prediction. The ensemble mechanism can be as simple as taking

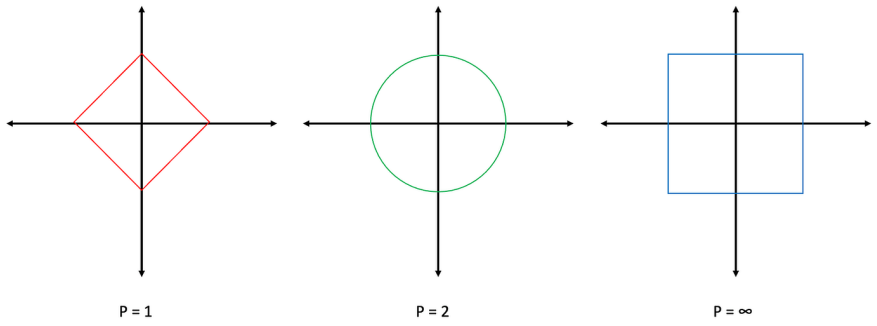


FIGURE 2.1: Illustration of the unit L_p norms in 2D. Image source: `https://en.wikipedia.org/wiki/Norm_(mathematics)`

the majority vote of the predictions of the ensemble members or averaging the predictions of the ensemble members. Averaging predictions can be done in the logit space or the probability space, depending on the context.

Distance Metric. Distance metric is a function that measures the distance between two samples which has been used frequently in this thesis, i.e., measuring the perceptually similarity between an input image and its adversarial counterpart. Generally, the L_p distance between two samples $\mathbf{x}$ and $\mathbf{x}'$ is defined as follows:

$$d_p(\mathbf{x}, \mathbf{x}') = \left(\sum_{i=1}^{d} |x_i - x_i'|^p \right)^{1/p}, \tag{2.9}$$

where d is the dimension of the input space.

Figure 2.1 illustrates the unit L_p norms in 2D of the three most common values of p: $p = 1$, $p = 2$, and $p = \infty$.

The L_1 distance is defined as the sum of absolute differences between two samples, which is also called the Manhattan distance. It is defined as follows:

$$d_1(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{d} |x_i - x_i'|. \tag{2.10}$$

Geometrically, it is the distance between two points in a city if a person can only travel along a city block grid.

The L_2 distance is also called the Euclidean distance which measures the magnitude or length of the vector between two samples. It is defined as follows:

$$d_2(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=1}^{d} (x_i - x_i')^2}. \tag{2.11}$$

Geometrically, it is the shortest distance between two points in a plane.

The L_∞ distance is defined as the maximum absolute difference between two samples, which is also called the Chebyshev distance.

$$d_\infty(\mathbf{x}, \mathbf{x}') = \max_{i=1}^{d} |x_i - x_i'|. \tag{2.12}$$

In other words, it represents the maximum distance between two points along any coordinate dimension. In optimization problems, the L_∞ is used for robust optimization or worst-case analysis, where the objective is to minimize the maximum deviation from a target value. In adversarial machine learning literature, the L_∞ distance is the most commonly used distance metric to measure the perceptual similarity between the input image and the adversarial example. For example, Goodfellow et al. (2015) argue that the L_∞ distance is the most appropriate distance metric given a maximum budget on the perturbation (Carlini and Wagner, 2017).

In Chapter 5 we use the Wasserstein distance to measure the distance between two distributions which will be discussed in Section 2.7.1.

2.2 An Overview of Adversarial Machine Learning

Adversarial machine learning (AML) is a subset of the broader field known as trustworthy machine learning (TML). TML research aims to develop and deploy machine learning systems that possess qualities of reliability, transparency, fairness, and security. The primary objective is to establish confidence in the ability of machine learning models and algorithms to make precise predictions and informed decisions, while also addressing potential risks and biases. The significance of trustworthiness becomes paramount as machine learning models find increasingly critical applications in sectors such as healthcare, finance, autonomous vehicles, and cybersecurity.

AML plays a crucial role in enhancing the security and trustworthiness of machine learning systems. It focuses on understanding and defending against adversarial attacks, where malicious actors intentionally manipulate input data to deceive or exploit vulnerabilities in machine learning models. The importance of AML lies in its ability to identify and address weaknesses in models, ensuring their reliability and trustworthiness. By studying adversarial attacks, researchers and practitioners can develop defense mechanisms and techniques that enhance the resilience of machine learning systems. AML helps protect critical applications such as autonomous vehicles, cybersecurity systems, and fraud detection, where the consequences of a successful attack can be severe. By actively considering and mitigating adversarial risks, AML contributes to building more secure and trustworthy machine learning models, ultimately fostering the adoption of these technologies in sensitive domains.

There are four main research directions in the field of AML including adversarial attacks, adversarial defenses, certified robustness, and applications of AML in real-world systems.

Adversarial Attacks. Briefly, adversarial attacks are techniques that can be used to manipulate input data to cause a machine learning model to make incorrect predictions or decisions. One of the most common types of adversarial attacks is the *adversarial examples*, where a small amount of noise is added to the input data to cause the model to misclassify it.

Adversarial Defenses. refers to a set of techniques and strategies employed to protect machine learning models from adversarial attacks. The goal of adversarial defenses is to improve the robustness of machine learning models, making them more resilient to adversarial attacks. There are various approaches to adversarial defenses, including robustifying the training process, preprocessing the input data, detecting specious input/output, and many more.

Certified robustness. refers to the idea of proving that a machine learning model is truly robust to adversarial attacks, by providing a formal guarantee that the model will not be affected by any adversarial attacks within a certain range. Notable examples of certified robustness include the work of Wong et al. (2019a), Wong and Kolter (2018), Raghunathan et al. (2018) which provide formal guarantees of robustness for neural networks.

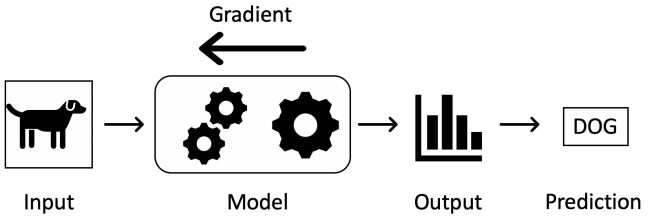
AML for Good. Last but not least, one of the most practical and impactful research directions in AML is studying the applications of AML in real-world systems. Research in this direction explores the behavior and limitations of machine learning models, revealing biases and preferences. For example, Geirhos et al. (2018) demonstrated that machine learning models are biased toward texture rather than shape when classifying images, while Brendel and Bethge (2019) showed that these models rely on bag-of-local features when making predictions. Geirhos et al. (2020) hypothesized that machine learning models prefer learning shortcuts rather than the actual underlying concepts from the data

On the other hand, adversarial examples have served as a useful tool to improve the generalization of models. Techniques like VAT (Virtual Adversarial Training) proposed by Miyato et al. (2018) and SAM (Sharpness-Aware Minimization) developed by (Foret et al., 2021) leverage adversarial learning principle to improve the overall performance of the model.

Moreover, AML research explores practical applications that have positive benefits. For instance, poisoning data can be strategically employed to immunize public photos against harmful manipulation, as demonstrated by Salman et al. (2023). Similarly, Sablayrolles et al. (2020) illustrated how poisoning data can be utilized to track whether a model has been trained on a specific "marked" dataset, aiding in identifying potential data leaks. Cloaking techniques, as proposed by Wu et al. (2020a), are developed to protect the privacy of users' data, ensuring that sensitive information remains secure. Adversarial perturbations also contribute to the resilience of model predictions against environmental changes, as shown by Salman et al. (2021). Additionally, backdoor attacks serve as an effective means of watermarking models to safeguard against intellectual property theft, as exemplified by Adi et al. (2018), Zhang et al. (2018), Uchida et al. (2017).

By exploring these real-world applications, AML research not only addresses critical challenges but also paves the way for tangible solutions that enhance security, privacy, and robustness in machine learning systems.

In this thesis, we focus on the first two research directions, adversarial attacks, and adversarial defenses which are the main topics of the next two sections.



Attack setting	Input	Architecture	Output	Gradient
White-box	✓	✓	✓	✓
Black-box	✓	⊘	✓	⊘
No-box	⊘	⊘	⊘	⊘

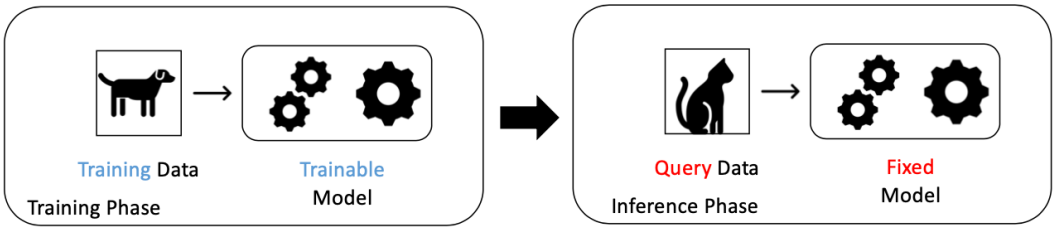
FIGURE 2.2: White-box and black-box settings.

2.3 Adversarial Attacks

Adversarial attacks are a class of attacks that exploit vulnerabilities in a target model to manipulate its behavior. This line of research in machine learning has a rich history, with foundational contributions by Dalvi et al. (2004), Lowd and Meek (2005), Barreno et al. (2006, 2010), Biggio et al. (2010, 2013). However, the field gained significant attention following the seminal work of Szegedy et al. (2014), which revealed the vulnerability of deep neural networks to adversarial examples. This vulnerability poses a substantial threat to the security and reliability of machine learning models, given their growing use in critical applications like autonomous vehicles, cybersecurity systems, and fraud detection. Since then, extensive research has been conducted on adversarial attacks, encompassing various attack types, settings, and goals. In the following, we provide a brief overview of the main types of adversarial attacks, beginning with the definition of white-box and black-box settings.

2.3.1 Category of Adversarial Attacks

White-box and Black-box Settings. Figure 2.2 provides a visual illustration of the white-box and black-box settings in the context of machine learning. The white-box attack refers to when the attacker has full knowledge of the target model, including its



Attack Category	Data	Training	Inference	Attacker's Aim
Poisoning Attack	W	B		Corrupting a learned model
Backdoor Attack	W			Injecting backdoor into a model
Model extraction			B	Reconstructing model functionality
Privacy Attack			W/B	Extracting sensitive training data
Evasion Attack			W/B	Manipulating model's prediction

W: White-box setting, attacker knows everything of model (i.e., architecture, parameters, training strategy), defending strategy (e.g., data augmentation) and intermediately adapt its attack accordingly to defending strategy
B: Black-box setting, attacker has no access to model internal information, just receive model's output (complete probability or just final prediction)

FIGURE 2.3: Category of Adversarial attacks based on their access to the data, training or inference process of the target model. Adapted from Chen (2022).

architecture, parameters, and training data. This type of attack is commonly associated with the adaptive setting, as the attacker can access the defense mechanism and dynamically adjust the attack strategy based on feedback.

Conversely, in the black-box setting, the attacker has no access to the target model's internal information, and can only query the model and observe the output, which can be the prediction or confidence score. However, the number of queries is an important consideration, as excessive queries may trigger the target model's anomaly detection system. There is also a restricted black-box setting, where the attacker has no access to the target model at all. Under this setting, the attacker can only use transferred adversarial examples from other models to attack the target model.

Given the two fundamental settings, Figure 2.3 categorizes adversarial attacks based on their access to the data, training, or inference process of the target model when making the attack.

Poisoning Attacks. A poisoning attack is a specific type of adversarial attack that aims to corrupt a target model during its training process, affecting its behavior when deployed in real-world scenarios (Biggio et al., 2012, Shafahi et al., 2018, Steinhardt et al., 2017, Tolpegin et al., 2020). To achieve this goal, an attacker requires to access

to both the training data and the training process of the target model. By strategically manipulating the training data, the attacker introduces malicious data points to deceive the model's learning process.

The success of a poisoning attack depends on various factors, including the quantity and placement of the malicious data points, the specific machine learning algorithm employed, and the inherent characteristics of the data itself. Poisoning attacks can be particularly potent when targeting machine learning models trained on small datasets or those heavily influenced by a few pivotal data points. These attacks exploit vulnerabilities in the training process, potentially leading to incorrect or biased predictions when the model is utilized in real-world applications.

Backdoor Attacks or Trojan Attacks. While poisoning attacks seek to corrupt a target model in order to introduce biases or inaccuracies in its predictions, backdoor attacks have a different objective: to implant a concealed backdoor within the model that can be activated by a specific input pattern (Gu et al., 2017, Liu et al., 2018b, Chen et al., 2017). In the absence of the trigger, the model operates normally; however, when the trigger is present, the model's behavior is manipulated in a manner advantageous to the attacker.

Recent advancements in backdoor attacks have focused on the development of more intricate triggers that are challenging to detect (Nguyen and Tran, 2021, Doan et al., 2021). Additionally, researchers have explored methods that directly modify model weights or the model structure without relying on the traditional training process (Dumford and Scheirer, 2020, Rakin et al., 2020, Li et al., 2022). These advancements allow attackers to subtly alter the model's functionality, making it more difficult for security measures to identify and mitigate the presence of the backdoor.

Model Extraction. Model extraction attacks, also known as model stealing, pose a security threat where an attacker aims to extract or replicate a target model without direct access to the model itself (Truong et al., 2021, Jia et al., 2021). These attacks are carried out by analyzing the inputs and outputs of the model. The motivation behind model extraction attacks can range from intellectual property theft to generating

adversarial examples or reverse engineering the decision-making process of the model for malicious purposes.

While model extraction shares similarities with model distillation, it differs in that the attacker lacks access to the model's parameters or training data (Truong et al., 2021). In order to execute a model extraction attack, the attacker typically queries the target model using samples from a surrogate dataset that is assumed to be similar to the original training set. However, it is crucial for the attacker to carefully consider the number of queries, as excessive queries might trigger the anomaly detection system of the target model, potentially exposing the attack.

On a positive note, model extraction attacks can also serve the purpose of interpreting the decision-making process of a black-box model (Bastani et al., 2017a,b). In such cases, model extraction can provide valuable insights into understanding how the model arrives at its predictions, even when the inner workings of the model are not directly accessible.

Privacy Attack. A privacy attack refers to an attempt to extract sensitive information from a target model, posing a significant threat to privacy, especially in sensitive domains like healthcare, finance, and law enforcement, where the confidentiality of data is critical.

These attacks can manifest in various forms, one of which is membership inference attacks (Choquette-Choo et al., 2021, Carlini et al., 2022, Dwork et al., 2017), aiming to determine if an individual's data was part of the training dataset By analyzing the model's responses to different inputs, an attacker can infer the presence or absence of certain data points, potentially revealing sensitive information about individuals (Rigaki and Garcia, 2020).

Evasion Attacks. Last but the most important type of adversarial attacks is evasion attacks or adversarial examples (Szegedy et al., 2014, Goodfellow et al., 2015, Madry et al., 2018), which is the main adversarial attack that we will focus on in this thesis. Evasion attacks in machine learning involve modifying the input data in a way that is imperceptible to human observers but can cause the model to make incorrect predictions or classifications. The existence of adversarial examples is a significant threat to the

security and reliability of machine learning models, particularly in applications such as image recognition, speech recognition, and natural language processing. In the next section, we will provide a detailed definition of adversarial examples, as well as an overview of some of the most popular methods for generating adversarial examples.

2.3.2 Adversarial Examples

Definition. Given a machine learning model f and an input x with corresponding label y, an adversarial example x^a is a perturbed version of x that causes f to make an incorrect prediction, i.e., $f(x^a) \neq y$. To make the adversarial examples to be a real threat, there is a condition on the perturbation $\delta = d(x^a, x)$ that it should be small enough to be imperceptible to human observers, measured by some distance metric d.

Formally, the adversarial examples are firstly defined in (Szegedy et al., 2014) as follows:

$$\begin{aligned} \underset{x^a}{\text{minimize}} \quad & d(x^a, x) \\ \text{subject to} \quad & f(x^a) \neq y \end{aligned} \tag{2.13}$$

Perceptual Similarity Measure. In computer vision, the most common metric to measure the difference between two images is the L_p norm, which is defined as follows:

$$d(x^a, x) = \left(\sum_{i=1}^{n} |x_i^a - x_i|^p \right)^{\frac{1}{p}} \tag{2.14}$$

where n is the number of pixels in the image, and x_i and x_i^a are the i-th pixel of the original image x and the adversarial image x^a, respectively. Further discussion about distance metrics can be found in Section 2.1. While the L_p norm is the most commonly used distance metric, it cannot capture well the perceptual similarity between images in some cases. For example, two images that are shifted by a few pixels may have a large L_p distance but be perceptually similar. Wasserstein distance is recently proposed to address this issue (Wong et al., 2019b, Wu et al., 2020c).

It is worth noting that, in discrete domains, such as natural language processing, the perceptual similarity between two inputs is often measured by the edit distance, which is the minimum number of edits required to transform one input into another. However,

unlike in computer vision, the perturbation in discrete domains is easier to detect, as it often changes the meaning of the input.

Our thesis focus on the classification problems in the image domain, thus we utilized the L_∞ norm which is also the most commonly used distance metric in the literature.

The Optimization Problem of Generating Adversarial Examples. The problem of finding adversarial examples is often formulated as an optimization problem introduced in Szegedy et al. (2014), as shown in Equation 2.13. However, solving this optimization problem is often computationally expensive, as it requires evaluating the model f multiple times. Moreover, the main optimization problem is in the distance metric d, which is not easy to interpret. For example, it is nontrivial to change the perturbation δ in order to keep adversarial example x^a imperceptible to human observers while remaining being predicted incorrectly.

To address these issues, Szegedy et al. (2014) consider the following optimization problem using box-constrained Limited-memory BFGS (Liu and Nocedal, 1989):

$$\underset{x^a}{\text{maximize}} \quad \mathcal{L}(f(x^a), y) - \lambda d(x^a, x) \tag{2.15}$$

where $\mathcal{L}$ is the loss function of the model f which commonly is cross-entropy loss, and λ is a hyperparameter that controls the trade-off between the loss and the distance metric. This formulation allows us to solve the optimization problem in a more interpretable way, as we can directly control the loss and the distance metric. The optimal λ can be found by using line search or binary search. However, the optimal solution of Equation 2.15 is not necessarily the optimal adversarial examples that maximize the prediction loss $\mathcal{L}(f(x^a), y)$. To guaranteely find the success adversarial examples, Goodfellow et al. (2015) proposed to solve the following optimization problem:

$$\begin{aligned} \underset{x^a}{\text{maximize}} \quad & \mathcal{L}(f(x^a), y) \\ \text{subject to} \quad & d(x^a, x) \leq \epsilon \end{aligned} \tag{2.16}$$

where ϵ is a hyperparameter that controls the maximum perturbation allowed. This formulation aims to priotize to maximize the prediction loss $\mathcal{L}(f(x^a), y)$ over the perturbation to guarantee to find the successful adversarial examples.

Targeted attacks. The above optimization problems are the untargeted attacks, which aim to cause the model to make an incorrect prediction to any class. In some cases, we may want to cause the model to make an incorrect prediction to a specific class, which is called targeted attacks. The targeted attacks can be formulated as follows:

$$\begin{aligned} \underset{x^a}{\text{minimize}} \quad & \mathcal{L}(f(x^a), y_t) \\ \text{subject to} \quad & d(x^a, x) \leq \epsilon \end{aligned} \tag{2.17}$$

where y_t is the target class.

2.3.3 Notable Adversaries

While considering the same optimization problem as in Equation 2.16 (untargeted attack) or Equation 2.17 (targeted attack), many different adversaries have been proposed which can be classified into two main categories: gradient-based methods and gradient-free methods.

Gradient-based Attacks

FGSM. While adversarial examples can be easily defined and generated (Szegedy et al., 2014, Goodfellow et al., 2015), the underlying reasons behind their existence in deep neural networks (DNNs) are still not fully understood. One of the earliest attempts to explain this phenomenon was made by Goodfellow et al. (2015). They hypothesized that the designs of modern DNNs, which prioritize linear behavior for computational efficiency, also make them vulnerable to inexpensive analytical attacks. Building upon this hypothesis, Goodfellow et al. (2015) proposed the Fast Gradient Sign Method (FGSM), a one-step gradient-based technique, as follows:

$$x^a = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f(x), y)) \tag{2.18}$$

where ϵ is the maximum perturbation allowed, and sign is the sign function. Because of using the sign function, it is optimized for the L_∞ norm.

The subsequent work by Tramèr et al. (2018) introduced R+FGSM, which incorporates a small random initialization step prior to linearizing the loss function. This initialization helps the method escape the non-smooth neighborhood of the input data x. The procedure for R+FGSM is as follows:

$$x^a = x' + (\epsilon - \alpha) \cdot \text{sign}(\nabla_{x'} \mathcal{L}(f(x'), y)) \tag{2.19}$$

where $x' = x + \alpha \cdot \text{sign}(\mathcal{N}(0, 1))$ is the input after one small random step. In their study, Tramèr et al. (2018) demonstrated that the inclusion of random initialization significantly enhances the effectiveness of R+FGSM compared to FGSM, regardless of whether the models are robust or non-robust. This approach has subsequently been adopted in various other attack methods, including MIM (Momentum Iterative Method) (Dong et al., 2018) and PGD (Projected Gradient Descent) (Madry et al., 2018).

Basic Iterative Method (BIM) and Momentum Iterative Method (MIM). The FGSM method perturbs images by taking a single large step in the direction that maximizes the loss function. However, this approach may not be sufficient, particularly when dealing with highly complex loss surfaces. To address this limitation, a straightforward extension involves iteratively taking multiple small steps while adjusting the direction after each step.

One of the pioneering works that followed this strategy is the Basic Iterative Method (BIM) introduced by Kurakin et al. (2016). BIM is defined as follows:

$$x_0^a = x, \quad x_{t+1}^a = \Pi_{B(x,\epsilon)} \left(x_t^a + \alpha \cdot \text{sign}(\nabla_{x_t^a} \mathcal{L}(f(x_t^a), y)) \right) \tag{2.20}$$

where α is the step size, and $\Pi_{B(x,\epsilon)}$ is the projection operator that projects the adversarial examples back to the ϵ-ball around the original image x. The BIM method starts with the original image x and then iteratively takes multiple small steps with step size α to find the adversarial examples.

The projection operator $\Pi_{B(x,\epsilon)}$ for L_∞ norm is defined as follows:

$$\Pi_{B(x,\epsilon)}(x^a) = \text{clip}_{x,\epsilon}(x^a) = \text{clip}(x^a, x - \epsilon, x + \epsilon) \tag{2.21}$$

where clip is the element-wise clipping function.

The Momentum Iterative Method (MIM) (Dong et al., 2018) is an extension of BIM that incorporates momentum into the iterative process to stabilize the update directions and accelerate convergence.

The MIM method is defined as follows:

$$x_0^a = x, \quad x_{t+1}^a = \Pi_{B(x,\epsilon)} \left(x_t^a + \alpha \cdot \text{sign}(g_{t+1}) \right) \tag{2.22}$$

where μ is the momentum parameter and g_{t+1} is the accumulated gradient at step $t+1$ defined as follows:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^a} \mathcal{L}(f(x_t^a), y)}{\|\nabla_{x_t^a} \mathcal{L}(f(x_t^a), y)\|_1} \tag{2.23}$$

where $\| \cdot \|_1$ is the L_1 norm.

PGD. While iterative methods like BIM and MIM are more effective than FGSM, they are still vulnerable to the non-smooth neighborhood of the input data x. To address this issue, Madry et al. (2018) proposed the projected gradient descent (PGD) method, which is a variant of BIM with uniform random initialization. The PGD procedure is defined as follows:

$$x_0^a = x + n, \quad x_{t+1}^a = \Pi_{B(x,\epsilon)} \left(x_t^a + \alpha \cdot \text{sign}(\nabla_{x_t^a} \mathcal{L}(f(x_t^a), y)) \right) \tag{2.24}$$

where α is the step size, and $\Pi_{B(x,\epsilon)}$ is the projection operator that projects the adversarial examples back to the ϵ-ball around the original image x. The PGD method starts with the original image x with a small random initialization n and then iteratively takes multiple small steps with step size α in order to search for the adversarial examples. Since its introduction, the PGD attack has emerged as one of the most widely used attacks in the literature, as it is easy to implement and recognised as one of the most powerful gradient-based attacks (Athalye et al., 2018).

Variants of PGD. While the Projected Gradient Descent (PGD) attack has gained popularity for its effectiveness, it is not without limitations. One of the main drawbacks is its computational cost, particularly when dealing with large models and datasets such

as ImageNet (Deng et al., 2009). To address this issue, several variants of PGD have been proposed to improve the efficiency of the attack while maintaining its effectiveness. It is worth noting that, the main motivation for accelerating generation speed is to benefit the adversarial training process, which will be introduced later. Several methods have been proposed to accelerate the PGD attack while maintaining its effectiveness (Wong et al., 2019a, Andriushchenko and Flammarion, 2020, Shafahi et al., 2019).

Another issue of the PGD is that it is sensitive to the step size α. With a large step size, the PGD may not converge to successful adversarial examples while with a small step size, the PGD requires more iterations to converge. Adaptive adjusting the step size during the optimization process is recently proposed in Croce and Hein (2020) to address this issue. PGD attack also suffers from the sensitivity to the scale of logits in the standard cross-entropy loss. To address this issue, Croce and Hein (2020) proposed an alternative logit loss which is both shift and rescaling invariant.

Other Gradient-based Methods. Finally, combining multiple attacks is also a common practice to improve the robustness of the attacks, where Auto-Attack is one of the most popular methods. It combines two new versions of PGD with the white-box attack FAB (Croce and Hein, 2019) and the black-box attack Square Attack (Andriushchenko et al., 2020) to form a parameter-free, state-of-the-art attack. The authors also proposed a benchmark (Croce et al., 2020) with many defense methods evaluating with Auto-Attack which is also a useful resource for the community. Therefore, besides PGD, Auto-Attack is considered the new standard evaluation for adversarial robustness, which is also used in our experiments.

Gradient-free Attacks

One of the main drawbacks of the gradient-based methods is that they required to access model's gradients, which is not always possible in practice. This assumption is often violated in the black-box setting, where the attacker only has access to the model's predictions. Many gradient-free methods have been proposed to address this issue, specifically in the black-box setting.

Decision-based attacks is type of gradient-free methods which only requires the model's predictions, with boundary attack (Brendel et al., 2017) and HopSkipJumpAttack (Chen et al., 2020b) are two representative methods. These attacks adopt an iterative approach, commencing from an initial point that is already adversarial, and then executing a randomized traversal along the boundary between the adversarial and non-adversarial regions. The movement is guided by two critical constraints: minimizing the distance to the target image while maintaining adversarial status. These attacks have been demonstrated its practical capability on real-world black-box systems (Brendel et al., 2017), showing a more realistic threat to the security of machine learning models.

2.4 Adversarial Defenses

2.4.1 Evaluating Robustness

Unlike the standard evaluation of machine learning models, evaluating the robustness of adversarial examples is extremely challenging. The main reason is that the adversarial examples are not naturally occurring data, which requires crafting them using adversaries. However, generating adversarial examples that truly reflects the threat model requires a lot of genuine efforts. For example, while gradient-based attacks are considered as the most powerful attacks, they are not always the most appropriate threat model. It is because the loss surface of the deep learning model is highly non-convex and non-smooth, which makes it easy to be trapped in local minima.

Gradient masking is one of the most common techniques used to prevent gradient information from being used to craft adversarial examples. Gradient-based attacks, such as PGD rely on computing gradients of the model's loss function to generate adversarial examples. However, in some cases, the model's architecture or training process may unintentionally or deliberately hinder the computation of accurate gradients, thereby impeding adversarial attacks. Gradient masking can occur due to various reasons, such as the use of non-differentiable operations, or defensive mechanisms. For example, Athalye et al. (2018) showed that 7 over 9 studied defenses rely on this phenomenon to prevent adversarial attacks, which later can be easily broken by gradient-free attacks such as transferred attacks or Backward Pass Differentiable Approximation (BPDA) (Athalye

et al., 2018). Therefore, it is important to recognize the presence of gradient masking when evaluating adversarial robustness.

Secondly, adversarial attacks are often sensitive to specific hyperparameters, making them effective only in a specific setting, while underperforming in other settings. For example, the PGD attack is sensitive to the step size α, the number of iterations T, and the scale of logits in the standard cross-entropy loss. On the other hand, transferred attacks are sensitive to the choice of the substitute model. Boundary attack often requires a large number of iterations to find adversarial examples and is not effective in real-world systems. Therefore, it is important to evaluate the robustness of adversarial examples in a wide range of settings. Finally, Carlini et al. (2019) proposed a checklist outlining common pitfalls in evaluating adversarial robustness and how to avoid them.

2.4.2 Adversarial Training

Procedure of a standard Adversarial Training method. The initial idea of adversarial training is first proposed in (Szegedy et al., 2014), where a deep model is trained a mixture of benign and adversarial examples. The optimization problem of adversarial training as follows:

$$\underset{\theta}{\text{minimize}} \quad \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\mathcal{S}} \mathcal{L}(f_\theta(x+\delta), y)\right] \tag{2.25}$$

where $\mathcal{D}$ is the training dataset, $\mathcal{S}$ is the set of all possible perturbations, and $\mathcal{L}$ is the loss function of the model f_θ. The adversarial training is an iterative process, which means that the model is retrained multiple times.

The pseudo code of the adversarial training is shown in Algorithm 1.

Algorithm 1 Procedure of a Standard Adversarial Training

Require: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, number of training iterations T, learning rate η, an adversary $\mathcal{A}$ with perturbation budget ϵ, and tradeoff parameter λ

1: Initialize model parameters θ randomly
2: **for** $t = 1, \ldots, T$ **do**
3: Sample a mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^M$ from $\mathcal{D}$
4: Generate adversarial examples $\mathcal{B}^a = \{(x_i^a, y_i)\}_{i=1}^M$ by applying $\mathcal{A}$ to $\mathcal{B}$
5: Update model parameters $\theta \leftarrow \theta - \eta\nabla_\theta \sum_{i=1}^M [(1-\lambda)\mathcal{L}(f_\theta(x_i), y_i) + \lambda\mathcal{L}(f_\theta(x_i^a), y_i)]$
6: **end for**

During each iteration of adversarial training, a batch of benign samples, denoted as $\mathcal{B}$, is randomly selected from the training dataset $\mathcal{D}$. Subsequently, an adversary $\mathcal{A}$ is employed to generate adversarial examples $\mathcal{B}^a$ from the batch $\mathcal{B}$. The model is then updated by minimizing the loss function $\mathcal{L}$ computed on both the benign and adversarial examples. A tradeoff parameter λ is introduced to determine the relative importance of these two terms in the loss function.

When $\lambda = 0$, the model is solely trained on benign examples, resulting in a training process similar to standard training. Conversely, if $\lambda = 1$, the model exclusively trains on adversarial examples. While this setting enhances the model's robustness against adversarial attacks, it may result in a noticeable decrease in accuracy on benign examples. In practice, a commonly used value for λ is 0.5, striking a balance between robustness and accuracy. Finding the optimal tradeoff between robustness and accuracy remains a challenging problem in adversarial training (Zhang et al., 2019).

Adversarial Training with PGD. Adversarial training with PGD (Projected Gradient Descent) adversarial examples (Madry et al., 2018) stands as the most widely adopted method in adversarial training, and its pseudo code is presented in Algorithm 2.

Algorithm 2 Adversarial Training with PGD (Madry et al., 2018)

Require: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, number of training iterations T, attack setting including perturbation budget ϵ, step size α, number ofiterations K, projection operation $\Pi_{B(x,\epsilon)}$, and tradeoff parameter λ

Initialize model parameters θ randomly
for $t = 1, \ldots, T$ **do**
  Sample a mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^M$ from $\mathcal{D}$
  Initialize $x_i^a = x_i + \mathcal{U}(-\epsilon, \epsilon)$ for $i = 1, \ldots, M$
  for $k = 1, \ldots, K$ **do**
    $x_i^a \leftarrow \Pi_{B(x_i,\epsilon)}(x_i^a + \alpha \text{sign}(\nabla_{x_i^a} \mathcal{L}(f_\theta(x_i^a), y_i)))$ for $i = 1, \ldots, M$
  end for
  Update model parameters $\theta \leftarrow \theta - \eta \nabla_\theta \sum_{i=1}^M [(1-\lambda)\mathcal{L}(f_\theta(x_i), y_i) + \lambda \mathcal{L}(f_\theta(x_i^a), y_i)]$
end for

The PGD attack employs an iterative approach, commencing with a small random initialization and subsequently taking multiple small steps with a step size of α to search for adversarial examples. The objective is to find perturbations that maximize the model's loss within the vicinity of the original image x, constrained by an ϵ-ball defined by the projection operator $\Pi_{B(x,\epsilon)}$. When using the L_∞ norm, the projection operator $\Pi_{B(x,\epsilon)}$

is equivalent to applying an element-wise clipping function to ensure the perturbations remain within the specified range.

2.5 Representation Learning

2.5.1 Overview

The primary objective of representation learning is not only to reduce the dimensionality of the input but also to capture the inter-sample correlations that enable effective downstream task performance (Bengio et al., 2013). It can be conducted through supervised or unsupervised methods, with the former requiring labeled data such as images of various classes for classification models and semantic segmented images for segmentation models. While representation learned from supervised learning is highly effective for downstream tasks, it requires large labeled data sets, which are expensive and time-consuming to acquire. Thus, a great amount of effort has been dedicated to unsupervised representation learning to leverage abundant unlabeled data.

Self-Supervised Learning (SSL) (Chen et al., 2020a, He et al., 2020, Grill et al., 2020) has emerged as a powerful tool for Deep Learning models to exploit structure from enormous amounts of unlabeled data, facilitating its transfer to downstream tasks. The key success factor of SSL is choosing a pretext task that heuristically introduces interaction among different parts of the data.

One of the earliest SSL frameworks in computer vision was the Auto-Encoder (Kramer, 1991, Kingma and Welling, 2013), which used the pretext task of reconstructing the input image. Similarly, in natural language processing, word2vec (Mikolov et al., 2013) is a well-known SSL framework that uses the pretext task of predicting the surrounding context words from the current word, resulting in the learning of powerful representations of words that preserve semantic and syntactic relationships without the need for labeled data. With the exponential growth of data and computation capacity, SSL has underpinned the recent success of Deep Learning in various fields and applications, such as Generative AI (Radford et al., 2021), Medical Imaging (Azizi et al., 2021), Protein Folding (Jumper et al., 2021), and many more.

More recently, Contrastive Learning (CL) has emerged as the most effective SSL framework, using the pretext task of learning a representation that maximizes the similarity between augmented variants of the same instance, while minimizing those from different instances. While recently taken off as the most effective SSL framework, the principle of Contrastive Learning has a long history in the field of metric learning (Hadsell et al., 2006, Weinberger and Saul, 2009).

It can be traced back to the early work of Bromley et al. (1993) then later extended by Hadsell et al. (2006) where the authors proposed to learn a metric space in which the distance between two embeddings is proportional to the probability that they are in the same class. Moving to the age of unlabeled data, to obtain similar inputs, we often use semantic preserving transformations such as random cropping, color jittering, rotation, etc to get augmented variants of the same instance. There are many variants of CL have been proposed, however, in the following discussion, we focus on the loss function derivations from the Triplet loss (Schultz and Joachims, 2003, Hoffer and Ailon, 2015) to the InfoNCE loss (Chen et al., 2020a, Oord et al., 2018) and Supervised Contrastive loss (Khosla et al., 2020). We then discuss about the way to construct positive and negative pairs, and finally, we discuss important practical tricks to improve the performance of CL.

2.5.2 Contrastive Losses

Triplet Loss. Triplet loss was originally proposed as a method for metric learning, as described in previous studies (Schultz and Joachims, 2003, Hoffer and Ailon, 2015). This approach has since gained widespread popularity for its effectiveness in face recognition tasks (Schroff et al., 2015). The triplet loss is defined as follows:

$$\mathcal{L}_{\text{Triplet}}(Z) = \sum_{\mathbf{z}_i \in Z, \mathbf{z}_j \in Z_i^p, \mathbf{z}_k \in Z_i^n} \max\left(0, m + d(\mathbf{z}_i, \mathbf{z}_j) - d(\mathbf{z}_i, \mathbf{z}_k)\right), \tag{2.26}$$

where Z is a set of embedded samples, Z_i^p and Z_i^n are sets of positive and negative samples of the anchor i-th sample, respectively. $d(\cdot, \cdot)$ is a distance function, and m is a margin hyper-parameter.

By enforcing the order of distances between the anchor-positive and anchor-negative pairs, the triplet loss is effective in learning a representation such that those with the same labels (i.e., anchor and positive) are closer to each other than those with different labels (i.e., anchor and negative). However, unlike t-SNE (Maaten and Hinton, 2008) or InfoNCE (Oord et al., 2018) which preserves embedding orders via probability distributions, the triplet loss works directly on embedded distances, which is sensitive to the choice of the margin hyper-parameter. Moreover, it is difficult to find informative triplets in high-dimensional spaces, requiring expensive sampling strategies and practical tricks (Schroff et al., 2015).

InfoNCE Loss. The InfoNCE loss (Oord et al., 2018) is a contrastive loss that uses categorical cross-entropy to identify the positive sample among a set of negative samples. More specifically, given an anchor sample $\mathbf{z}_i \in Z$, the InfoNCE loss assumes that there is a positive sample $\mathbf{z}_j$ among a set of negative samples Z_i^n w.r.t. the anchor sample $\mathbf{z}_i$. The InfoNCE loss is defined as follows:

$$\mathcal{L}_{\text{InfoNCE}}(Z) = -\sum_{\mathbf{z}_i \in Z} \log \left(\frac{\exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau\right)}{\exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau\right) + \sum\limits_{\mathbf{z}_k \in Z_i^n} \exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau\right)} \right) \tag{2.27}$$

where τ is a temperature hyper-parameter and sim is a similarity function, which is often the cosine similarity, i.e., $\text{sim}(u, v) = \frac{u^T v}{\|u\|\|v\|}$.

Supervised Contrastive Loss. The SCL (Khosla et al., 2020) takes advantages of the labels to have a better control of the positive and negative pairs.

$$\mathcal{L}_{\text{SCL}}(Z) = \sum_{\mathbf{z}_i \in Z} \sum_{\mathbf{z}_j \in Z_i^p} -\log \left(\frac{\exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau\right)}{\sum\limits_{\mathbf{z}_k \in Z_i^p \cup Z_i^n} \exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau\right)} \right) / \left|Z_i^p\right| \tag{2.28}$$

2.5.3 Contrastive Learning Frameworks

SimCLR. In SSCL, the positives are constructed by applying a set of augmentations $\mathcal{T}$ to the same instance $\mathbf{z}$, while the negatives are constructed by applying the same

augmentations to different instances $\mathbf{z}'$, as shown in Figure **??**. The set of augmentations $\mathcal{T}$ is often a combination of random cropping, color jittering, rotation, etc.

SSL. The SCL framework leverages the idea of contrastive learning with the presence of label supervision to improve the regular cross-entropy loss. The positive set and the negative set are $\boldsymbol{Z}_i^+ = \{\mathbf{z}_j^{\mathcal{T}}, \mathbf{z}_j^{\mathcal{A}} \mid j \neq i, \mathbf{y}_j = \mathbf{y}_i\}$ and $\boldsymbol{Z}_i^- = \{\mathbf{z}_j^{\mathcal{T}}, \mathbf{z}_j^{\mathcal{A}} \mid j \neq i, \mathbf{y}_j \neq \mathbf{y}_i\}$, respectively. As mentioned in Khosla et al. (2020), there is a major advantage of SCL compared with SSCL in the context of regular machine learning. Unlike SSCL in which each anchor has only single positive sample, SCL takes advantages of the labels to have many positives in the same batch size N. This strategy helps to reduce the false negative cases in SSCL when two samples in the same class are pushed apart. As shown in Khosla et al. (2020), the SCL training is more stable than SSCL and also achieves a better performance.

2.5.4 Important factors for Contrastive Learning

Data augmentation. Data augmentation is a crucial factor in machine learning in general, that helps the model to learn invariance to specific transformations of the input. For example, to learn a model that is invariant to color information, we can apply grayscale or color jittering transformations to the input images. In the context of contrastive learning, where positive and negative samples are required to compute the contrastive loss, data augmentation plays an even more important role. Chen et al. (2020a) found that simple transformations such as noise injection are not sufficient to create two different views of the same instance. Stronger techniques such as a combination of random cropping, color jittering, rotation, etc are required in order to generate useful positive and negative samples. The gap between data augmentation can be significant as shown in Table 1 in Chen et al. (2020a), while supervised methods' performance is not affected much by the same set of augmentation.

Batch size. As shown in Figure 9 in Chen et al. (2020a), the batch size is an important factor that strongly affects the performance of the contrastive learning framework. A larger batch size comes with larger positive and negative sets, which helps to generalize the contrastive correlation better and therefore improves the performance. He et al.

(2020) proposed a memory bank to store the previous batch information which can lessen the batch size issue.

Projection head. Normally, the representation vector which is the output of the encoder network has very high dimensionality, e.g., the final pooling layer in ResNet-50 and ResNet-200 has 2048 dimensions. Therefore, applying contrastive learning directly on this intermediate layer is less effective. Alternatively, CL frameworks usually use a projection network $p()$ to project the normalized representation vector $\mathbf{z}$ into a lower dimensional vector $\tilde{\mathbf{z}} = p(\mathbf{z})$ which is more suitable for computing the contrastive loss. To avoid over-parameterized, CL frameworks usually choose a small projection head with only one or two fully-connected layers.

Hard Negative Mining. One of the drawbacks of the InfoNCE loss is that it utilized all negative samples in the batch, which may include easy negative samples that are not informative for learning. For example, those samples that are in different classes and already far away from the anchor sample in the embedding space are not supportive to increase the contrastiveness. Even worse, as shown in Chuang et al. (2020) using false negative samples can lead to significant performance degradation. Increasing batch size or using memory bank (He et al., 2020) implicitly introduces more hard negative samples, but it leads to a higher computational cost (Weng, 2021). To address this issue, there are several works (Robinson et al., 2021, Kalantidis et al., 2020) that proposed to up-weight the samples that are hard to distinguish from the anchor sample.

2.6 Multi-Objective Optimization

2.6.1 Pareto Optimality

Multi-objective optimization (MOO) or Pareto optimization is an area of optimization that deals with optimization problems that involve multiple objectives. MOO has been applied to many fields such as engineering design, economics, and machine learning, where optimal decisions need to be taken in the presence of trade-offs between two or more objectives that are often in conflict with each other. In the context of adversarial machine learning, the trade-off between the attack performance and the perceptual

preservation as discussed in Section 2.3.2 is a typical example of a multi-objective optimization problem, where increasing the attack performance often leads to a decrease in perceptual preservation and vice versa. Another specific example presented in chapter 4 is the trade-off between the attack performance on different base models in the ensemble. Especially when the base models are diverse and have conflict gradients, it is difficult to find a single perturbation that can fool all the base models.

Formally, a multi-objective optimization problem can be defined as follows:

$$\min_{\mathbf{x}\in\mathcal{X}} \mathbf{f}(\mathbf{x}), \tag{2.29}$$

where $\mathbf{x} \in \mathcal{X}$ is the decision variable, $\mathcal{X}$ is the decision space, and $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ is the vector-valued objective function with m objectives.

The goal of the multi-objective optimization problem is to find a solution $\mathbf{x}^* \in \mathcal{X}$ that minimizes all the objectives simultaneously. However, in many cases, it is impossible to find a single solution that minimizes all the objectives simultaneously. Therefore, attention is paid to find Pareto optimal solutions, which are solutions that cannot be improved in any of the objectives without degrading at least one of the other objectives. Formally, a solution x^* is called a Pareto optimal solution if there is no other solution x' such that $f_i(x') \leq f_i(x^*)$ for all $i \in \{1, \ldots, m\}$ and $f_j(x') < f_j(x^*)$ for at least one $j \in \{1, \ldots, m\}$.

The set of all Pareto optimal solutions is called the Pareto front or Pareto set.

Formally, the Pareto front is defined as follows:

$$\mathcal{X}^* = \{\mathbf{x} \in \mathcal{X} \mid \nexists \mathbf{x}' \in \mathcal{X} \text{ such that } \mathbf{f}(\mathbf{x}') \prec \mathbf{f}(\mathbf{x})\}, \tag{2.30}$$

where $\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}')$ means that $\mathbf{f}(\mathbf{x})$ is strictly better than $\mathbf{f}(\mathbf{x}')$ in all objectives.

2.6.2 Multi-Gradient Descent Algorithm

While there are many algorithms for solving multi-objective optimization problems, we focus on the Multi-Gradient Descent Algorithm (MGDA) (D'esid'eri, 2012) which is the foundation tool for our proposed methods in chapter 4. Specifically, MGDA

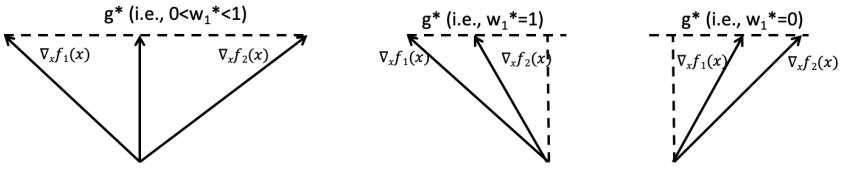


FIGURE 2.4: Illustration of the optimal norm point in the case of two objectives.

combines the gradients of individual objectives into a single optimal gradient direction that increases all objectives simultaneously. The optimal gradient direction corresponds to the minimum-norm point that can be found by solving the quadratic programming problem:

$$w^* = \text{argmin}_{w \in \Delta_m} w^T Q w, \tag{2.31}$$

where $\Delta_m = \left\{\pi \in \mathbb{R}_+^m : \|\pi\|_1 = 1\right\}$ is the m-simplex and $Q \in \mathbb{R}^{m \times m}$ is the matrix with $Q_{ij} = \nabla_{\mathbf{x}} f_i(\mathbf{x})^T \nabla_{\mathbf{x}} f_j(\mathbf{x})$.

Finally, the solution of the problem 2.31 can be found iteratively with each update step $\mathbf{x} = \mathbf{x} + \eta g^*$ where g^* is the combined gradient $g^* = \sum_{i=1}^m w_i^* \nabla_{\mathbf{x}} f_i(\mathbf{x})$ and $\eta > 0$ is a sufficiently small learning rate. Furthermore, D'esid'eri (2012) also proved that by using an appropriate learning rate at each step, we reach the Pareto optimality point $\mathbf{x}^*$ at which there exist $w \in \Delta_m$ such that $\sum_{i=1}^m w_i \nabla_{\mathbf{x}} f_i(\mathbf{x}^*) = \mathbf{0}$.

Figure 2.4 illustrates the minimum-norm point in the case of two objectives corresponding to two gradient vectors $\nabla_{\mathbf{x}} f_1(\mathbf{x})$ and $\nabla_{\mathbf{x}} f_2(\mathbf{x})$. The left figure illustrates the case where the two gradient vectors are partly conflicted, which leads to the optimal solution $w^* = [w_1^*, w_2^*]$ where $0 < w_1^* < 1$. In the middle and the right figure, the two gradient vectors are not conflicted, which leads to the minimum-norm direction being the exact direction of the smaller gradient vector.

2.7 Wasserstein Distance and Distributional Robustness

2.7.1 Wasserstein Distance

Problem of KL divergence. Measuring the distance between two distributions is a fundamental problem in statistics and machine learning. In the context of AML,

it can be used to measure the perceptual similarity between the input image and the adversarial example, or to measure the distance between the data distribution and the adversarial distribution. The most well-known distance metric between two distributions is the Kullback-Leibler (KL) divergence, which is defined as follows:

$$D_{KL}(\mu, \nu) = \int \mu(x) \log \frac{\mu(x)}{\nu(x)} dx. \tag{2.32}$$

where $\mu(x)$ and $\nu(x)$ are the probability density functions of the two distributions. However, the KL divergence is not a true distance metric because it is not symmetric (i.e., $D_{KL}(\mu, \nu) \neq D_{KL}(\nu, \mu)$) and does not satisfy the triangle inequality. The KL divergence is also not defined if there exists an x where $\mu(x) > 0$ and $\nu(x) = 0$.

The Wasserstein distance, also known as the Earth Mover's distance (EMD), which is motivated by the optimal transport problem, is a metric for measuring the distance between two probability distributions that satisfies both the symmetry and the triangle inequality properties. Intuitively, it can be interpreted as the minimum cost of moving and transforming a pile of dirt into another pile of dirt.

Push-Forward operation. Given two spaces $(\mathcal{X}, \mathcal{Y})$, a set of Radon measures $\mathcal{M}(\mathcal{X})$ on $\mathcal{X}$, and a continuous map $T : \mathcal{X} \to \mathcal{Y}$, the corresponding push-forward measure $\nu = T_{\#}\mu \in \mathcal{M}(\mathcal{Y})$ of some $\mu \in \mathcal{M}(\mathcal{X})$ is defined as follows:

$$\nu(B) = \mu(T^{-1}(B)), \tag{2.33}$$

where B is any measurable set in $\mathcal{Y}$.

Intuitively, the map T can be considered as a function to move a single point from a measure μ on $\mathcal{X}$ to another single point from a measure ν on $\mathcal{Y}$. The push-foward operator $T_{\#}$ is an extension of T that map the entire measure μ to ν.

Monge problem. The Monge problem is the original formulation of the optimal transport problem, which is named after Gaspard Monge, who first introduced the problem in 1781. It involves finding the most efficient way to transport a given amount of resources from a set of sources to a set of destinations. The optimal solution is called optimal transport plan or optimal transport map, which assigns each point in the source space

to a unique point in the destination space, such that the total transportation cost is minimized.

Given the source space $\mathcal{X}$, the destination space $\mathcal{Y}$, and two probability measures μ and ν on $\mathcal{X}$ and $\mathcal{Y}$ respectively, the Monge problem is defined as follows:

$$\inf_{T:T_{\#\mu=\nu}} \int_{\mathcal{X}} d(x, T(x)) d\mu(x), \tag{2.34}$$

where $d(x, y)$ is the distance metric between x and y, or the transportation cost function of moving a unit of mass from single point x to y.

Kantorovich relaxation. The former formulation of the optimal transport problem as the assignment problem in Equation 2.34 is a discrete problem, where a source point is assigned to a unique destination point, which makes it difficult to solve. Kantorovich (1960) proposes instead that the mass at any source point x can be potentially mapped to several points in the destination space $\mathcal{Y}$. Kantorovic relaxation reformulates the nature of the former optimal transport problem from a deterministic to a probabilistic manner. It involves introducing a coupling or a transport plan π whose marginals are μ and ν. The optimal transport plan π^* is the optimum of the problem as follows:

$$\pi^* = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} d(x, y) d\pi \tag{2.35}$$

Wasserstein distance. The Wasserstein distance is a special case of the optimal transport problem in Equation 2.35 where the distance $d(x, y) = [\|x - y\|^p]^{1/p}$. Compared to KL divergence, Wasserstein distance is the true distance metric for measuring the distance between two probability distributions that satisfies both the symmetry and the triangle inequality properties.

$$W(\mu, \nu) = \inf_{\gamma \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} [\|x - y\|^p]^{1/p} \, d\pi \tag{2.36}$$

where $\Pi(\mu, \nu)$ is the set of all joint distributions $\gamma(x, y)$ whose marginals are $\mu(x)$ and $\nu(y)$.

The optimization problem in Equation 2.36 is the primal form of the Wasserstein distance, which can be computed efficiently using the Sinkhorn algorithm (Cuturi, 2013) or the entropic regularization of optimal transport (Genevay et al., 2018).

Dual formulation of Wasserstein distance. The Kantorovich-Rubinstein theorem (Villani et al., 2009) states that the Wasserstein distance can be expressed as the supremum of the set of all 1-Lipschitz functions, which is defined as follows:

$$W(\mu, \nu) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mu}[f(x)] - \mathbb{E}_{y \sim \nu}[f(y)], \tag{2.37}$$

where $\|f\|_L \leq 1$ is the set of all 1-Lipschitz functions.

This formulation is called the dual form of the Wasserstein distance, which allows us to compute the Wasserstein distance efficiently by optimizing over a set of functions instead of optimizing over the set of all joint distributions.

Applications of Wasserstein distance. The Wasserstein distance has a wide range of applications. In economics, it has been used to model the allocation of resources and solve problems related to supply chain management and economic equilibrium (Galichon, 2016, Peyré et al., 2019, Fajgelbaum and Schaal, 2020). In computer vision and image processing, it has been employed for image registration (Haker et al., 2004), shape matching (Su et al., 2015), and texture mixing (Rabin et al., 2012). In machine learning, optimal transport has found applications in domain adaptation (Lee et al., 2019), generative modeling (Arjovsky et al., 2017, Gulrajani et al., 2017), and data alignment (Chen et al., 2020c). In the context of AML, the Wasserstein distance has been used to measure the perceptual similarity between the input image and the adversarial example (Wong et al., 2019b, Wu et al., 2020c). It is also used to measure the distance between the data distribution and the adversarial distribution which is the foundation of distributional robustness (Staib and Jegelka, 2017, Sinha et al., 2017) as presented in the next section.

2.7.2 Distributional Robustness

Distributional robustness (DR) is an emerging framework for learning and decision-making under uncertainty, which seeks the worst-case expected loss among a ball of distributions, containing all distributions that are close to the empirical distribution (Gao et al., 2017). As the Wasserstein distance is a powerful and convenient tool of measuring closeness between distributions, Wasserstein DR has been one of the most widely-used variant of DR, which has rich applications in (semi)-supervised learning (Blanchet and Kang, 2017, Chen and Paschalidis, 2018, Yang, 2020), transfer learning and domain adaptation (Lee and Raginsky, 2017, Duchi et al., 2019, Zhao et al., 2019), and reinforcement learning (Abdullah et al., 2019, Smirnova et al., 2019, Derman and Mannor, 2020). In this section, we introduce the concept of distributional robustness that lay the foundation for our proposed methods in chapter 5.

Here we consider a generic Polish space S endowed with a distribution $\mathbb{Q}$. Let $f : S \to \mathbb{R}$ be a real-valued (risk) function and $c : S \times S \to \mathbb{R}_+$ be a cost function. Distributional robustness setting aims to find the distribution $\tilde{\mathbb{Q}}$ in the vicinity of $\mathbb{Q}$ and maximizes the risk in the $\mathbb{E}$ form (Sinha et al., 2017, Blanchet and Murthy, 2019):

$$\min_{\tilde{\mathbb{Q}}:\mathcal{W}_c\left(\tilde{\mathbb{Q}},\mathbb{Q}\right)<\epsilon} \mathbb{E}_{\tilde{\mathbb{Q}}}\left[f\left(z\right)\right], \tag{2.38}$$

where $\epsilon > 0$ and $\mathcal{W}_c$ denotes the optimal transport (OT) cost, or a Wasserstein distance if c is a metric, defined as:

$$\mathcal{W}_c\left(\tilde{\mathbb{Q}}, \mathbb{Q}\right) := \inf_{\gamma \in \Gamma\left(\tilde{\mathbb{Q}},\mathbb{Q}\right)} \int c d\gamma, \tag{2.39}$$

where $\Gamma\left(\tilde{\mathbb{Q}}, \mathbb{Q}\right)$ is the set of couplings whose marginals are $\tilde{\mathbb{Q}}$ and $\mathbb{Q}$.

With the assumption that $f \in L^1\left(\mathbb{Q}\right)$ is upper semi-continuous and the cost c is a non-negative lower semi-continuous satisfying $c(z, z') = 0$ iff $z = z'$, (Sinha et al., 2017, Blanchet and Murthy, 2019) show that the *dual* form for Eq. (2.38) is:

$$\min_{\lambda \geq 0}\left\{\lambda\epsilon + \mathbb{E}_{z\sim\mathbb{Q}}[\max_{z'}\left\{f\left(z'\right) - \lambda c\left(z', z\right)\right\}]\right\}. \tag{2.40}$$

Sinha et al. (2017) further employs a Lagrangian for Wasserstein-based uncertainty sets to arrive at a relaxed version with $\lambda \geq 0$:

$$\max_{\tilde{\mathbb{Q}}} \left\{ \mathbb{E}_{\tilde{\mathbb{Q}}} \left[f\left(z\right) \right] - \lambda \mathcal{W}_c \left(\tilde{\mathbb{Q}}, \mathbb{Q} \right) \right\} = \mathbb{E}_{z \sim \mathbb{Q}} [\max_{z'} \left\{ f\left(z'\right) - \lambda c\left(z', z\right) \right\}]. \tag{2.41}$$

Chapter 3

Representation Learning Approaches to Adversarial Robustness

3.1 Introduction

In this chapter, we present our novel contributions towards improving adversarial robustness of models through representation learning in the two papers (Bui et al., 2020) and Bui et al. (2021a). The central theme of our works was based on the idea that a robust representation should capture both local and global information of the data manifold, which is critical for enhancing the resilience of models against adversarial attacks. In particular, the representations of examples within a small, local neighborhood of each data point should exhibit proximity, a quality that contributes significantly to enhanced adversarial robustness. However, it's crucial to strike a balance, as excessive local compactness can inadvertently lead to unfavorable outcomes. For instance, representations of examples from the same class may be pushed apart, or, more concerning, those from distinct classes may converge, which runs counter to our goals. Hence, it is imperative to also incorporate the global compactness property to ensure a comprehensive representation of the data space within the latent space.

In the work Bui et al. (2020), we proposed a regularization method that imposes the local and global compactness properties on the intermediate representations. Specifically,

by explicitly strengthening local compactness, we enforce the intermediate representations of a benign example and its adversarial examples to be as proximal as possible. However, enforcing the local compactness itself was not be sufficient to guarantee a robust defense model as the representations might be encouraged to globally spread out in the intermediate space, significantly hurting accuracies on both benign and adversarial examples. To address this issue, we further proposed to impose global compactness to encourage the representations of examples in the same class to be proximal yet those in different classes to be more distant. We empirically showed that our proposed method could significantly improve adversarial robustness of the model while maintaining the generalization ability.

When we first published our work, we were the pioneers in considering both global and local information in learning robust representations. However, our work was limited by our lack of understanding of the importance of the distance metric in the latent space. Contrastive learning (Chen et al., 2020a), on the other hand, has become increasingly popular as an effective self-supervised representation learning approach. At the center of contrastive learning is the contrastive loss that encourages the representations of an anchor example and its positive examples to be proximal while those of the anchor example and its negative examples to be distant. While having a similar principle to our previous work, the contrastive loss considers the relative distance between the representations in the latent space which interpret the global information better than the absolute distance as in our previous work.

In the work Bui et al. (2021a), we proposed to use the contrastive loss to learn a robust representation. Intuitively, as the divergence in the latent space is the focus of both robustness learning and contrastive learning, it is natural to leverage contrastive learning to improve the robustness. However, we demonstrated that directly adopting CL into AML can hardly improve adversarial robustness, indicating that a deeper understanding of the relationships between the CL mechanism, latent space compactness, and adversarial robustness is required. Pursuing this comprehension, we tried to answer three research questions: why can CL help to improve adversarial robustness, how to integrate CL with adversarial training in the context of AML, and what are the important factors that affect the performance of CL in AML? To this end, we proposed *Adversarial Supervised Contrastive Learning (ASCL)* framework that combines adversarial training and contrastive learning to learn robust representations. Moreover, we

developed a novel series of strategies for selecting positive and negative samples which judiciously picks the most relevant samples of the anchor that help to further improve adversarial robustness. We empirically showed that our ASCL framework could outperform several adversarial training methods including our previous work (Bui et al., 2020) by a large margin and achieve comparable performance with the state-of-the-art robust defenses in the literature.

Finally, while not included in this thesis, in the work Le et al. (2022), we completed the understanding of the relationships between the latent divergence and adversarial robustness by proposing a novel game theory framework of two players attackers and defenders. More specifically, we developed attack and defense-guaranteed bounds that can be meaningfully and intuitively interpreted from the perspective of both attacks and defenses. Technically, the lower bound, which is useful for the attack side, reveals that to attack more efficiently, adversaries need to globally push the representations of adversarial examples to be more intermingled. On the other hand, the upper bound, which is useful for the defense side, shows that to defend more effectively, defenders need to keep the representations of adversarial examples as close to those of the benign examples as possible. Our theory aligns with the empirical results of our previous work (Bui et al., 2021a) and (Bui et al., 2020), providing further insights into the complex interplay between representation learning, adversarial attacks, and defenses. Although not included in this thesis, our proposed game theory framework is a promising direction for future research in developing more robust defenses against adversarial attacks.

The major content of this chapter is in the following attached papers:

- **Anh Bui**, Trung Le, He Zhao, Paul Montague, Olivier de Vel, Tamas Abraham, Dinh Phung, *"Improving Adversarial Robustness by Enforcing Local and Global Compactness"*. In Proceedings of the European Conference on Computer Vision (ECCV) 2020.

  The code of this paper is released at https://github.com/tuananhbui89/Adversarial-Divergence-Reduction.

- **Anh Bui**, Trung Le, He Zhao, Paul Montague, Sayit Camtepe, Dinh Phung, *"Understanding and Achieving Efficient Robustness with Adversarial Supervised Contrastive Learning"*, Submitted to Neurocomputing, 2023.

The code of this paper is released at https://github.com/tuananhbui89/ASCL.

3.2 Related Work

Unlike an input space $\boldsymbol{X}$, a latent space $\boldsymbol{Z}$ has a lower dimensionality and a higher mutual information with the prediction space than the input one $I(Z,Y) \geq I(X,Y)$ (Tishby and Zaslavsky, 2015). Therefore, defense with the latent space has particular characteristics to deal with adversarial attacks notably (Samangouei et al., 2018, Mao et al., 2019, Bui et al., 2020, Zhang and Wang, 2019, Xie et al., 2019).

3.2.1 Modeling the data manifold

Ilyas et al. (2019) made a hypothesis that adversarial vulnerability is the result of the sensitivity of deep learning models to well-generalizable but imperceptible-to-human features. This phenomenon was later analyzed as the phenomenon of learning a shortcut in deep models (Geirhos et al., 2020). Based on this hypothesis, the authors proposed an ideal framework that learns from useful and robust features only that can achieve both robustness and generalization. However, to achieve this goal, the authors proposed a method to disentangle the robust/non-robust features relying on a pre-trained robust model which is limited by the robustness of the pre-trained model.

Samangouei et al. (2018) proposed a GAN-based method to model the data manifold and then use the learned generator to approximate the input sample. During the training of the GAN, the generator is encouraged to resemble the training data. It is therefore expected to eliminate the adversarial effect on input samples. At test time, prior to feeding an image to the classifier, it was projected to the range of the generator by finding the latent code that minimizes the distance between the input and the generator output. The substituted image which was eliminate adversarial perturbation was then fed to the classifier. In this way, the gradient-based attacks could not find the adversarial examples because of the non-differentiability of the generator. However, later Jalal et al. (2017) proposed an overpowered attack method to efficiently attack these kinds of non-differentiable-based defenses.

Stutz et al. (2019) found that the regular adversarial examples leave the manifold of benign data, which explains the drop of generalization when using these adversarial examples in adversarial training. The authors proposed a VAE-GAN architecture to approximate the data manifold and based on that, they proposed an on-manifold adversarial generation by using a pre-trained encoder-decoder. More specifically, they perturb the latent representation of the benign input getting by the encoder and then use the decoder to generate the adversarial example. However, this method solely relies on the assumption that the data manifold is well approximated by a VAE-GAN architecture which is not feasible in real-world applications.

3.2.2 Regularization on latent space

The aforementioned works can be categorized as direct robust representation learning methods, as they focus on directly modeling the robust representations of the data. Conversely, there are other approaches that indirectly learn robust representations by incorporating additional regularizations that introduce inductive bias properties in the latent space. Empirical evidence suggests that these indirect methods tend to be more effective.

Metric learning (Mao et al., 2019), which discovered a shift in the representations of adversarial examples towards a cluster associated with a false class. To address this issue, they proposed minimizing the distance between the representations of adversarial examples and those of benign examples, using a triplet loss formulation.

Incorporating Self-Supervised Contrastive Learning (SSCL) (Chen et al., 2020a), (Jiang et al., 2020) and Kim et al. (2020) focused on learning unsupervised robust representations to enhance robustness in unsupervised and semi-supervised settings. Both methods introduced an adversary that maximizes the InfoNCE loss, instead of relying on standard cross-entropy loss (Goodfellow et al., 2015) or KL divergence (Zhang et al., 2019). Through adversarial pre-training with these generated adversarial examples, the encoder becomes resilient against instance-wise attacks and achieves comparable robustness to supervised adversarial training, as reported in Kim et al. (2020). In contrast, Jiang et al. (2020) proposed three pre-training options, with their best method utilizing two adversarial examples that come with a higher computational cost to generate.

3.3 Adversarial Divergence Reduction

Improving Adversarial Robustness by Enforcing Local and Global Compactness

Anh Bui[1][0000−0003−4123−2628], Trung Le[1][0000−0003−0414−9067], He Zhao[1][0000−0003−0894−2265], Paul Montague[2][0000−0001−9461−7471], Olivier deVel[2][0000−0001−5179−3707], Tamas Abraham[2][0000−0003−2466−7646], and Dinh Phung[1][0000−0002−9977−8247]

[1] Monash University, Australia
{tuananh.bui,trunglm,ethan.zhao,dinh.phung}@monash.edu
[2] Defence Science and Technology Group, Australia
{paul.montague,olivier.devel,tamas.abraham}@dst.defence.gov.au

Abstract. The fact that deep neural networks are susceptible to crafted perturbations severely impacts the use of deep learning in certain domains of application. Among many developed defense models against such attacks, adversarial training emerges as the most successful method that consistently resists a wide range of attacks. In this work, based on an observation from a previous study that the representations of a clean data example and its adversarial examples become more divergent in higher layers of a deep neural net, we propose the Adversary Divergence Reduction Network which enforces local/global compactness and the clustering assumption over an intermediate layer of a deep neural network. We conduct comprehensive experiments to understand the isolating behavior of each component (i.e., local/global compactness and the clustering assumption) and compare our proposed model with state-of-the-art adversarial training methods. The experimental results demonstrate that augmenting adversarial training with our proposed components can further improve the robustness of the network, leading to higher unperturbed and adversarial predictive performances.

Keywords: Adversarial Robustness, Local Compactness, Global Compactness, Clustering assumption

1 Introduction

Despite the great success of deep neural nets, they are reported to be susceptible to crafted perturbations [25, 6], even state-of-the-art ones. Accordingly, many defense models have been developed, notably [17, 27, 26, 20]. Recently, the work of [1] undertakes an in-depth study of neural network defense models and conduct comprehensive experiments on a complete suite of defense techniques, which has lead to postulating one common reason why many defenses provide apparent robustness against gradient-based attacks, namely *obfuscated gradients*.

According to the above study, adversarial training with Projected Gradient Descent (PGD) [17] is one of the most successful and widely-used defense

techniques that remained consistently resilient against attacks, which has inspired many recent advances including Adversarial Logit Pairing (ALP) [11], Feature Denoising [26], Defensive Quantization [15], Jacobian Regularization [9], Stochastic Activation Pruning [5], and Adversarial Training Free [22].

In this paper, we propose to build robust classifiers against adversarial examples by learning better representations in the intermediate space. Given an image classifier based on a multi-layer neural net, conceptually, we divide the network into two parts with an intermediate layer: the generator network from the input layer to the intermediate layer and the classifier network from the intermediate layer to the output prediction layer. The output of the generator network (i.e., the intermediate layer) is the intermediate representation of the input image, which is fed to the classifier network to make prediction. For image classifiers, an adversarial example is usually generated by adding small perturbations to a clean image. The adversarial example may look very similar to the original image but leads to significant changes to the prediction of the classifier. It has been observed that in deep neural networks, the representations of a clean data example and its adversarial example might become very diverge in the intermediate space, although their representations are proximal in the data space [26]. Due to the above divergence in the intermediate space, a classifier may be hard to predict the same class of the adversarial and real images. Inspired by this observation, we propose to learn better representations that reduce the above divergence in the intermediate space, so as to enhance the classifier robustness against adversarial examples.

In particular, we propose an enhanced adversarial training framework that imposes the *local and global compactness* properties on the intermediate representations, to build more robust classifiers against adversarial examples. Specifically, by explicitly strengthening local compactness, we enforce the intermediate representations output from the generator of a clean image and its adversarial examples to be as proximal as possible. In this way, the classifier network is less easy to be misled by the adversarial examples. However, enforcing the local compactness itself may not be sufficient to guarantee a robust defense model as the representations might be encouraged to globally spread out in the intermediate space, significantly hurting accuracies on both clean and adversarial images. To address this, we further propose to impose global compactness to encourage the representations of examples in the same class to be proximal yet those in different classes to be more distant. Finally, to increase the generalization capacity of the deep network and reduce the misclassification of adversarial examples, our framework enjoys the flexibility to incorporate the clustering assumption [3], which aims to force the decision boundary of a classifier to lie in the gap between clusters of different classes. By collaboratively incorporating the above three properties, we are able to learn better intermediate representations, which help to boost the adversarial robustness of classifiers. Intuitively, we name our proposed framework to the *Adversary Divergence Reduction Network* (ADR).

To comprehensively exam the proposed framework, we conduct extensive experiments to investigate the influence of each component (i.e., local/global com-

pactness and the clustering assumption), visualize the smoothness of the loss surface of our robust model, and compare our proposed ADR method with several state-of-the-art adversarial defenses. The experimental results consistently show that our proposed method can further improve over others in terms of better adversarial and clean predictive performances. The contributions of this work are summarized as follows:

- We propose the local and global compactness properties on the intermediate space to enforce the better representations, which lead to more robust classifiers;
- We incorporate our local and global compactness with clustering assumption to further enhance adversarial robustness;
- We plug the above three components into an adversarial training framework to introduce our Adversary Divergence Reduction Network;
- We extensively analyze the proposed framework and compare it with state-of-the-art adversarial training methods to verify its effectiveness.

2 Related works

Adversarial training defense Adversarial training can be traced back to [6], in which models were challenged by producing adversarial examples and incorporating them into training data. The adversarial examples could be the worst-case examples (i.e., $x_a \triangleq \text{argmax}_{x' \in B_\varepsilon(x)} \ell(x', y, \theta)$) [6] or most divergent examples (i.e., $x_a \triangleq \text{argmax}_{x' \in B_\varepsilon(x)} D_{KL}(h_\theta(x') \,||\, h_\theta(x))$) [27] where D_{KL} is the Kullback-Leibler divergence and h_θ is the current model. The quality of the adversarial training defense crucially depends on the strength of the injected adversarial examples – e.g., training on non-iterative adversarial examples obtained from FGSM or Rand FGSM (a variant of FGSM where the initial point is randomised) are not robust to iterative attacks, for example PGD [17] or BIM [13].

Although many defense models were broken by [1], the adversarial training with PGD [17] was among the few that were resilient against attacks. Many defense models were developed based on adversarial examples from a PGD attack or attempts made to improve and scale up the PGD adversarial training. Notable examples include Adversarial Logit Pairing (ALP) [11], Feature Denoising [26], Defensive Quantization [15], Jacobian Regularization [9], Stochastic Activation Pruning [5], and Adversarial Training for Free [22].

Defense with a latent space These works utilized a latent space to enable adversarial defense, notably [10]. DefenseGAN [21] and PixelDefense [24] use a generator (i.e., a pretrained WS-GAN [7] for DefenseGAN and a PixelCNN [19] for PixelDefense) together with the latent space to find a denoised version of an adversarial example on the data manifold. These works were criticized by [1] as being easy to attack and impossible to work within the case of the CIFAR-10 dataset. Jalal et al. [10] proposed an overpowered attack method to

efficiently attack both DefenseGAN and PixelDefense and subsequently injected those adversarial examples to train the model. Though that work was proven to work well with simple datasets including MNIST and CelebA, no experiments were conducted on more complex datasets including, for example, CIFAR-10.

3 Proposed method

In what follows, we present our proposed method, named the *Adversary Divergence Reduction Network* (ADR). As shown in the previous study [26], although an adversarial example x_a and its corresponding clean example x are in close proximity in the data space (i.e., differ by a small perturbation), when brought forward up to the higher layers in a deep neural network, their representations become markedly more divergent, hence causing different prediction results. Inspired by this observation, we propose imposing local compactness for those representations in an intermediate layer of a neural network. The key idea is to enforce that the representations of an adversarial example and its clean counterpart be as proximal as possible, hence reducing the chance of misclassifying them. Moreover, we observe that enforcing the local compactness itself is not sufficient to guarantee a robust defense model as this enforcement might encourage representations to globally spread out across the intermediate space (i.e., the space induced by the intermediate representations), significantly hurting both adversarial and clean performances. To address this, we propose to impose global compactness for the intermediate representations such that representations of examples that belong in the same class are proximal and those in different classes are more distant. Finally, to increase the generalization capacity of the deep network and reduce the misclassification of adversarial examples, we propose to apply the clustering assumption [3] which aims to force the decision boundary to lie in the gap between clusters of different classes, hence increasing the chance for adversarial examples to be correctly classified.

3.1 Local compactness

Local compactness, which aims to reduce the divergence between the representations of an adversarial example and its clean example in an intermediate layer, is one of the key aspects of our proposed method. Let us denote our deep neural network by $h_\theta(\cdot)$, which decomposes into $h_\theta(\cdot) = g_\theta(f_\theta(\cdot))$ where the first (generator) network f_θ maps the data examples onto an intermediate layer where we enforce the compactness constraints. The following (classifier) network g_θ maps the intermediate representations to the prediction output. For local compactness, given a clean data example x, denote $\mathcal{A}_\varepsilon$ as a stochastic adversary that renders adversarial examples for x as $x_a \sim \mathcal{A}_\varepsilon(x)$ in a ball $B_\varepsilon(x) = \{x' : \|x - x'\| < \varepsilon\}$, our aim is to compress the representations of x and x_a in the intermediate layer by minimizing

$$\mathcal{L}_{\text{com}}^{\text{lc}} = \mathbb{E}_{x \sim \mathcal{D}_x}\left[\mathbb{E}_{x_a \sim \mathcal{A}_\varepsilon(x)}\left[\|f_\theta(x) - f_\theta(x_a)\|_p\right]\right] \tag{1}$$

where we use $\mathcal{D}_x = \{x_1, ..., x_N\}$ to represent both training examples and the corresponding empirical distribution and $\|\cdot\|_p$ with $p = 1, 2, \infty$ to specify the p-norm.

3.2 Global compactness

For global compactness, we want the representations of data examples in the same class to be closer and data examples in different classes to be more separate. As demonstrated later, global compactness in conjunction with the clustering assumption helps increase the margin of a data example (i.e., the distance from that data example to the decision boundary), hence boosting the generalization capacity of the classifier network and adversarial robustness.

More specifically, given two examples (x_i, y_i) and (x_j, y_j) drawn from the empirical distribution over $\mathcal{D}_{x,y} = \{(x_1, y_1), ..., (x_N, y_N)\}$ where the label $y_k \in \{1, 2, ..., M\}$, we compute the weight w_{ij} for this pair as follows:

$$w_{ij} = \frac{\alpha - \mathbb{I}_{y_i \neq y_j}}{\alpha} = \begin{cases} 1 & \text{if } y_i = y_j \\ \frac{\alpha - 1}{\alpha} & \text{otherwise} \end{cases} \tag{2}$$

where $\mathbb{I}_S$ is the indicator function which returns 1 if S holds and 0 otherwise. We consider $\alpha \in (0, 1)$, yielding $w_{ij} < 0$ if $y_i \neq y_j$ and $w_{ij} > 0$ if otherwise.

We enforce global compactness by minimizing

$$\mathcal{L}_{\text{com}}^{\text{gb}} = \mathbb{E}_{(x_i, y_i), (x_j, y_j) \sim \mathcal{D}_{x,y}} \left[w_{ij} \left\| f_\theta(x_i) - f_\theta(x_j) \right\|_p \right] \tag{3}$$

where we overload the notation $\mathcal{D}_{x,y}$ to represent the empirical distribution over the training set, which implies that the intermediate representations $f_\theta(x_i)$ and $f_\theta(x_j)$ are encouraged to be closer if $y_i = y_j$ and to be separate if $y_i \neq y_j$ for a global compact representation.

Note that in our experiment, we set $\alpha = 0.99$, yielding $w_{ij} \in \{1, -0.01\}$, and calculate global compactness with each random minibatch.

3.3 Clustering assumption and label supervision

At this stage, we have achieved compact intermediate representations for the clean data and adversarial examples obtained from a stochastic adversary $\mathcal{A}_\varepsilon$. Our next step is to enforce some constraints on the subsequent classifier network g_θ to further exploit this compact representation for improving adversarial robustness. The first constraint we impose on the classifier network g_θ is that this should classify both clean data and adversarial examples correctly by minimizing

$$\mathcal{L}_{\text{c}} = \mathbb{E}_{(x,y) \sim D_{x,y}} \left[\mathbb{E}_{x_a \sim \mathcal{A}_\varepsilon(x)} \left[\ell \left(h_\theta \left(x_a \right), y \right) \right] + \ell \left(h_\theta \left(x \right), y \right) \right] \tag{4}$$

where ℓ is the cross-entropy loss function.

In addition to this label supervision, the second constraint we impose on the classifier network g_θ is the clustering assumption [3], which states that the decision boundary of g_θ in the intermediate space should not break into any high density region (or cluster) of data representations in the intermediate space, forcing the boundary to lie in gaps formed by those clusters. The clustering assumption when combined with the global compact representation property should increase the data example margin (i.e., the distance from that data example to the decision boundary). If this is further combined with the fact that the representations of adversarial examples are compressed into the representation of its clean data example (*i.e.* local compactness) this should also reduce the chance that adversarial examples are misclassified. To enforce the clustering assumption, inspired by [23], we encourage the classifier confidence by minimizing the conditional entropy and maintain classifier smoothness using Virtual Adversarial Training (VAT) [18], respectively:

$$\mathcal{L}_{\text{conf}} = \mathbb{E}_{x\sim D_x}\left[\mathbb{E}_{x_a\sim\mathcal{A}_\varepsilon(x)}\left[-h_\theta(x_a)^T \log h_\theta(x_a)\right] - h_\theta(x)^T \log h_\theta(x)\right] \tag{5}$$

$$\mathcal{L}_{\text{smt}} = \mathbb{E}_{x\sim D_x}\left[\mathbb{E}_{x_a\sim\mathcal{A}_\varepsilon(x)}\left[D_{KL}\left(h_\theta(x)\|h_\theta(x_a)\right)\right]\right] \tag{6}$$

3.4 Generating adversarial examples

We can use any adversarial attack algorithm to define the adversary $\mathcal{A}_\varepsilon$. For example, Madry et al. [17] proposed to find the worst-case examples $x_a \triangleq \text{argmax}_{x'\in B_\varepsilon(x)}\ell\left(x', y, \theta\right)$ using PGD, while Zhang et al. [27] aimed to find the most divergent examples $x_a \triangleq \text{argmax}_{x'\in B_\varepsilon(x)} D_{KL}\left(h_\theta\left(x'\right) \,||\, h_\theta\left(x\right)\right)$. By enforcing local/global compactness over the adversarial examples obtained by $\mathcal{A}_\varepsilon$, we make them easier to be trained with the label supervision loss in Eq. (4), hence eventually improving adversarial robustness. The quality of adversarial examples obviously affects to the overall performance, however, in the experimental section, we empirically prove that our proposed components can boost the robustness of the adversarial training frameworks of interest.

3.5 Putting it all together

We combine the relevant terms regarding local/global compactness, label supervision, and the clustering assumption and arrive at the following optimization problem:

$$\min_\theta \mathcal{L} \triangleq \mathcal{L}_{\text{c}} + \lambda^{\text{lc}}_{\text{com}}\mathcal{L}^{\text{lc}}_{\text{com}} + \lambda^{\text{gb}}_{\text{com}}\mathcal{L}^{\text{gb}}_{\text{com}} + \lambda_{\text{conf}}\mathcal{L}_{\text{conf}} + \lambda_{\text{smt}}\mathcal{L}_{\text{smt}} \tag{7}$$

where $\lambda^{\text{lc}}_{\text{com}}, \lambda^{\text{gb}}_{\text{com}}, \lambda_{\text{conf}},$ and λ_{smt} are non-negative trade-off parameters.

In Figure 1, we illustrate how the three components, namely local/global compactness, label supervision, and the clustering assumption can mutually collaborate to improve adversarial robustness. The representations of data examples

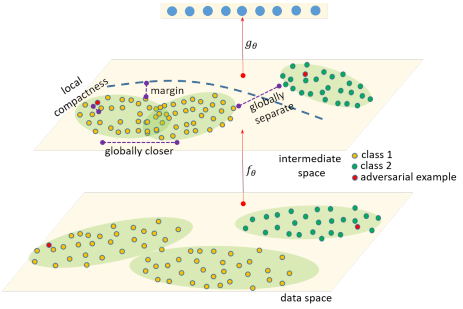


Fig. 1. Overview of Adversary Divergence Reduction Network. The local/global compactness and clustering assumption are intended to improve adversarial robustness.

via the network f_θ are enforced to be locally/globally compact, whereas the position of the decision boundary of the classifier network g_θ in the intermediate space is enforced using the clustering assumption. Ideally, with the clustering assumption, the decision boundary of g_θ preserves the cluster structure in the intermediate space and when combined with label supervision training ensures clusters in a class remain completely inside the decision region for this class. Moreover, global compactness encourages clusters of a class to be closer and those of different classes to be more separate. As a result, the decision boundary of g_θ lies in the gaps among clusters as well as with a sufficiently large margin for the data examples. Finally, local compactness requires adversarial examples to stay closer to their corresponding clean data example, hence reducing the chance of misclassifying them and therefore improving adversarial robustness.

Comparison with the contrastive learning. Interestingly, the contrastive learning [4, 8] and our proposed method aim to learn better representations by the principle of enforcing similar elements to be equal and dissimilar elements to be different. However, the contrastive learning works on an instance level, which enforces the representation of an image to be proximal with those of its transformations and to be distant with those of any other images. On the other hand, our method works on a class level, which enforces the intermediate representations of each class to be compact and well separated with those in other classes. Therefore, our method and the contrastive learning complement each other and intuitively improve both visual representation and adversarial robustness when combining together.

4 Experiments

In this section, we first introduce the general setting for our experiments regarding datasets, model architecture, optimization scheduler, and adversary attackers. Second, we compare our method with adversarial training with PGD, namely ADV [17] and TRADES [27]. We employ either ADV or TRADES as

the stochastic adversary $\mathcal{A}$ for our ADR and demonstrate that, when enhanced with local/global compactness and the clustering assumption, we can improve these state-of-the-art adversarial training methods.

Specifically, we begin this section with an ablation study to investigate the model behaviors and the influence of each component, namely local compactness, global compactness, and the clustering assumption, on adversarial performance. In addition, we visualize the smoothness of the loss surface of our model to understand why it can defend well. Finally, we undertake experiments on the MNIST and CIFAR-10 datasets to compare our ADR with both ADV and TRADES.

4.1 Experimental setting

General setting We undertook experiments on both the MNIST [14] and CIFAR-10 [12] datasets. The inputs were normalized to $[0, 1]$. For the CIFAR-10 dataset, we apply random crops and random flips as describe in [17] during training. For the MNIST dataset, we used the standard CNN architecture with three convolution layers and three fully connected layers described in [2]. For the CIFAR-10 dataset, we used two architectures in which one is the standard CNN architecture described in [2] and another is the ResNet architecture used in [17]. We note that there is a serious overfitting problem on the CIFAR-10 dataset as mentioned in [2]. In our setting, with the standard CNN architecture, we eventually obtained a 98% training accuracy, but only a 75% testing accuracy. With the ResNet architecture, we used the strategy from [17] to adjust the learning rate when training to reduce the gap between the training and validation accuracies. For the MNIST dataset, a drop-rate equal to 0.1 at epochs 55, 75, and 90 without weight decay was employed. For the CIFAR-10 dataset, the drop-rate was set to 0.1 at epochs 100 and 150 with weight decay equal to 2×10^{-4}. We use a momentum-based SGD optimizer for the training of the standard CNN for the MNIST dataset and the ResNet for the CIFAR-10 dataset, while using the Adam optimizer for training the standard CNN on the CIFAR-10 one. The hyperparameters setting can be found in the supplementary material.

Choosing the intermediate layer. The intermediate layer for enforcing compactness constraints immediately follows on from the last convolution layer for the standard CNN architecture and from the penultimate layer for the ResNet architecture. Moreover, we provide an additional ablation study to investigate the importance of choosing the intermediate layer which can be found in the supplementary material.

Attack methods We use PGD to challenge the defense methods in this paper. Specifically, the setting for the MNIST dataset is PGD-40 (i.e., PGD with 40 steps) with the distortion bound ε increasing from 0.1 to 0.7 and step size $\eta \in \{0.01, 0.02\}$, while that for CIFAR-10 is PGD-20 with ε increasing from 0.0039 ($\approx 1/255$) to 0.11 ($\approx 28/255$) and step size $\eta \in \{0.0039, 0.007\}$. The distortion metric is l_∞ for all attacks. For the adversarial training, we use $k = 10$ for CIFAR10 and $k = 20$ for MNIST for all defense methods.

Non-targeted and multi-targeted attack scenarios We used two types of attack scenarios, namely non-targeted and multi-targeted attacks. The non-targeted attack derives adversarial examples by maximizing the loss w.r.t. its clean data label, whilst the multi-targeted attack is undertaken by performing simultaneously targeted attack for all possible data labels. The multi-targeted attack is considered to be successful if any individual targeted attack on each target label is successful. While the non-targeted attack considers only one direction of the gradient, the multi-targeted attack takes multi-directions of gradient into account, which guarantees to get better local optimum.

4.2 Experimental results

In this section, we first conduct an ablation study using the MNIST dataset in order to investigate how the different components (local compactness, global compactness, and the clustering assumption) contribute to adversarial robustness. We then conduct experiments on the MNIST and CIFAR-10 datasets to compare our proposed method with ADV and TRADES. Further evaluation can be found in the supplementary material.

Ablation study We first study how each proposed component contributes to adversarial robustness. We use adversarial training with PGD as the baseline model and experiment on the MNIST dataset. Recall that our method consists of three components: the local compactness loss $\mathcal{L}_{\text{com}}^{\text{lc}}$, the global compactness loss $\mathcal{L}_{\text{com}}^{\text{gb}}$, and the clustering assumption loss which combines $\{\mathcal{L}_{\text{smt}} + \mathcal{L}_{\text{conf}}\}$. In this experiment, we simply set the trade-off parameters $\lambda_{\text{com}}^{\text{lc}}, \lambda_{\text{com}}^{\text{gb}}, \lambda_{\text{smt}} = \lambda_{\text{conf}} = \lambda_{\text{ca}}$ to 0/1 to deactivate/activate the corresponding component. We consider two metrics: the natural accuracy (i.e., the clean accuracy) and the robustness accuracy to evaluate a defense method. The natural accuracy is that evaluated on the clean test images, while the robustness accuracy is that evaluated on adversarial examples generated by attacking the clean test images. It is noteworthy that for many existing defense methods, improving robustness accuracy usually harms natural accuracy. Therefore, our proposed method aims to reach a better trade-off between the two metrics.

Table 1 shows the results for the PGD attack with $k = 40$, $\varepsilon = 0.325$, and $\eta = 0.01$. We note that ADR-None is our base model without any additional components. The base model can be any adversarial training based method, e.g., ADV or TRADES. Without loss of generality, we use ADV as the base model, i.e., ADR-None. By gradually combining the proposed additional components with ADR-None we produce several variants of ADR (e.g., ADR+LC is ADR-None together with the local compactness component). Since the standard model was trained without any defense mechanism, its natural accuracy is high at 99.5% whereas the robustness accuracy is very poor at 0.88%, indicating its vulnerability to adversarial attacks. Regarding the variants of our proposed models,

those with additional components generally achieve higher robustness accuracies compared with ADR-None (i.e. ADV), without hurting the natural accuracy. In addition, the robust accuracy was significantly improved with global compactness and the clustering assumption terms.

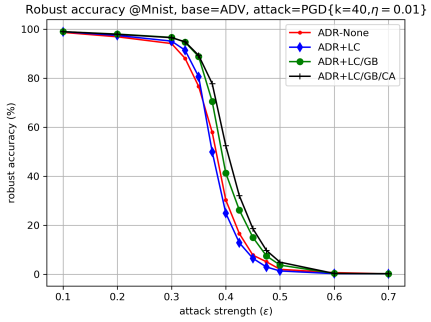


Fig. 2. Variation of the robustness accuracies under different attack strengths. The base model is ADV (ADR-None).

Table 1. Results of the PGD-40 attack on the MNIST dataset for the base ADV model together with its variants with the different components (LC = local compactness, GB = global compactness, CA = clustering assumption) and $\varepsilon = 0.325$.

	Nat. acc.	Rob. acc.
Standard model	99.5%	0.84%
ADR-None (ADV)[a]	99.27%	88.1%
ADR+LC	99.41%	91.43%
ADR+LC/GB	99.35%	94.52%
ADR+LC/GB/CA	99.36%	94.96%

[a] The performance of ADV is lower than that in [17] because of the difference of the attack strength and model architecture

We also evaluate the metrics of interest with different attack strength by increasing the distortion boundary ε as shown in Figure 2. By just adding a single local compactness component, our method can improve the base model (ADV or ADR-None) for attacks with strength $\varepsilon \leq 0.35$. By adding the global compactness component, our method can significantly improve over the base model, especially for stronger attacks. Recall that as we generate adversarial examples from the PGD attack with $k = 20, \varepsilon_d = 0.3, \eta = 0.01$ to train the defense models, is is unsurprised to see a model defends well with $\varepsilon \leq 0.3$. Interestingly, by adding our components, our defense methods can also achieve reasonably good robustness accuracy of 80%, even when ε varies from 0.34 to 0.37, indicating the better generality of our methods.

To gain a better understanding of the contribution of the local compactness component, we visualize the loss surface of the base model (ADV as ADR-None) and the base model with only the local compactness term (ADR+LC). In Figure 3, the left image is a clean data example x, while the middle image is the loss surface over the input region around x in which the z-axis indicates the cross-entropy loss w.r.t. the true label (the higher value means more incorrect prediction) and the x- and y-axis indicates the variance of the input image along the gradient direction w.r.t. x and a random orthogonal direction, respectively.

By varying along the two axes, we create a grid of images which represents the neighborhood region around x. The right-hand image depicts the predicted labels corresponding with this input grid.

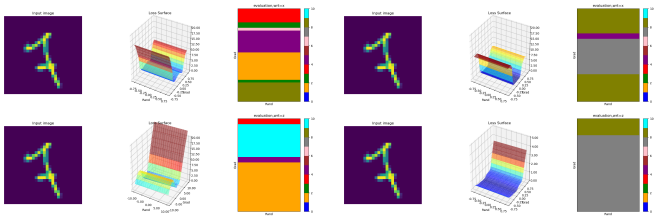


Fig. 3. Loss surface at local region of a clean data example. Top-left: ADR-None w.r.t input. Top-right: ADR+LC w.r.t input. Bottom-left: ADR-None w.r.t latent. Bottom-right: ADR+LC w.r.t latent

From Figure 3, for ADR-None, that its neighborhood region is non-smooth, resulting in incorrect predictions to the label 1 and 4. Meanwhile, for our ADR+LC method (adversarial training with local compactness), the loss surface w.r.t. the input is smoother in its neighborhood region, resulting in correct predictions. In addition, in our method, the prediction surface w.r.t. the latent feature in the intermediate representation layer is smoother than that w.r.t. input. This means that our local compactness makes the local region more compact, hence improving adversarial robustness. Visualization with an adversarial example as input can be found in our supplementary material which provides more evidence of our improvement over the base model.

Furthermore, we use t-SNE [16] to visualize the intermediate space for demonstrating the effect of our global compactness component. We choose to show a *positive adversarial example* defined as an adversarial example which successfully fools a defense method. We compare the base model (ADV as ADR-None) with our method with the compactness terms and use t-SNE to project clean data and adversarial examples onto 2D space as in Figure 4. For ADR-None, its adversarial examples seem to distribute more broadly and randomly. With our global compactness constraint, the adversarial examples look well-clustered in a low density region, while rarely present in the high density region of natural clean images. We leverage the entropy of the prediction probability of examples as the third dimension in Figure 5. A lower entropy mean that the prediction is more confident (i.e., closer to a one-hot vector) and vice versa. It can be observed that for the base model, the prediction outputs of adversarial examples seem to be randomly distributed, while for our ADR+LC/GB method, the prediction outputs of adversarial examples mainly lie in the high entropy region and are well-separated from those of the clean data examples. In other words, adversarial examples can be more easily detected from clean examples in our method,

according to the predication entropy. In addition, the visualization for a *negative adversarial example* can be found in our supplementary material.

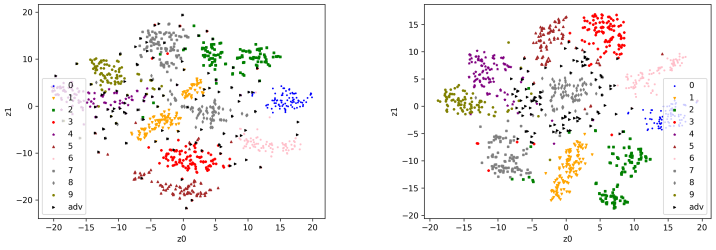


Fig. 4. T-SNE visualization of latent space. Black triangles are (positive) adversarial examples while others are clean images. Left: ADR-None. Right: ADR+LC/GB

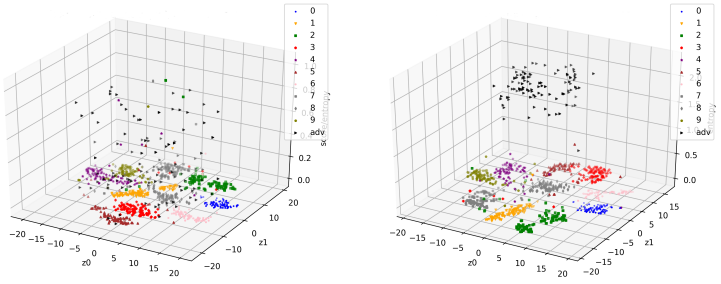


Fig. 5. T-SNE visualization of latent space with entropy of the prediction probability. Black triangles are (positive) adversarial examples while others are clean images. Left: ADR-None. Right: ADR+LC/GB

To summarize, in this ablation study, we have demonstrated how our proposed components can improve adversarial robustness. In the next section, we will compare the best variant (with all components) of our method with both ADV and TRADES on more complex datasets to highlight the capability of our method.

Experiment on the MNIST dataset We compare our method with ADV and TRADES on the MNIST dataset. For our method, in addition to using its full version with all of the proposed terms, we consider two variants ADR-ADV and ADR-TRADES wherein the adversary $\mathcal{A}$ is set to be ADV and TRADES respectively. We use PGD/TRADES generated adversarial examples with $k = 20, \varepsilon_d =$

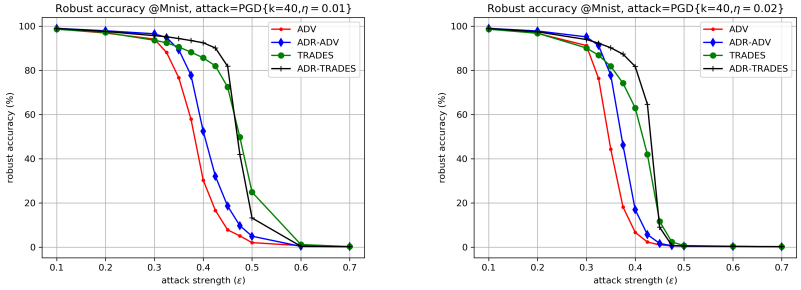


Fig. 6. Robust accuracy against PGD attack on MNIST. Base models include ADV and TRADES. Left: $\eta = 0.01$. Right: $\eta = 0.02$

$0.3, \eta_d = 0.01$ for adversarial training as proposed in [17] and employ the PGD attack with $k = 40$, using two iterative size $\eta \in \{0.01, 0.02\}$ and different distortion boundaries ε to attack. The results shown in Figure 6 illustrate that our variants outperform the baselines, especially for $\{\varepsilon = \varepsilon_d = 0.3, \eta = 0.01\}$. For example, our ADR-ADV improves ADV by 2.4% (from 94.15% to 96.55%) while ADR-TRADES boosts TRADES by 2.07% (from 93.64% to 95.71%). While for attack setting $\{\epsilon = \epsilon_d = 0.3, \eta = 0.02\}$, our method improves ADV and TRADES by 4.0% and 3.8% respectively. Moreover, the improvement gap increases when the attack goes stronger.

Experiment on the CIFAR-10 dataset We conduct experiments on the CIFAR-10 dataset under two different architectures: standard CNN from [2] and ResNet from [17]. We set $k = 10, \varepsilon_d = 0.031, \eta_d = 0.007$ for ADV and TRADES and use a PGD attack with $k = 20$, $\eta \in \{0.0039, 0.007\}$ and different distortion boundary ε. The results for standard CNN architecture in figure 7 show that our methods significantly improve over the baselines. Moreover, the results for standard CNN architecture at a checking point $\{\varepsilon = \varepsilon_d = 0.031, \eta = \eta_d = 0.007\}$ in Table 2 show that our methods significantly outperform their baselines in terms of natural and robust accuracies. Moreover, Figure 7 indicates that our proposed methods can defend better in a wide range of attack strength. Particularly, when with varied distortion boundary ε in $[0.02, 0.1]$, our proposed methods always produce better robust accuracies than its baselines. Finally, the results for ResNet architecture in Table 2 show a slight improvement of our methods comparing with ADV but around 2.5% improvement from TRADES on both Non-targeted and Multi-targeted attacks.[3] The quality of adversarial examples and the chosen network architecture obviously affects the overall performance,

[3] The performance of TRADES is influenced by the model architectures and parameter tunings. The works [20, 10] also reported that TRADES cannot surpass ADV all the time which explains the lower performance of TRADES on ResNet architecture in this paper. More analysis can be found in the supplementary material.

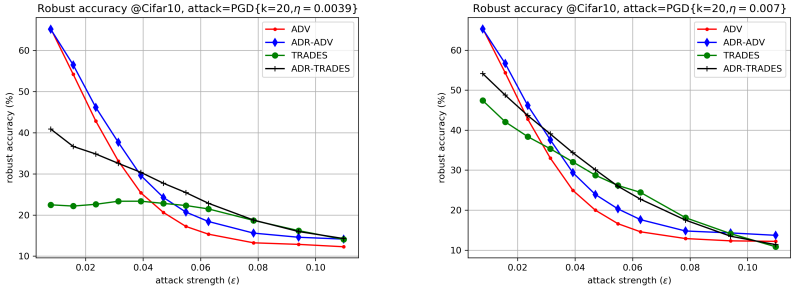


Fig. 7. Robust accuracy against PGD attack on CIFAR-10, using Standard CNN architecture. Base models include ADV and TRADES. Left: $\eta = 0.0039$. Right: $\eta = 0.007$.

however, in this experiment, we empirically prove that our proposed components can boost the robustness under different combinations of the adversarial training frameworks and network architectures.

Table 2. Robustness comparison on the CIFAR-10 dataset against PGD attack at $k = 20, \epsilon = 0.031, \eta = 0.007$ using Standard CNN and ResNet architectures

	Standard CNN			ResNet		
	Nat. acc.	Non-target	Mul-target	Nat. acc.	Non-target	Mul-target
Standard model	75.27%	12.26%	0.00%	92.51%	0.00%	0.00%
ADV	67.86%	33.12%	18.73%	78.84%	44.08%	41.20%
TRADES	71.37%	35.84%	18.01%	83.27%	37.52%	35.05%
ADR-ADV	69.09%	37.67%	22.58%	78.43%	44.72%	41.43%
ADR-TRADES	69.0%	39.68%	26.7%	82.02%	40.17%	37.70%

5 Conclusion

Previous studies have shown that adversarial training has been one of the few defense models resilient to various attack types against deep neural network models. In this paper, we have shown that by enforcing additional components, namely local/global compactness constraints together with the clustering assumption, we can further improve the state-of-the-art adversarial training models. We have undertaken comprehensive experiments to investigate the effect of each component and have demonstrated the capability of our proposed methods in enhancing adversarial robustness using real-world datasets.

Acknowledgement: This work was partially supported by the Australian Defence Science and Technology (DST) Group under the Next Generation Technology Fund (NTGF) scheme.

References

1. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
2. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)
3. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: AISTATS. vol. 2005, pp. 57–64 (2005)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020)
5. Dhillon, G.S., Azizzadenesheli, K., Lipton, Z.C., Bernstein, J., Kossaifi, J., Khanna, A., Anandkumar, A.: Stochastic activation pruning for robust adversarial defense. arXiv preprint arXiv:1803.01442 (2018)
6. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
7. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in neural information processing systems. pp. 5767–5777 (2017)
8. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
9. Jakubovitz, D., Giryes, R.: Improving dnn robustness to adversarial attacks using jacobian regularization. In: Proceedings of the European Conference on Computer Vision. pp. 514–529 (2018)
10. Jalal, A., Ilyas, A., Daskalakis, C., Dimakis, A.G.: The robust manifold defense: Adversarial training using generative models. arXiv preprint arXiv:1712.09196 (2017)
11. Kannan, H., Kurakin, A., Goodfellow, I.: Adversarial logit pairing. arXiv preprint arXiv:1803.06373 (2018)
12. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)
13. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
15. Lin, J., Gan, C., Han, S.: Defensive quantization: When efficiency meets robustness. arXiv preprint arXiv:1904.08444 (2019)
16. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
17. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
18. Miyato, T., Maeda, S., Koyama, M., Ishii, S.: Virtual adversarial training: A regularization method for supervised and semi-supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(8), 1979–1993 (Aug 2019)
19. Oord, A.v., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. arXiv preprint arXiv:1601.06759 (2016)
20. Qin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Fawzi, A., De, S., Stanforth, R., Kohli, P.: Adversarial robustness through local linearization. In: Advances in Neural Information Processing Systems. pp. 13824–13833 (2019)
21. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-gan: Protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605 (2018)

22. Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! In: Advances in Neural Information Processing Systems. pp. 3353–3364 (2019)
23. Shu, R., Bui, H.H., Narui, H., Ermon, S.: A dirt-t approach to unsupervised domain adaptation. arXiv preprint arXiv:1802.08735 (2018)
24. Song, Y., Kim, T., Nowozin, S., Ermon, S., Kushman, N.: Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. arXiv preprint arXiv:1710.10766 (2017)
25. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
26. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 501–509 (2019)
27. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. arXiv preprint arXiv:1901.08573 (2019)

Supplementary to "Improving Adversarial Robustness by Enforcing Local and Global Compactness"

1 Hyperparameters

The hyperparameters for our experiments as Table 1. The hyperparameters of local compactness, global compactness, and smoothness are set to be either 1 or 0, meaning they are switched ON/OFF. Although finer tuning of these parameters can lead to better results, our method outperforms the baselines in these initial settings, which demonstrates the effectiveness of those components.

Table 1. Hyper-parameter setting for the experiment section

	λ_{com}^{lc}	λ_{com}^{gb}	λ_{smt}	λ_{conf}
MNIST	1.	1.	1.	0.
CIFAR-10-CNN	1.	1.	1.	1.
CIFAR-10-ResNet	1.	1.	1.	0.

2 Model architectures and experimental setting

We summarize the experimental setting in Table 2.

For the MNIST dataset, we used the standard CNN architecture with three convolution layers and three fully connected layers described in [2]. For the CIFAR-10 dataset, we used two architectures in which one is the standard CNN architecture described in [2] and another is the ResNet architecture used in [6]. The ResNet architecture has 5 residual units with (16, 16, 32, 64) filters each. We choose the convolution layers as the Generator and the last fully connected layers as the Classifier for ResNet architecture. The standard CNN architectures are redescribed as follow:

CNN-4C3F(32) Generator: 2×Conv(32)→ Max Pooling→ 2×Conv(32)→ Max Pooling→ Flatten

CNN-4C3F(32) Classifier: FC(200)→ ReLU→ Dropout(0.5)→ FC(200)→ ReLU→ FC(10)→ Softmax

CNN-4C3F(64) Generator: 2×Conv(64)→ Max Pooling→ 2×Conv(64)→ Max Pooling→ Flatten

CNN-4C3F(64) Classifier: FC(256)→ ReLU→ Dropout(0.5)→ FC(256)→ ReLU→ FC(10)→ Softmax

3 Choosing the intermediate layer

The intermediate layer for enforcing compactness constraints immediately follows on from the generator. We additionally conduct an ablation study to investigate the importance of choosing the intermediate layer and report natural accuracy and robust accuracy against non-targeted/multiple-targeted attacks respectively. We use the standard CNN architecture (which has 4 Convolution layers in Generator and 3 FC layers in Classifier), with four additional variants corresponding to different choices of the intermediate layer (right after the generator). We use PGD ($k = 100, \epsilon = 0.3, \eta = 0.01$ for MNIST, $k = 100, \epsilon = 0.031, \eta = 0.007$ for CIFAR-10) to evaluate these models. It can be seen from the results as showing in Table 3 that the performance slightly downgrades if choosing shallower layers. The higher impact is expected on a larger architecture (i.e., Resnet), which can be investigated in future.

4 The performance of TRADES

TRADES aims to find the most divergent adversarial examples, while ADV aims to find the worst-case examples to improve a model (see Sec. 2.2 in our paper for more detail). Hence theoretically, there is no guarantee that TRADES outperforms ADV. In practice, the performance of TRADES is influenced by the classifier architectures and parameter tunings. The works [7,4] also reported that TRADES cannot surpass ADV all the time (Table 1 and footnote 8 in [7], Table 1 in [4]), which is in line with the findings in our paper.

5 Further experiments

We conduct an additional evaluation with further state-of-the-art attack methods (e.g., the Basic Iterative Method - BIM [5] and the Momentum Iterative Method - MIM [3]) to convince that our method indeed boots the robustness rather than suffers the gradient obfuscation [1]. Three attack methods PGD, BIM and MIM share the same setting, i.e., $\{k = 100, \epsilon = 0.3, \eta = 0.01\}$ for MNIST and $\{k = 100, \epsilon = 0.031, \eta = 0.007\}$ for CIFAR-10. The result as in Table 4 show that our components can improve the robustness of the baseline framework against all three kind of attacks which again proves the efficacy of our method.

5.1 Loss surface of adversarial examples

We separate adversarial examples into two classes: positive adversarial example which successfully fools a defense method and negative adversarial example which is an unsuccessful attack. The loss surface of positive adversarial example as Figure 1. In particular, both ADV (ADR-None) and our method (ADR+LC) predicted x_a with the label 8, whereas its true label is 3. From Figure 1, it is

Table 2. Experimental settings for our experiments. The model architectures are from [2] [6] and redescribed in the supplementary material.

	MNIST	CIFAR-10 (CNN)	CIFAR-10 (Resnet)
Architectures	CNN-4C3F(32)[2]	CNN-4C3F(64)[2]	RN-34-10[6]
Optimizer	SGD	Adam	SGD
Learning rate	0.01	0.001	0.1
Momentum	0.9	N/A	0.9
Training stratery	Batch size 128, 100 epochs	Batch size 128, 200 epochs	Batch size 128, 200 epochs
Perturbation	$k = 20, \epsilon_d = 0.3, \eta_d = 0.01, l_\infty$	$k = 10, \epsilon_d = 0.031, \eta_d = 0.007, l_\infty$	$k = 10, \epsilon_d = 0.031, \eta_d = 0.007, l_\infty$

Table 3. Performance comparison on different choices of the intermediate layer. The results in each setting are natural accuracy and robust accuracy against non-targeted/multiple-targeted attacks respectively.

	MNIST	CIFAR10
G=2Conv, C=2Conv+3FC	99.52/93.88/92.78	68.78/36.46/21.99
G=3Conv, C=1Conv+3FC	99.44/94.38/93.59	69.17/37.05/22.44
CNN (G=4Conv, C=3FC)	99.48/95.06/94.26	69.08/37.06/22.44
G=4Conv+1FC, C=2FC	99.51/94.38/93.47	69.39/37.31/22.87
G=4Conv+2FC, C=1FC	99.52/94.26/93.45	69.13/37.31/22.57

Table 4. Robustness comparison on the MNIST and CIFAR-10 datasets using Standard CNN with higher attack iteration (i.e., $k = 100$). The results in each setting are natural accuracy and robust accuracy against non-targeted/multiple-targeted attacks respectively.

	Dataset	ADV	ADR-ADV
PGD	MNIST	99.43/93.13/92.09	99.48/95.06/94.26
BIM	MNIST	99.43/93.00/91.70	99.48/94.86/93.99
MIM	MNIST	99.43/94.05/92.63	99.48/95.41/94.56
PGD	CIFAR-10	67.61/32.87/18.74	69.16/36.85/22.71
BIM	CIFAR-10	67.61/32.89/18.71	69.16/36.82/22.69
MIM	CIFAR-10	67.61/33.00/18.59	69.16/36.96/22.56

evident that for ADV, that most of its neighborhood region is non-smooth, resulting in incorrect predictions in almost all of the grid. By contrast, for our method (ADR+LC), the loss surface w.r.t. the input is smoother, resulting in more correct predictions in this neighborhood region. In addition, in our method, the prediction surface w.r.t. the latent feature in the intermediate representation layer is smoother than that w.r.t. input. This means that our local compactness makes the local region more compact, hence improving adversarial robustness.

We provide the loss surface of negative adversarial examples from adversarial training method and adversarial training with our components as Figure 2. Both examples show that the loss function smooth in local region of an adversarial example.

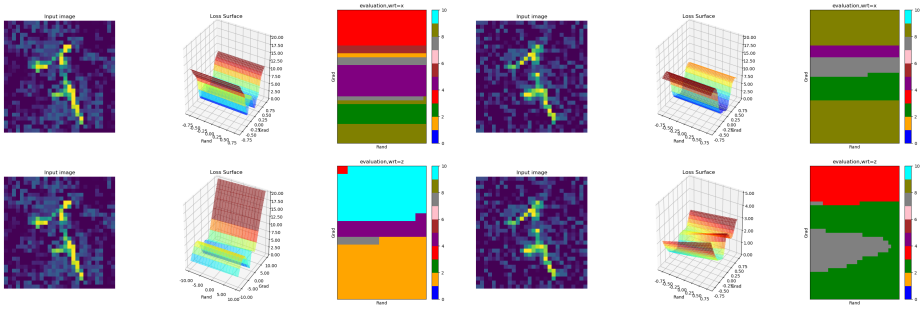


Fig. 1. Loss surface at local region of a positive adversarial example. Top-left: ADR-None w.r.t input. Top-right: ADV+LC w.r.t input. Bottom-left: ADR-None w.r.t latent. Bottom-right: ADV+LC w.r.t latent

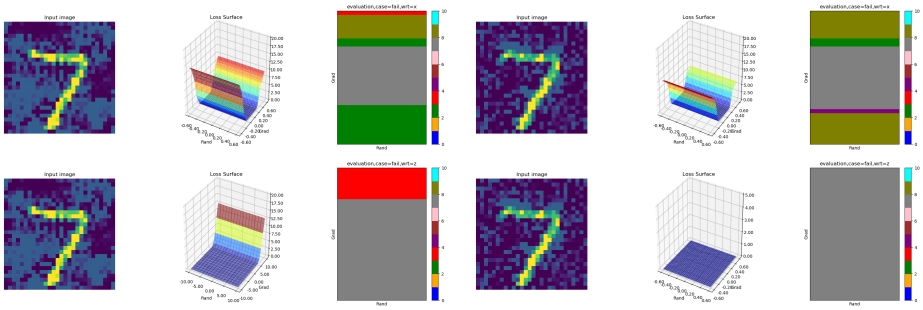


Fig. 2. Loss surface at local region of a negative adversarial example. Top-left: ADR-None w.r.t input. Top-right: ADV+LC w.r.t input. Bottom-left: ADR-None w.r.t latent. Bottom-right: ADV+LC w.r.t latent

5.2 T-SNE visualization of adversarial examples

In addition to positive adversarial examples, we provide the t-SNE visualization of the negative adversarial examples from adversarial training (ADR-None) and adversarial training with our components (ADR+LC/GB) as Figure 3. In adversarial training method, the unsuccessful attacks have been mixed insight the natural/clean data. In contrast, in case adversarial training with our components, the attack representation consistently is separated from those from natural data, similar to positive adversarial examples. Additionally, the unsuccessful attacks in adversarial training have the same confidence level with natural data, while those in our methods are totally different levels. In summary, our method can produce a better latent representation which is well separated between natural data and adversarial example (both positive and negative). This feature can be used for adversarial detection.

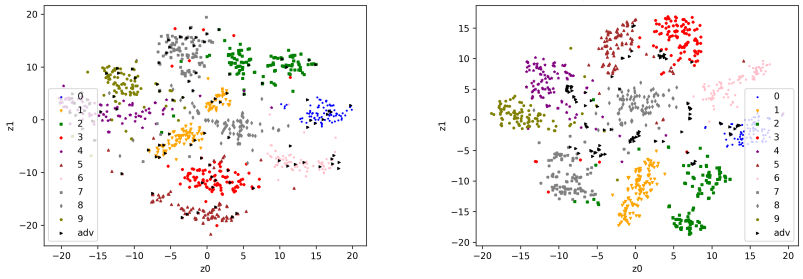


Fig. 3. T-SNE visualization of latent space. Black triangles are (negative) adversarial examples while others are clean images. Left: ADR-None. Right: ADR+LC/GB

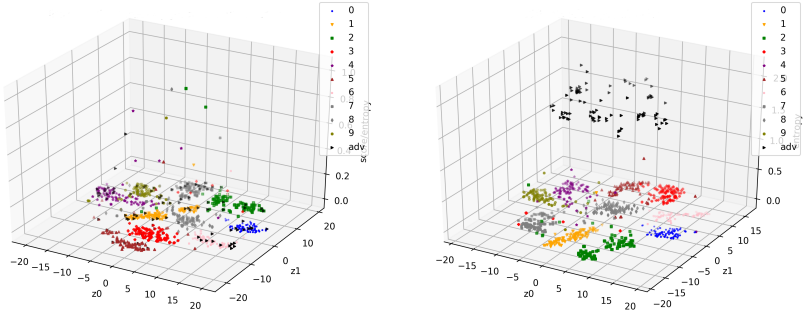


Fig. 4. T-SNE visualization with entropy of prediction with entropy of prediction probability. Black triangles are (negative) adversarial examples while others are clean images. Left: ADR-None. Right: ADR+LC/GB

References

1. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
2. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)
3. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9185–9193 (2018)
4. Jalal, A., Ilyas, A., Daskalakis, C., Dimakis, A.G.: The robust manifold defense: Adversarial training using generative models. arXiv preprint arXiv:1712.09196 (2017)
5. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
6. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
7. Qin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Fawzi, A., De, S., Stanforth, R., Kohli, P.: Adversarial robustness through local linearization. In: Advances in Neural Information Processing Systems. pp. 13824–13833 (2019)

3.4 Adversarial Supervised Contrastive Learning

Understanding and Achieving Efficient Robustness with Adversarial Supervised Contrastive Learning

Tuan-Anh Bui[a,*], Trung Le[a], He Zhao[b], Paul Montague[c], Seyit Camtepe[b] and Dinh Phung[a]

[a]Monash University, Australia
[c]Defence Science and Technology Group, Australia
[b]CSIRO Data61, Australia

ARTICLE INFO

Keywords:
adversarial robustness
contrastive learning

ABSTRACT

Contrastive learning (CL) has recently emerged as an effective approach to learning representation in a range of downstream tasks. Central to this approach is the selection of positive (similar) and negative (dissimilar) sets to provide the model the opportunity to 'contrast' between data and class representation in the latent space. In this paper, we investigate CL for improving model robustness using adversarial samples. We first designed and performed a comprehensive study to understand how adversarial vulnerability behaves in the latent space. Based on this empirical evidence, we propose an effective and efficient supervised contrastive learning to achieve model robustness against adversarial attacks. Moreover, we propose a new sample selection strategy that optimizes the positive/negative sets by removing redundancy and improving correlation with the anchor. Extensive experiments show that our Adversarial Supervised Contrastive Learning (ASCL) approach achieves comparable performance with the state-of-the-art defenses while significantly outperforms other CL-based defense methods by using only 42.8% positives and 6.3% negatives.

1. Introduction

Recently, there has been a considerable research effort on adversarial defense methods including Akhtar and Mian (2018); Lecuyer, Atlidakis, Geambasu, Hsu and Jana (2019); Carlini, Athalye, Papernot, Brendel, Rauber, Tsipras, Goodfellow, Madry and Kurakin (2019); Metzen, Genewein, Fischer and Bischoff (2017) which aim to develop a robust Deep Neural Network against adversarial attacks. Among them, the adversarial training methods (e.g, FGSM, PGD adversarial training (Goodfellow, Shlens and Szegedy, 2015; Madry, Makelov, Schmidt, Tsipras and Vladu, 2018) and TRADES (Zhang, Yu, Jiao, Xing, Ghaoui and Jordan, 2019)) that utilize adversarial examples as training data, have been one of the most effective series of approaches, which truly boost the model robustness without the facing the problem of obfuscated gradients (Athalye, Carlini and Wagner, 2018). In adversarial training, recently Xie, Wu, Maaten, Yuille and He (2019); Bui, Le, Zhao, Montague, deVel, Abraham and Phung (2020) show that reducing the divergence of the representations of images and their adversarial examples in the latent space (e.g., the feature space output from an intermediate layer of a classifier) can significantly improve the robustness. For example, in Bui et al. (2020), the latent representations of images in the same class are pulled closer together than those in different classes, which lead to a more compact latent space and consequently, better robustness.

On the other hand, as proposed recently, contrastive learning (CL) has been an increasingly popular and effective self-supervised representation learning approach (Chen, Kornblith, Norouzi and Hinton, 2020; He, Fan, Wu, Xie and Girshick, 2020; Khosla, Teterwak, Wang, Sarna, Tian, Isola, Maschinot, Liu and Krishnan, 2020). Specifically, CL learns representations of unlabeled data by choosing an anchor $\mathbf{x}_i$ and pulling the anchor and its positive samples closer in latent space while pushing it away from many negative samples. Intuitively, as the divergence in latent space is the focus of both AML and CL, it is natural to leverage CL to improve model robustness in adversarial training. However, we in this paper demonstrate that directly adopting CL into AML can hardly improve adversarial robustness, indicating that a deeper understanding of the relationships between the CL mechanism, latent space compactness, and adversarial robustness is required. Pursuing this comprehension, we give a detailed study on the above aspects, and subsequently propose a new framework for enhancing robustness using the principles of CL. Our paper provides answers for three research questions:

(Q1) ***Why*** *can CL help to improve the adversarial robustness?* To answer this question, we first introduce two kinds of divergences in the latent space: the *intra-class divergence* measured on benign images and their adversarial examples of the same class and the *inter-class divergence* measured on those samples of different classes. By comprehensively investigating the behavior of divergence in latent space, our study shows that the robustness of a model can be interpreted by the ratio between the intra- and inter-divergences: The lower the ratio is, the more robustness can be achieved. These observations motivate the idea that a robust model can be achieved by simultaneously contrasting the intra-class divergence between images and their adversarial examples with the inter-class divergence. We provide detailed analysis in Section 2.

(Q2) ***How*** *to integrate CL with adversarial training in*

*Corresponding author
tuananh.bui@monash.edu (T. Bui)
tuananhbui89.github.io (T. Bui)
ORCID(s): 0000-0003-4123-2628 (T. Bui)
[1]The code is available at https://github.com/tuananhbui89/ASCL

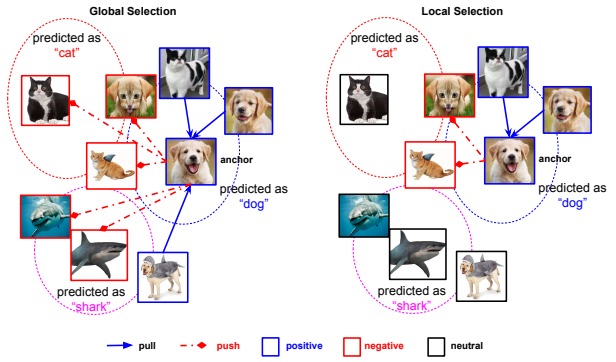


Figure 1: Illustration of ASCL with Global/Local Selection strategies in the latent space. While Global Selection considers all other images in the batch as either positives or negatives, Local Selection nominates the most relevant samples to the anchor when operating contrastive learning. The decision is based on the correlation between the true labels and the predicted labels as in Table 1.

the context of AML? CL originally works with the case where data labels are unavailable, which does not fit the context of AML for classifiers in the supervised setting. The recent research of Supervised Contrastive Learning (SCL) (Khosla et al., 2020) extends CL by leveraging label information, where the latent representations from the same class are pulled closer together than those from different classes. While it might seem to be straightforward to apply SCL for AML, we show in this paper that it is highly nontrivial to do so. To this end, we propose *Adversarial Supervised Contrastive Learning (ASCL)* to tackle this task by developing the following adaptions. Firstly, for an anchor image, we use its adversarial images as the transformed/augmented samples, which is different from the standard data augmentation techniques used in conventional CL methods (Chen et al., 2020; Khosla et al., 2020). Secondly, we integrate SCL with adversarial training (Madry et al., 2018) in addition to the clustering assumption (Chapelle and Zien, 2005), to enforce compactness in latent space and subsequently improve the adversarial robustness.

(Q3) ***What*** *are the important factors for the application of the ASCL framework in the context of AML?* One of the key steps of CL/SCL is the selecting of positive and negative samples for an anchor image. Although different approaches have been proposed, most of them focus on natural images and can usually be ineffective for AML. Specifically, in a data batch, CL and SCL consider the samples that are not from the same instance or not in the same class of the anchor image as its negative samples, which are hard splits between positive and negative sets, without taking into account the correlation between a sample and the anchor image. This can lead to too many true negative but useless samples which are highly uncorrelated with the anchor in the latent space as illustrated in Figure 1. This issue aggravates with more diverse data and in the AML context, making the original CL/SCL approaches inapplicable. We therefore develop a novel series of strategies for selecting positive and negative samples in our ASCL framework, which judiciously picks the most relevant samples of the anchor that help to further improve adversarial robustness.

By providing the answers to the above research questions, we summarize our contributions in this paper as follows:

1) We provide a comprehensive and insightful understanding of adversarial robustness regarding the divergences in latent space, which sheds light on adapting the CL principle to enhance robustness.

2) We propose a novel Adversarial Supervised Contrastive Learning (ASCL) framework, where the well-established contrastive learning mechanism is leveraged to make the latent space of a classifier more compact, leading to a more robust model against adversarial attacks.

3) By analyzing the intrinsic characteristics of AML, we develop effective strategies for selecting positive and negative samples more judiciously, which are critical to making contrastive learning principle effective in AML by using much less positives and negatives.

4) As shown in extensive experiments, our proposed framework is able to significantly improve a classifier's robustness, outperforming several adversarial training defense methods against strong attacks while achieving comparable performance with SOTA defenses in the RobustBench (Croce, Andriushchenko, Sehwag, Debenedetti, Flammarion, Chiang, Mittal and Hein, 2020).

2. Analysis of Latent Space Divergence

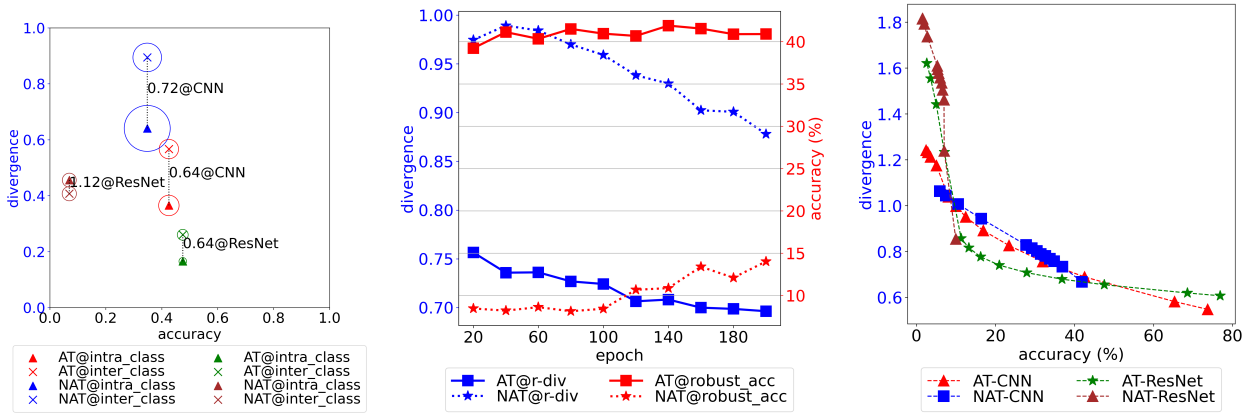
By examining the question "Why can CL help to improve the adversarial robustness?", we design experiments to show the connection of adversarial robustness to the latent divergence of an anchor and its contrastive samples.

Let $\mathcal{B} = \{\mathbf{x}_i, \boldsymbol{y}_i\}_{i=1}^N$ be a batch of benign images where image $\mathbf{x}_i$ is associated $\boldsymbol{y}_i$. Given an adversarial transformation $\mathcal{T}$ from an adversary $\mathcal{A}$ (e.g., PGD attack in Madry et al. (2018)), we consider two kinds of samples w.r.t. an anchor $\{\mathbf{x}_i, \boldsymbol{y}_i\}$: the *positive* set $\boldsymbol{X}_i^+ = \{\mathbf{x}_j, \mathbf{x}_j^{\mathcal{T}} \mid j \neq i, \boldsymbol{y}_j = \boldsymbol{y}_i\}$ including benign examples and their counterparts in the same class with the anchor and the *negative* set $\boldsymbol{X}_i^- = \{\mathbf{x}_j, \mathbf{x}_j^{\mathcal{T}} \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i\}$ including benign examples and their counterparts in different classes with the anchor. We are interested in the latent representations of begin and transformed images at a specific intermediate layer of the neural net classifier f. Let us further denote those representations by $\mathbf{z}_i$ for benign images and $\mathbf{z}_i^{\mathcal{T}}$ for adversarially transformed images according $\mathcal{T}$. We define some types of divergences between benign images and transformed images via transformation $\mathcal{T}$ at some intermediate layers of f.

(i) ***Absolute*** **intra-class divergence**:
$d_a^+ = \frac{1}{N}\sum_{i=1}^N \frac{1}{|\boldsymbol{X}_i^+|}\sum_{\mathbf{x}_j \in \boldsymbol{X}_i^+} d(\mathbf{z}_i, \mathbf{z}_j)$

(ii) ***Absolute*** **inter-class divergence**:
$d_a^- = \frac{1}{N}\sum_{i=1}^N \frac{1}{|\boldsymbol{X}_i^-|}\sum_{\mathbf{x}_j \in \boldsymbol{X}_i^-} d(\mathbf{z}_i, \mathbf{z}_j)$

Here we note that d is cosine distance between two representations, and $|.|$ represents the cardinality of a set.



(a) Pairs of Absolute-DIVs with corresponding R-DIV.

(b) R-DIV over the training progress with standard CNN model.

(c) R-DIV under different attack strengths.

Figure 2: Correlation between the *Relative intra-class* divergence (R-DIV) and the robust accuracy on the CIFAR10 dataset. The variance of Absolute-DIV in Figure 2a is scaled by 0.2 for better visualization. Using PGD attack with $\epsilon = 8/255, \eta = 2/255$ with $k = 10$ for training and $k = 250$ for testing.

(iii) ***Relative* intra-class divergence** (R-DIV): $d_r^+ = \frac{d_a^+}{d_a^-}$; hence relative divergence generally represents how large the magnitude of intra-class divergence is relative to the inter-class divergence.

In Figure 2, we conduct an empirical study on the CIFAR-10 dataset to figure out the relationship between R-DIV for adversarial examples and robust accuracy. The findings and observations are very important for us to devise our framework in the sequel. More specifically, we train a CNN and a ResNet20 model in two modes: natural mode (NAT and cannot defend at all) and adversarial training mode (AT and can defend quite well). We observe how robust accuracy together with R-DIV vary with training progress to draw conclusions. The detailed settings and further demonstrations can be found in the supplementary material. Some observations are drawn from our experiment:

(O1) **The robustness varies *inversely* with the *relative* intra-class divergence between benign images and their adversarial examples** (the adversarial *R-DIV* $d_r^{+,adv}$). As shown in Figure 2b, during the training process, the robust accuracy of the AT model tends to improve, which corresponds with a decrease of the adversarial *R-DIV* $d_r^{+,adv}$. Similarly, when the robust accuracy of the NAT model starts increasing at the epoch 100, the adversarial *R-DIV* $d_r^{+,adv}$ starts decreasing. In addition, the robust accuracy of the AT model is significantly higher than that of the NAT model, whilst its $d_r^{+,adv}$ is far lower than that of the NAT model. In Figure 2c, we visualize the correlation between the R-DIV and the robust accuracy by generating different attack strengths. It can be seen that there is a common trend such that the lower robust accuracy the higher R-DIV, regardless of the model architecture or defense methods. These observations support our claim of the relation between robust accuracy and R-DIV.

(O2) In Figure 2a, we visualize the absolute intra-class divergence (d_a^+) and the absolute inter-class divergence (d_a^-) for the cases of the NAT/AT models with their corresponding robust accuracies. It can be observed that: (i) in the same architecture, the d_a^+ of the AT model is much smaller than that of NAT model. However, the d_a^- of the AT model is also much smaller than that of NAT model. It implies that, the AT method helps to compact the representations of intra-class samples, but undesirably makes the representations of inter-class samples closer. (ii) Overall, the relative intra-class divergence of the AT model is smaller than that of the NAT model – which might explain why the NAT model is easy to be attacked, and again confirms our O1.

Conclusions from the observations. Mao, Zhong, Yang, Vondrick and Ray (2019) and Bui et al. (2020) reached a conclusion that the *absolute adversarial intra-class divergence* d_a^+ is a key factor for robustness against adversarial examples. However, as indicated by our O1, it is only one side of the coin. The reason is that the *absolute adversarial intra-class divergence* only cares about how far adversarial examples of a class are from their counterpart benign images, and does not pay attention to the inter-divergence to other classes. As analysed in our observation of O2-i, low d_a^- possibly harms the robust accuracy, because in this case, adversarial examples of other classes are very close to those of the given class. This further indicates that the *absolute adversarial inter-class divergence* d_a^- needs to be taken into account and it is necessary to minimize the *relative adversarial intra-class divergence* $d_r^+ = \frac{d_a^+}{d_a^-}$ better controls both *the absolute adversarial intra-class divergence* and *absolute adversarial inter-class divergence* for strengthening robustness. The above analytical and empirical study confirms the feasibility of applying SCL to enhance robustness in AML but one can also see that it is non-trivial to develop an appropriate strategy to be the combination effective.

3. Proposed method

In this section, we provide the answer for the question "***How** to integrate CL with adversarial training in the context of AML?*". We first propose an adapted version of SCL which we call Adversarial Supervised Contrastive Learning (ASCL) for the AML problem. We then introduce three sample selection strategies to nominate the most relevant positives and negatives to the anchor, which further improve robustness with much fewer samples.

3.1. Adversarial Supervised Contrastive Learning

Terminologies. We consider a prediction model $h(\mathbf{x}) = g(f(\mathbf{x}))$ where $f()$ is the encoder which outputs the latent representation $\mathbf{z} = f(\mathbf{x})$ and $g()$ is the classifier upon the latent $\mathbf{z}$. Also we have a batch of N pairs $\{\mathbf{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$ of benign images and their labels. With an adversarial transformation $\mathcal{A}$ (e.g., PGD), each pair $\{\mathbf{x}_i, \boldsymbol{y}_i\}$ has two corresponding sets, a positive set $\boldsymbol{X}_i^+ = \{\mathbf{x}_j, \mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j = \boldsymbol{y}_i\}$ and a negative set $\boldsymbol{X}_i^- = \{\mathbf{x}_j, \mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i\}$. We then have the corresponding sets in the latent space $\boldsymbol{Z}_i^+ = \{f(\mathbf{x}_j) \mid \mathbf{x}_j \in \boldsymbol{X}_i^+\}$ and $\boldsymbol{Z}_i^- = \{f(\mathbf{x}_j) \mid \mathbf{x}_j \in \boldsymbol{X}_i^-\}$.

Supervised Contrastive Loss. The supervised contrastive loss for an anchor $\mathbf{x}_i$ as follow:

$$\mathcal{L}_i^{\text{scl}} = \frac{-1}{\left|\boldsymbol{Z}_i^+\right| + 1} \sum_{\mathbf{z}_j \in \boldsymbol{Z}_i^+ \cup \{\mathbf{z}_i^a\}} \log \frac{e^{\frac{sim(\mathbf{z}_j, \mathbf{z}_i)}{\tau}}}{\sum_{\mathbf{z}_k \in \boldsymbol{Z}_i^+ \cup \boldsymbol{Z}_i^- \cup \{\mathbf{z}_i^a\}} e^{\frac{sim(\mathbf{z}_k, \mathbf{z}_i)}{\tau}}} \quad (1)$$

where $sim(\mathbf{z}_j, \mathbf{z}_i)$ represents the similarity metric between two latent representations and τ is a temperature parameter. It is worth noting that there are two changes in our SCL loss compared with the original one in Khosla et al. (2020). Firstly, $sim(\mathbf{z}_j, \mathbf{z}_i)$ is a general form of similarity, which can be any similarity metric such as cosine similarity $\frac{\mathbf{z}_j \cdot \mathbf{z}i}{\|\mathbf{z}_j\| \times \|\mathbf{z}_i\|}$ or Lp norm $-\left|\mathbf{z}_j - \mathbf{z}_i\right|_p$. Secondly, in term of terminology, in Khosla et al. (2020), the positive set was defined including those samples in the same class with the anchor $\mathbf{x}_i$ (e.g. $\boldsymbol{X}_i^+$) and the anchor's transformation $\mathbf{x}_i^a$. However, in our paper, we want to emphasize the importance of the anchor's transformation, therefore, we use two separate terminologies $\boldsymbol{X}_i^+$ and $\{\mathbf{x}_i^a\}$. Similarly, the SCL loss for an anchor $\mathbf{x}_i^a$ as follow:

$$\mathcal{L}_i^{\text{a,scl}} = \frac{-1}{\left|\boldsymbol{Z}_i^+\right| + 1} \sum_{\mathbf{z}_j \in \boldsymbol{Z}_i^+ \cup \{\mathbf{z}_i\}} \log \frac{e^{\frac{sim(\mathbf{z}_j, \mathbf{z}_i)}{\tau}}}{\sum_{\mathbf{z}_k \in \boldsymbol{Z}_i^+ \cup \boldsymbol{Z}_i^- \cup \{\mathbf{z}_i\}} e^{\frac{sim(\mathbf{z}_k, \mathbf{z}_i)}{\tau}}} \quad (2)$$

The average SCL loss over a batch is as follows:

$$\mathcal{L}^{\text{SCL}} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathcal{L}_i^{\text{scl}} + \mathcal{L}_i^{\text{a,scl}}\right) \quad (3)$$

As mentioned in Khosla et al. (2020), there is a major advantage of SCL compared with Self-Supervised CL (SSCL) in the context of regular machine learning. Unlike SSCL in which each anchor has only single positive sample, SCL takes advantages of the labels to have many positives in the same batch size N. This strategy helps to reduce the false negative cases in SSCL when two samples in the same class are pushed apart. As shown in Khosla et al. (2020), SCL training is more stable than SSCL and also achieves better performance.

Adaptations in the context of AML. However, original SCL is not sufficient to achieve adversarial robustness. In the context of adversarial machine learning, we need the following adaptations to improve the adversarial robustness:

(i) As shown in Table 1 in Kim, Tack and Hwang (2020), the original SCL slightly improves the robustness of a standard model but cannot defend strong adversarial attacks. Therefore, we use an adversary $\mathcal{A}$ (e.g., PGD) as the transformation $\mathcal{T}$ instead of the traditional data augmentation (e.g., combination of random cropping and random jittering) as in other contrastive learning frameworks (Chen et al., 2020; Khosla et al., 2020; He et al., 2020). This helps to reduce the divergence in latent representations of a benign image and its adversarial example directly.

(ii) We apply SCL as a regularization on top of the Adversarial Training (AT) method (Madry et al., 2018; Zhang et al., 2019; Shafahi, Najibi, Ghiasi, Xu, Dickerson, Studer, Davis, Taylor and Goldstein, 2019; Xie, Tan, Gong, Yuille and Le, 2020). Therefore, instead of pre-training the encoder $f()$ with contrastive learning loss as in previous work, we can optimize the AT and the SCL simultaneously. The AT objective function with the cross-entropy loss $C()$ is as follows:

$$\mathcal{L}^{AT} = \frac{1}{N} \sum_{i=1}^{N} C\left(h(\mathbf{x}_i), \boldsymbol{y}_i\right) + C\left(h(\mathbf{x}_i^a), \boldsymbol{y}_i\right) \quad (4)$$

Regularization on the prediction space. The clustering assumption (Chapelle and Zien, 2005) is a technique that encourages the classifier to preserve its predictions for data examples in a cluster. Theoretically, the clustering assumption enforces the decision boundary of a given classifier to lie in the gap among the data clusters and never cross over any clusters. As shown in Chen et al. (2020); Khosla et al. (2020), with the help of CL, latent representations of those samples in the same class form into clusters. Therefore, coupling our SCL framework with the clustering assumption can help to increase the margin from a data sample to the decision boundary. To enforce the clustering assumption, we use Virtual Adversarial Training (VAT) (Miyato, Maeda, Koyama and Ishii, 2019) to maintain the classifier smoothness:

$$\mathcal{L}^{\text{VAT}} = \frac{1}{N} \sum_{i=1}^{N} D_{KL}\left(h(\mathbf{x}_i) \parallel h(\mathbf{x}_i^a)\right) \quad (5)$$

Putting it all together. We combine the relevant terms to the final objective function of our framework which we name

Table 1
Definitions of positives and negatives with Global Selection and Local Selection strategies given an anchor $\{\mathbf{x}_i, \boldsymbol{y}_i\}$ and a predicted label $p = \text{argmax}\, h(\mathbf{x})$, $p^a = \text{argmax}\, h(\mathbf{x}^a)$

	$\boldsymbol{X}_i^+$	$\boldsymbol{X}_i^-$
Global	$\{\mathbf{x}_j, \mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j = \boldsymbol{y}_i\}$	$\{\mathbf{x}_j, \mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i\}$
Hard-LS	$\{\mathbf{x}_j, \mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j = \boldsymbol{y}_i\}$	$\{\mathbf{x}_j \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i, p_j = \boldsymbol{y}_i\} \cup \{\mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i, p_j^a = \boldsymbol{y}_i\}$
Soft-LS	$\{\mathbf{x}_j, \mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j = \boldsymbol{y}_i\}$	$\{\mathbf{x}_j \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i, p_j = p_i\} \cup \{\mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i, p_j^a = p_i\}$
Leaked-LS	$\{\mathbf{x}_j \mid j \neq i, \boldsymbol{y}_j = \boldsymbol{y}_i, p_j = p_i\} \cup \{\mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j = \boldsymbol{y}_i, p_j^a = p_i\}$	$\{\mathbf{x}_j \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i, p_j = p_i\} \cup \{\mathbf{x}_j^a \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i, p_j^a = p_i\}$

as Adversarial Supervised Contrastive Learning (ASCL) as follows:

$$\mathcal{L} = \mathcal{L}^{\text{AT}} + \lambda^{\text{SCL}}\mathcal{L}^{\text{SCL}} + \lambda^{\text{VAT}}\mathcal{L}^{\text{VAT}} \tag{6}$$

where λ^{SCL} and λ^{VAT} are hyper-parameters to control the SCL loss and VAT loss, respectively. As mentioned in the observation (O2), minimizing the AT loss $\mathcal{L}^{AT}$ alone compresses not only the representations of intra-class clusters but also reduces the inter-class distance, which hurts the natural discrimination. Therefore, coupling with $\mathcal{L}^{SCL}$ can compensate the aforementioned weakness by simultaneously minimizing the intra-class divergence and maximizing the inter-class divergence. Finally, by forcing predictions of intra-class samples to be close, the VAT regularization $\mathcal{L}^{VAT}$ help to maintain the classifier smoothness and further improve the robustness. In addition to the intuitive analysis, we also provide an empirical ablation study to further understand the contribution of each component in the supplementary material.

3.2. Global and Local Selection Strategies

Global Selection. The SCL as in Equations 1,2 can be understood as SCL with a Global Selection strategy, where each anchor $\mathbf{x}_i$ takes all other samples in the current batch into account and splits them into a positive set $\boldsymbol{X}_i^+$ and a negative set $\boldsymbol{X}_i^-$. For example, as illustrated in Figure 1, given an anchor, with the help of SCL, it will push away all negatives and pull all positives regardless of their correlation in the space. However, there are two issues of this strategy:

(I1) The high inter-class divergence issue of a diverse dataset. Specifically, there are true negative (but uncorrelated) samples which are very different in appearance (e.g., an anchor-dog and negative samples-sharks) and latent representations. Therefore, pushing them away does not make any contribution to the learning other than making it more unstable. The number of uncorrelated negatives is increased when the dataset is more diverse.

(I2) The high intra-class divergence issue when the dataset is very diverse in some classes. For example, a class "dog" in the ImageNet dataset may include many sub-classes (breeds) of dog. Specifically, there are true positive (but uncorrelated) samples which are in the same class with the anchor but different in appearance. In the context of AML, two samples in the same class (e.g., "dog") can be attacked to be very different classes (e.g., one to the class "cat", one to the class "shark"), therefore the latent representations of their adversarial examples are even more uncorrelated.

Local Selection. Based on the above analysis, we leverage label supervision to propose a series of Local Selection (LS) strategies for the SCL framework, which consider *local and important* samples only and ignore other samples in the batch as illustrated in Figure 1. They are *Hard-LS, Soft-LS* and *Leaked-LS* as defined in Table 1.

More specifically, in Hard-LS and Soft-LS, we consider the same set of positives as in Global Selection. However, we filter out the true negative but uncorrelated samples by only considering those are predicted as similar to the anchor's true label (Hard-LS) or to the anchor's predicted label (Soft-LS). These two strategies are to deal with the issue (I1) by choosing negative samples that have most correlation with the current anchor. Because they are very close in prediction space, their representation is likely high correlated with the anchor's representation.

In Leaked-LS, we add an additional constraint on the positive set to deal with the issue (I2). Specifically, we filter out the true positive but uncorrelated samples by only choosing those are currently predicted as similar to the anchor's prediction. It is worth noting that, the additional constraint is applied on the positive set $\boldsymbol{X}_i^+$ only. It means that, each anchor $\mathbf{x}_i$ and its adversarial example $\mathbf{x}_i^a$ are always pulled close together. However, instead of pulling all other positive samples in current batch, we only pull those samples which are close with the anchor's representation to further support and stabilize the contrastive learning.

From a practical perspective, as later shown in the experimental section, ASCL with Leaked-Local Selection (Leaked ASCL) improves the robustness over that with Global Selection most notably, and with much fewer positive and negative samples. It has been shown that, optimal negative samples for contrastive learning are task-dependent which guide representations towards task-relevant features that improve performance (Tian, Sun, Poole, Krishnan, Schmid and Isola, 2020; Frankle, Schwab, Morcos et al., 2020). However, while these previous works focused on unsupervised-setting, our Local-ASCL is the first work to leverage supervision to select not only optimal negative samples but also optimal positive samples for robust classification task.

4. Experiments

In this section, we first introduce the experimental setting for adversarial attacks and defenses. We then provide an extensive robustness evaluation between our best method (which is Leaked-ASCL) with other defenses to demonstrate the significant improvement of ours. Finally, we empirically answer the question "***What** are the important factors for the application of the ASCL framework in the context of AML?*" through our experiments. We provide a comparison among Global/Local Selection strategies and show that the Leaked-ASCL not only outperforms the Global ASCL but also makes use of much fewer positives and negatives. An ablation study to investigate the importance of each component to the performance can be found in the supplementary material.

4.1. Experimental Setting

General Setting. We use CIFAR10 and CIFAR100 datasets (Krizhevsky et al., 2009) as the benchmark datasets in our experiment. Both datasets have 50,000 training images and 10,000 test images. However, while the CIFAR10 dataset has 10 classes, CIFAR100 is more diverse with 100 classes. The inputs were normalized to $[0, 1]$. We apply random horizontal flips and random shifts with scale 10% for data augmentation as used in Pang, Xu, Du, Chen and Zhu (2019). We use four architectures including standard CNN, ResNet18, ResNet20 (He, Zhang, Ren and Sun, 2016) and WideResNet-34-10 (Zagoruyko and Komodakis, 2016) in our experiment. The architecture and training setting for each dataset are provided in our supplementary material.

Contrastive Learning Setting. We choose the penultimate layer (l_y^{-1}) as the intermediate layer to apply our regularization. The analytical study for the effect of choosing projection head in the context of AML can be found in the supplementary material. In the main paper, we report the experimental results without the projection head. The temperature $\tau = 0.07$ as in Khosla et al. (2020).

Attack Setting. We use different state-of-the-art attacks to evaluate the defense methods including: (i) **PGD attack** which is a gradient based attack. We use $k = 250, \epsilon = 8/255, \eta = 2/255$ for the CIFAR10 dataset and $k = 250, \epsilon = 0.01, \eta = 0.001$ for the CIFAR100 dataset. We use two versions of the PGD attack: the non-targeted PGD attack (PGD) and the multi-targeted PGD attack (mPGD). (ii) **Auto-Attack** (Croce and Hein, 2020) which is an ensemble based attack. We use $\epsilon = 8/255$ for the CIFAR10 dataset and $\epsilon = 0.01$ for the CIFAR100 dataset, both with the standard version of Auto-Attack (AA), which is an ensemble of four different attacks. The distortion metric we use in our experiments is l_∞ for all measures. We use the full test set for the attacks (i) and 1000 test samples for the attacks (ii).

Generating Adversarial Examples for Defenders. We employ PGD as the stochastic adversary to generate adversarial examples. These adversarial examples have been used as transformations of benign images in our contrastive framework. Specifically, the configuration for the CIFAR10 dataset is $k = 10, \epsilon = 8/255, \eta = 2/255$ and that for the CIFAR100 dataset is $k = 10, \epsilon = 0.01, \eta = 0.001$.

Table 2
Robustness evaluation on the CIFAR10 and CIFAR100 datasets with ResNet20 architecture. Ours is Leaked-ASCL variant. GAP represents the average gap of robust accuracies between ours and the compared method.

	CIFAR10					CIFAR100				
	Nat.	PGD	mPGD	AA	GAP	Nat.	PGD	mPGD	AA	GAP
ADV	**78.8**	48.1	36.4	36.1	5.37	**60.7**	35.7	25.3	25.7	6.27
TRADES	**76.1**	51.9	38.2	36.3	3.43	**59.0**	37.2	25.3	25.7	5.77
ADR	76.8	51.5	38.9	38.6	2.57	59.1	40.0	29.1	28.6	2.60
Ours	75.5	**53.7**	**41.0**	**42.0**	0	59.0	**42.5**	**31.1**	**31.9**	0

Table 3
Robustness evaluation against Auto-Attack with ResNet18 and WideResNet on the full test set of CIFAR10 dataset. ⋆ Results are copied from Croce et al. (2020). ‡ Results are copied from original papers, using a larger batch size (bs). ∗ Omit the cross-entropy loss of natural images and VAT loss. Detail can be found in the supplementary material.

	Model	Nat	AA	PGD
Ours ∗	WideResNet	87.70	52.80	54.05
Zhang et al. (2020) ⋆	WideResNet	84.52	53.51	-
Huang et al. (2020) ⋆	WideResNet	83.48	53.34	-
Zhange et al. (2019) ⋆	WideResNet	84.92	53.08	-
Cui et al. (2021) ⋆	WideResNet	88.22	52.86	-
Ours ∗	ResNet18	85.02	50.31	53.40
ACL-DS (bs=512) ‡	ResNet18	82.19	-	52.82
RoCL-TRADES (bs=256) ‡	ResNet18	84.55	-	43.85

Baseline methods. Most closely related to our work is ADR (Bui et al., 2020) which also aims to realize compactness in the latent space to improve robustness in the supervised setting. We also compare with RoCL-TRADES (Kim et al., 2020) and ACL-DS (Jiang, Chen, Chen and Wang, 2020) which pre-trains with adversarial examples founded by InfoNCE loss and post-trains with standard supervised adversarial training[2].

4.2. Robustness evaluation

We conduct extensive evaluations to demonstrate the advantages of our method (Leaked-ASCL variant) over other defenses. Table 2 shows the robustness comparison on the CIFAR10 and CIFAR100 datasets with ResNet20 architecture. It can be seen that our method achieves much better robustness than the baseline methods on both datasets. More specifically, on the CIFAR10 dataset, the average gaps of robust accuracies against three attacks (PGD, mPGD and AA) between ours and ADR, TRADES, and ADV are 2.57%, 3.43% and 5.37%, respectively. The similar gaps for the CIFAR100 dataset are 2.60%, 5.77% and 6.27%, respectively. Figure 3 shows the tradeoff between natural accuracy and ro-

[2]The best reported version RoCL-AT-SS is a fine-tuned on a self-supervised ImageNet pretrained model, therefore, might not as a reference for comparison.

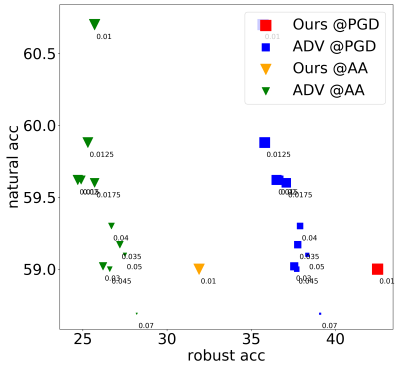
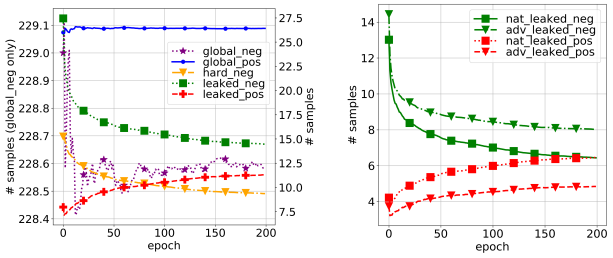


Figure 3: Tradeoff between natural/robust accuracies when increasing perturbation magnitude (specified beside markers). For better visualization, bigger marker indicates smaller perturbation. Ours is Leaked-ASCL.



(a) Global/Local (b) Detail of Leaked Local

Figure 4: Number of positives and negatives with different Global/Local Selection strategy on CIFAR10 dataset with batch size 128

bust accuracies when increasing perturbation magnitude. It can be seen that simply increasing the magnitude of adversarial examples cannot reach our performance even with a fine-range of perturbation. With the same level of natural accuracy, our method outperforms the baseline by around 5% which again emphasizes the advantage of our method. Finally, we compare our method with recently listed methods on the RobustBench (Croce et al., 2020) which have a similar setting (e.g., without additional data) as shown in Table 3. With WideResNet architecture, our method achieves 52.80% robust accuracy against Auto-Attack and 87.70% natural accuracy which is comparable with the SOTA method from Cui, Liu, Wang and Jia (2021). Compare to the best robust method from Zhang, Xu, Han, Niu, Cui, Sugiyama and Kankanhalli (2020), our method has 0.7% lower in robust accuracy but 3.2% higher in natural performance. With a smaller batch size, our method still achieves much better performance than RoCL and ACL which are two SOTA self-supervised contrastive learning based defenses.

4.3. Global and Local Selection strategies

We compare the effect of different global or local selection strategies to the final performance. Table 4 shows that while the Hard-ASCL and Soft-ASCL show a small improvement over ASCL, the Leaked-ASCL achieves the best robustness compared with other strategies. We also measure the average number of positive and negatives samples per batch corresponding with different selection strategies as shown in Figure 4a. With batch size 128, we have a total of 256 samples per batch including benign images and their adversarial examples. It can be seen that, the average positives and negatives by the Global Selection are stable at 26.4 and 228.6, respectively. In contrast, the number of positives and negatives by the Leaked-LS vary corresponding with the current performance of the model. More specifically, there are four advantages of the Leaked-LS over the Global Selection:

Table 4
Comparison among Global/Local Selection Strategies on the CIFAR10 dataset with ResNet20

	Nat.	PGD	mPGD	AA
(Global) ASCL	**76.4**	52.7	40.4	40.9
Hard-ASCL	75.5	53.1	41.0	41.3
Soft-ASCL	75.5	53.4	40.6	40.4
Leaked-ASCL	75.5	**53.7**	**41.0**	**42.0**

(i) at the beginning of training, approximately 7.5 positive samples and 25 negative samples were selected. This is because of the low classification performance of the model. Moreover, the strength of the contrastive loss is directly proportional with the size of the positive set. Therefore, with a small positive set, the contrastive loss is weak in comparison with other components of ASCL. This helps the model focuses more on improving the classification performance first.

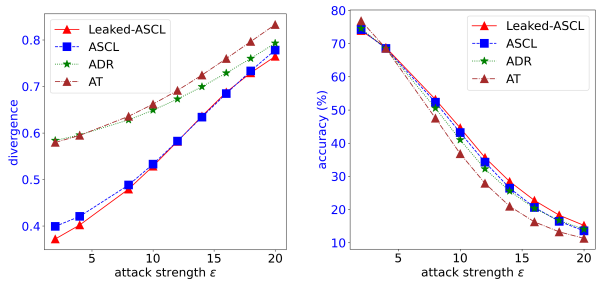
(ii) when the model is improved, the number of positive samples is increased, while the number of negative samples is decreased significantly. In addition to the bigger positive set, the contrastive loss become stronger in comparison with other components. This helps the model now focus more on contrastive learning and learning the compact latent representation.

(iii) unlike Global Selection, Leaked-LS considers natural images and adversarial images differently based on their hardness to the current anchor. As shown in Figure 4b, there are more adversarial images than natural images in the negative set, which helps the encoder focus to contrast the anchor with the adversarial images.

(iv) at the last epoch, Leaked-LS chooses only 11.3 positives and 14.3 negatives, which equate to 42.8% and 6.3% of the positive set and negative set with the Global Selection strategy, respectively.

4.4. Why do ASCL and Local-ASCL improve adversarial robustness

In this subsection, we connect with the hypothesis in Section 2 to explain why our ASCL and especially our Leaked-ASCL help to improve adversarial robustness. Figure 5 shows the Relative intra-class divergence (R-DIV) and robust accuracy under PGD attack $\{k = 250, \eta = 2/255\}$ with different attack strengths ϵ. It can be seen that (i) our ASCL and Leaked-ASCL have lower R-DIV than baseline methods, and Leaked-ASCL achieves the lowest measure, (ii) consequently, our ASCL and Leaked-ASCL achieves better robust accuracy than baseline methods. Leaked-ASCL achieve



(a) R-DIV (b) Robust accuracy

Figure 5: R-DIV and robust accuracy under different attack strengths on CIFAR10 with ResNet20.

the best performance regardless of attack scenarios. The experimental results concur with the proposed correlation between the Relative intra-class divergence and the adversarial robustness as pointed out in Section 2. Our methods help the representations of intra-class samples to be more compact while increasing the margin between inter-class clusters, and therefore improve the robustness.

5. Conclusion

In this paper, we have shown the connection between robust accuracy and the divergence in latent spaces. We demonstrated that contrastive learning can be applied to improve adversarial robustness by reducing the intra-instance divergence while maintaining the inter-class divergence. Moreover, we have shown that, instead of using all negatives and positives as per the regular contrastive learning framework, by judiciously picking highly correlated samples, we can further improve the adversarial robustness.

Acknowledgement

This work was partially supported by the Australian Defence Science and Technology (DST) Group under the Next Generation Technology Fund (NGTF) scheme.

Appendix

A. Experimental setting

General Setting. We use CIFAR10 and CIFAR100 dataset (Krizhevsky et al., 2009) as the benchmark datasets in our experiment. Both datasets have 50,000 training images and 10,000 test images. However, while the CIFAR10 dataset has 10 classes, CIFAR100 is more diverse with 100 classes. The training time is 200 epochs for both CIFAR10 and CIFAR100 datasets with batch size 128. The inputs were normalized to $[0, 1]$. We apply random horizontal flips and random shifts with scale 10% for data augmentation as used in Pang et al. (2019).

We use four architectures including CNN, ResNet18/20 (He et al., 2016) and WideResNet-34-10 (Zagoruyko and Komodakis, 2016) in our experiment. The standard CNN architecture has 4 convolution layers followed by 3 FC layers as described in Carlini and Wagner (2017). For ResNet20 architectures, we use the same training setting as in Pang et al. (2019). More specifically, we use Adam optimizer, with learning rate $10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$ at epoch 0th, 80th, 120th, and 160th, respectively. We use Adam optimization with learning rate 10^{-3} for training the standard CNN architecture.

For ResNet18 and WideResNet architectures, we use the same training setting as in Pang, Yang, Dong, Su and Zhu (2020). More specifically, we use SGD optimizer, with momentum $5{\times}10^{-4}$, with learning rate $10^{-1}, 10^{-2}, 10^{-3}$ at epoch 0th, 100th and 150th, respectively. It is a worth noting that, there are some modifications in the experiment in Table 3 to match the performance as in RobustBench: (i) we omit the cross-entropy loss of natural images in Eq. (4), so that the model sacrifices natural performance to gain more robust performance. The AT objective function becomes: $\mathcal{L}^{AT} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{C}\left(h(\mathbf{x}_i^a), \mathbf{y}_i\right)$ which is similar as in Pang et al. (2020). (ii) we omit the VAT loss to show that the improvement truly comes from the contribution of the adversarial contrastive loss.

Contrastive Learning Setting. We apply the contrastive learning on the intermediate layer (l_y^{-1}) which is intermediately followed by the last FC layer of either CNN or ResNet or WideResNet architectures. The analytical study for the effect of choosing projection head in the context of AML can be found in Section C.1. In the main paper, we report the experimental results without the projection head. The temperature $\tau = 0.07$ as in Khosla et al. (2020).

Attack Setting. We use different state-of-the-art attacks to evaluate the defense methods including: (i) **PGD attack** which is a gradient based attack. We use $k = 250, \epsilon = 8/255, \eta = 2/255$ for the CIFAR10 dataset and $k = 250, \epsilon = 0.01, \eta = 0.001$ for the CIFAR100 dataset. We use two versions of the PGD attack: the non-targeted PGD attack (PGD) and the multi-targeted PGD attack (mPGD). (ii) **Auto-Attack** (Croce and Hein, 2020) which is an ensemble based attack. We use $\epsilon = 8/255$ for the CIFAR10 dataset and $\epsilon = 0.01$ for the CIFAR100 dataset, both with the standard version of Auto-Attack (AA), which is an ensemble of four different attacks. The distortion metric we use in our experiments is l_∞ for all measures. We use the full test set for the attacks (i) and 1000 test samples for the attacks (ii).

Generating Adversarial Examples for Defenders. We employ PGD as the stochastic adversary to generate adversarial examples. These adversarial examples have been used as transformations of benign images in our contrastive framework. Specifically, the configuration for the CIFAR10 dataset is $k = 10, \epsilon = 8/255, \eta = 2/255$ and that for the CIFAR100 dataset is $k = 10, \epsilon = 0.01, \eta = 0.001$.

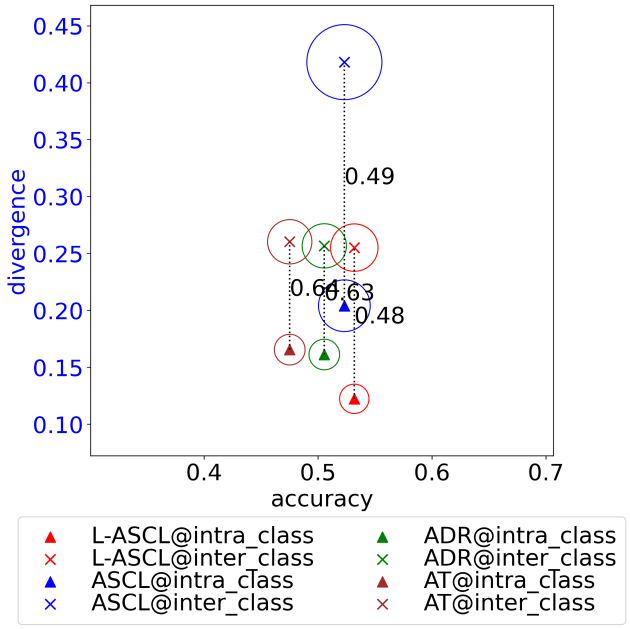


Figure 6: Pairs of Absolute-DIV with corresponding robust accuracy and R-DIV (noted in each line).

B. Additional Analysis of Latent Space Divergence

Experimental setting. The training setting has been described in Section A. Because the intra-class/inter-class divergences are averagely calculated on all N^2 pairs of latent representations which is over our computational capacity, therefore, we alternately calculate these divergences on a mini-batch (128) and take the average over all mini-batches.

Additional evaluation. In addition to the comparison in Section 4.2, we provide a further evaluation on R-DIV and robust accuracy on the CIFAR10 dataset with ResNet20 architecture, under PGD attack $\{\epsilon = 8/255, \eta = 2/255, k = 250\}$ as shown in Figure 6. It can be observed that (i) the value of R-DIV decreases in order from AT (0.64), ADR (0.63), ASCL (0.49), Leaked-ASCL (0.48), respectively. On the other hand, the robust accuracy increases in the same order. (ii) ASCL has much higher absolute intra-class divergence and inter-class divergence than ADR and AT methods, however, ASCL has much lower R-DIV comparing with two baseline methods, therefore, explaining its higher robust accuracy. This result is similar with the comparison on Figure 2.a and the observation O2-i in the main paper and further confirm our conclusion such that "the robustness varies inversely with the relative intra-class divergence between benign images and their adversarial examples".

t-SNE visualization. In addition to the quantitative evaluation as provided in Section 4.4 in the main paper, we provide a qualitative comparison via the t-SNE visualization as shown in Figure 7. The experiments have been conducted on the CIFAR10 dataset with ResNet20 architecture under PGD attack $\{\epsilon = 8/255, \eta = 2/255, k = 250\}$. We visualize latent representations of 100 adversarial examples in addition to 1000 natural samples of the CIFAR10 dataset. It can be seen that: (i) In the NAT model as Figure 7a, the latent representations of natural images are well separate, which explains the high natural accuracy. However, the adversarial examples also are well separate and lay on the high confident area of each class (low entropy). It indicates that, adversarial examples fool the natural model easily with very high confident. (ii) In the AT model as Figure 7b, the latent representations of natural images are less detached, which explains the lower natural accuracy than the NAT model. The adversarial examples distribute *randomly inside* each cluster. The predictions of natural images and adversarial examples have higher entropy which means that the model is less confident. (iii) In our ASCL and Leaked-ASCL as Figure 7c, 7d, the latent representations of natural images are better distinguishable among classes. More specifically, the adversarial examples' representations lay in *the boundary* of each cluster, which has higher entropy than those of natural images.

C. Additional Experimental Results

C.1. Projection Head in the context of AML

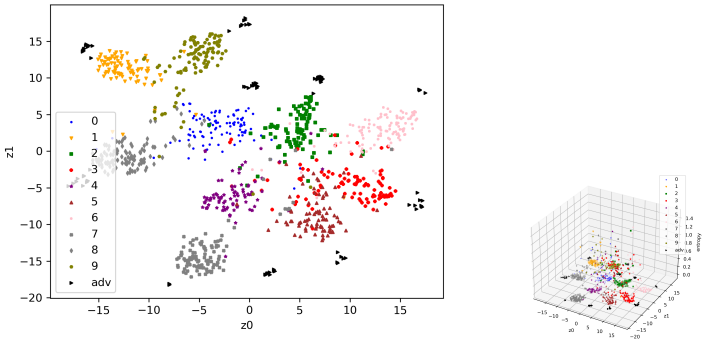
In this section we provide an additional ablation study to further understand the effect of the projection head in the context of AML. We apply our methods (ASCL and Leaked ASCL) with three options of the projection head as shown in Figure 8:

- A projection head with only single linear layer $\tilde{\mathbf{z}} = p^1(\mathbf{z}) = W^1(\mathbf{z})$ with layer weight $W^1 \in \mathcal{R}^{h \times \tilde{h}}$, where $h(\tilde{h})$ is the dimensionality of latent $\mathbf{z}(\tilde{\mathbf{z}})$. We choose $\tilde{h} = 128$ in our experiments.
- A projection head with two fully connected layers without bias $\tilde{\mathbf{z}} = p^2(\mathbf{z}) = W^2\left(Relu(W^1(\mathbf{z}))\right)$ with layer weight $W^1 \in \mathcal{R}^{h \times 200}$ and $W^2 \in \mathcal{R}^{200 \times 128}$ and
- Identity mapping $\tilde{\mathbf{z}} = \mathbf{z}$.

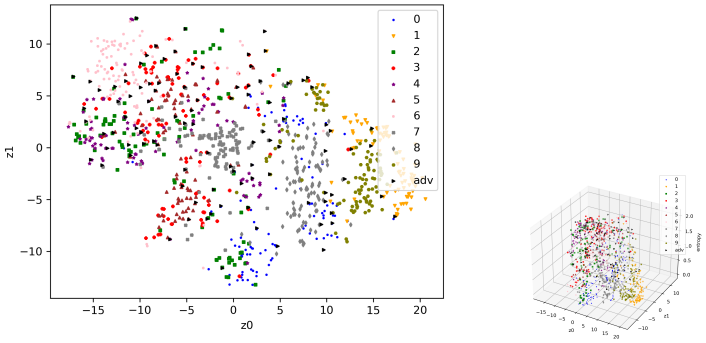
Table 5 shows the performances of three options on the CIFAR10 dataset with ResNet20 architecture. We observe that the linear projection head $p^1()$ is better than the identity mapping on both natural accuracy (by around 1%) and robust accuracy (on average 0.7%). In contrast, the non-linear projection head $p^2()$ reduces the robust accuracy on average 0.5%.

The improvement on the natural accuracy concurs with the finding in Chen et al. (2020) which can be explained by the fact that the projection head helps to reduce the dimensionality to apply the contrastive loss more efficiently. As shown in Section B.4 in Chen et al. (2020) that even using the same output size, the weight of the projection head has relatively few large eigenvalues, indicating that it is approximately low-rank.

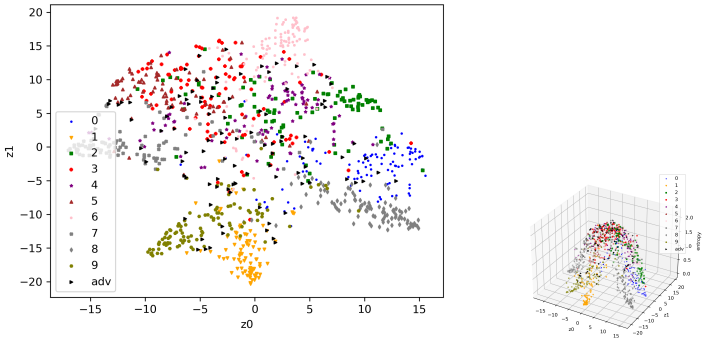
On the other hand, the effect of the projection head to the robust accuracy is due to its non-linearity. Figure 8a



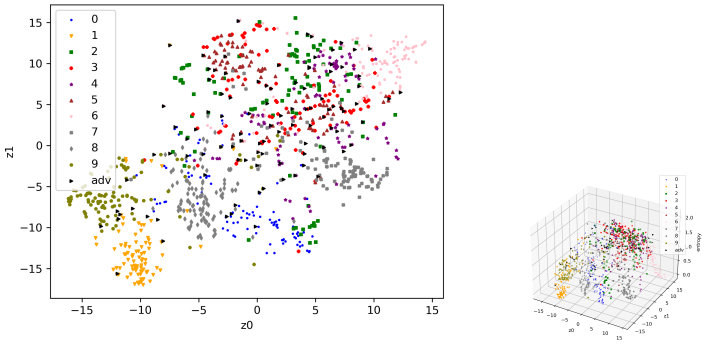
(a) NAT



(b) AT

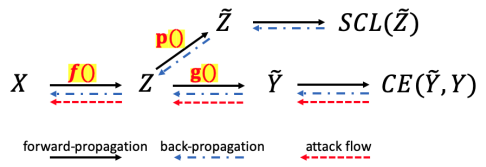


(c) ASCL

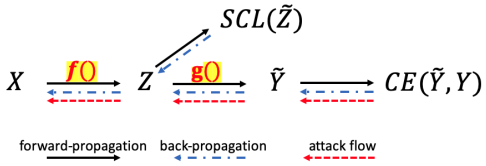


(d) Leaked-ASCL

Figure 7: t-SNE visualization of the latent space. Experiment on the CIFAR10 dataset with ResNet20 architecture. In each subfigure, the black-triangles represents for the adversarial examples. The left-subfigure is 2D t-SNE visualization while the right-subfigure is 2D t-SNE with entropy of prediction in the z-axis.



(a) with the projection head



(b) without the projection head

Figure 8: Training/Attack flows with/without the projection head

Table 5
Performance comparison with/without the projection head $p()$ on the CIFAR10 dataset with ResNet20 architecture. $p^1()$ and $p^2()$ represent for the projection head with one layer and two layers respectively.

	Nat.	PGD	AA
ASCL without $p()$	76.4	52.7	40.9
ASCL with $p^1()$	77.3	**53.3**	**41.3**
ASCL with $p^2()$	76.6	52.3	39.7
(Leaked)ASCL without $p()$	75.5	53.7	42.0
(Leaked)ASCL with $p^1()$	76.5	**54.1**	**42.3**
(Leaked)ASCL with $p^2()$	75.7	52.9	41.1

demonstrates the training flow and attack flow on our framework with the projection head. The contrastive loss $\mathcal{L}^{SCL}$ is applied in the projected layer $\tilde{\mathbf{z}}$ which induces the compactness on the projected layer but not the intermediate layer $\mathbf{z}$. Therefore, when using a non-linear projection head (e.g., p^2), the compactness in the intermediate layer is weaker than the projected layer. For example, a relationship $\left\|\tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_j\right\|_p \leq \left\|\tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_k\right\|_p$ in the projected layer can not imply a relationship $\left\|\mathbf{z}_i - \mathbf{z}_j\right\|_p \leq \left\|\mathbf{z}_i - \mathbf{z}_k\right\|_p$ in the intermediate layer. It explains why using the non-linear projection head reduces the effectiveness of the SCL to the adversarial robustness.

C.2. Contribution of each component in ASCL

We provide an ablation study to investigate the contribution of each of ASCL's components to the performance and emphasize the importance of our SCL component. We experiment on the CIFAR10 dataset with two architectures, i.e., ResNet18 and WideResNet-34-10 (WRN). There are two remarks that can be observed from Table 6 such that:

(i) Using original SCL slightly improves the adversarial robustness against weak adversarial attacks but cannot defend strong ones. More specifically, the robust accuracy against PGD with $\epsilon = 8/255, k = 5, \eta = 2/255$ is 1.2%

Table 6
Ablation study on the CIFAR10 dataset with different architectures. *orgSCL* represents the original SCL version with two standard data-augmentations. **Bold** numbers indicate there are improvements over the previous settings (i.e., only using ADV or using both ADV and VAT) when adding our SCL.

	Model	Nat.	PGD	AA
orgSCL	ResNet18	93.80	0.0	0.0
ADV	ResNet18	82.75	52.95	48.81
ADV+SCL	ResNet18	**85.02**	**53.40**	**50.31**
ADV+VAT	ResNet18	83.73	53.00	49.39
ADV+VAT+SCL	ResNet18	**84.54**	**54.29**	**49.66**
ADV	WRN	84.93	55.04	51.12
ADV+SCL	WRN	**87.70**	54.05	**52.80**
ADV+VAT	WRN	85.96	54.76	51.87
ADV+VAT+SCL	WRN	**87.12**	**55.93**	**52.53**

while that for non-defence model is 0.0%. However, the robust accuracy drops to 0.0% against stronger attacks, i.e., PGD with $\epsilon = 8/255, k = 250, \eta = 2/255$ or Auto-Attack. A similar observation was observed in Kim et al. (2020) when the original SCL only achieves 0.08% robust accuracy. The result shows that while the original contrastive learning induces weak robustness in DNN models as our analysis in Section 2, directly adopting contrastive learning into AML hardly improves the adversarial robustness against strong attacks which emphasizes the importance of our adaptions.

(ii) Adding SCL significantly improves the natural performance and adversarial robustness of the model. More specifically, with ResNet18 architecture, adding *SCL* to *ADV* can gain improvements of 2.3% of natural accuracy and 1.5% of robust accuracy against Auto-Attack. With WideResNet architecture, the improvements of natural/robust accuracies are 2.7% and 1.7%, respectively. Similar improvements can be observed when adding *SCL* to *ADV+VAT*. More specifically, the gaps of natural accuracy with/without *SCL* are 0.8% and 1.1% in experiments with ResNet18 and WideResNet, respectively. These gaps of robust accuracy against Auto-Attack are 0.3% and 0.7%.

We provide an additional experiment to further understand the contribution of each component in our framework. Table 7 shows the result on the CIFAR10 dataset with ResNet20 architecture. We observe that using SCL alone can helps to improve the natural accuracy, but enforcing the contrastive loss too much reduces the effectiveness. On the other hand, increasing the VAT's weight increases the robustness but significantly reduces the natural performance which concurs with the finding in Zhang et al. (2019). Therefore, to balance the trade-off between natural accuracy and robustness, we choose $\lambda^{SCL} = 1, \lambda^{VAT} = 2$ as the default setting in our framework.

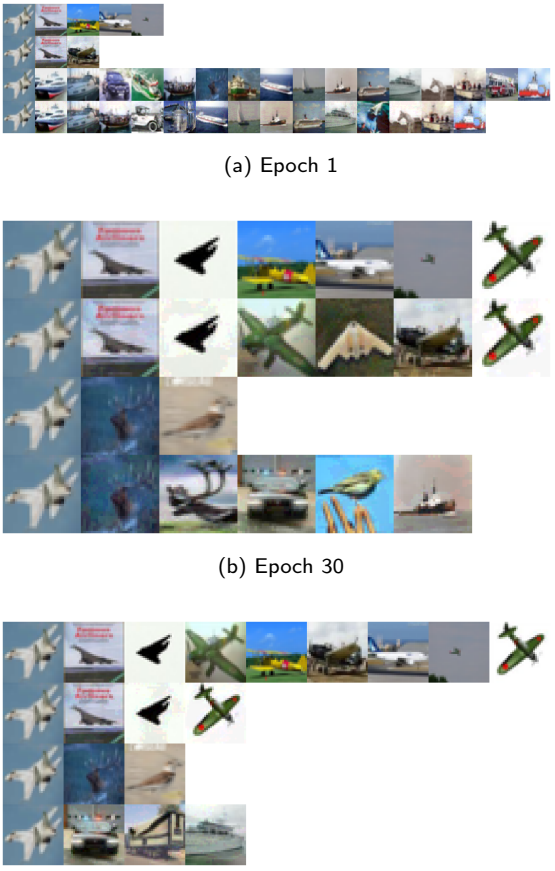
C.3. Global and Local Selections

We provide an example of selected positive and negative samples which have been chosen by the Leaked-Local Selection as Figure 9. It can be seen that, with the same query image, the corresponding negatives and positives have been

Table 7
Ablation study with different parameter settings on the CIFAR10 dataset with ResNet20.

	Nat.	PGD	AA
$\lambda^{SCL}=0, \lambda^{VAT}=0$	78.8	48.1	36.1
$\lambda^{SCL}=1, \lambda^{VAT}=0$	80.1	46.5	34.7
$\lambda^{SCL}=2, \lambda^{VAT}=0$	79.5	46.7	34.7
$\lambda^{SCL}=3, \lambda^{VAT}=0$	79.6	45.8	34.4
$\lambda^{SCL}=4, \lambda^{VAT}=0$	79.2	45.6	34.3
$\lambda^{SCL}=0, \lambda^{VAT}=1$	77.4	50.6	38.2
$\lambda^{SCL}=0, \lambda^{VAT}=2$	75.4	53.0	40.0
$\lambda^{SCL}=0, \lambda^{VAT}=3$	73.3	54.4	42.3
$\lambda^{SCL}=0, \lambda^{VAT}=4$	71.2	55.0	43.1

selected differently overtime. More specifically, at the beginning of training progress (epoch 1, Figure 9a), only few positive images (2-4 images) were picked, while those of negatives are larger (around 14-16 images). Correlating with the model performance, the number of positive images increases while the number of negative images decreases. At the end of training progress (epoch 200, Figure 9c) there are 8 natural images and 3 adversarial images in the positive set, while those in the negative set are 2 natural images and 3 adversarial images. The changing of positives/negatives in this example is inline with the statistic as in Figure 3b in the main paper. In addition, given an anchor image $\mathbf{x}_i$, the natural image $\mathbf{x}_j$ and adversarial image $\mathbf{x}_j^a$ ($j \neq i$) have been treated independently as in Table 1 in the main paper, therefore, we get more flexible in the positive and negative set, for example, only one of $\mathbf{x}_j$ or $\mathbf{x}_j^a$ has been selected as a negative (or a positive) as in Figure 9.



(a) Epoch 1

(b) Epoch 30

(c) Epoch 200

Figure 9: Positive and negative samples from the Leaked Local Selection strategy. In each image, the first column represents the anchor followed by its positive and negative samples. Row 1 and 2 represent the natural and adversarial positive samples respectively. Row 3 and 4 represent the natural and adversarial negative samples respectively.

D. Background and Related works

In this section, we present a fundamental background and related works to our approach. First, we introduce well-known contrastive learning frameworks, followed by a brief introduction of adversarial attack and defense methods. We then provide a comparison of our approach with defense methods on a latent space, especially, those integrated with contrastive learning frameworks.

D.1. Contrastive Learning

D.1.1. General formulation

Self-Supervised Learning (SSL) became an important tool that helps Deep Neural Networks exploit structure from gigantic unlabeled data and transfers it to downstream tasks. The key success factor of SSL is choosing a pretext task that heuristically introduces interaction among different parts of the data (e.g., CBOW and Skip-gram Mikolov, Sutskever, Chen, Corrado and Dean (2013), predicting rotation Gidaris, Singh and Komodakis (2018)). Recently, Self-Supervised Contrastive Learning (SSCL) with contrastive learning as the pretext task surpasses other SSL frameworks and nearly achieves supervised-learning's performance. The main principle of SSCL is to introduce a contrastive correlation among visual representations of positives ('similar') and negatives ('dissimilar') with respect to an anchor one. There are several SSCL frameworks have been proposed (e.g., MoCo He et al. (2020), BYOL Grill, Strub, Altché, Tallec, Richemond, Buchatskaya, Doersch, Avila Pires, Guo, Gheshlaghi Azar et al. (2020), CURL Srinivas, Laskin and Abbeel (2020)), however, in this section, we mainly introduce the SSCL in Chen et al. (2020) which had been integrated with adversarial examples to improve adversarial robustness in Kim et al. (2020); Jiang et al. (2020) followed by the Supervised Contrastive Learning (SCL) Khosla et al. (2020) which has been used in our approach.

Consider a batch of N pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ of benign images and their labels. With two random transformations $\mathcal{T}, \mathcal{A}$ we have a set of transformed images $\{\mathbf{x}_i^{\mathcal{T}}, \mathbf{x}_i^{\mathcal{A}}, \mathbf{y}_i\}_{i=1}^N$. The general formulation of contrastive learning as follow:

$$\mathcal{L}^{\text{CL}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i^{\mathcal{T},cl} + \mathcal{L}_i^{\mathcal{A},cl} \tag{7}$$

where $\mathcal{L}_i^{\mathcal{T},cl}$ is the contrastive loss w.r.t. the anchor $\mathbf{x}_i^{\mathcal{T}}$:

$$\mathcal{L}_i^{\mathcal{T},cl} = \frac{-1}{\left|\boldsymbol{Z}_i^+\right| + 1} \sum_{\mathbf{z}_j \in \boldsymbol{Z}_i^+ \cup \{\mathbf{z}_i^{\mathcal{A}}\}} \log \frac{e^{\frac{sim(\mathbf{z}_j, \mathbf{z}_i^{\mathcal{T}})}{\tau}}}{\sum_{\mathbf{z}_k \in \boldsymbol{Z}_i^+ \cup \boldsymbol{Z}_i^- \cup \{\mathbf{z}_i^{\mathcal{A}}\}} e^{\frac{sim(\mathbf{z}_k, \mathbf{z}_i^{\mathcal{T}})}{\tau}}} \tag{8}$$

and $\mathcal{L}_i^{\mathcal{A},cl}$ is the contrastive loss w.r.t. the anchor $\mathbf{x}_i^{\mathcal{A}}$:

$$\mathcal{L}_i^{\mathcal{A},cl} = \frac{-1}{\left|\boldsymbol{Z}_i^+\right| + 1} \sum_{\mathbf{z}_j \in \boldsymbol{Z}_i^+ \cup \{\mathbf{z}_i^{\mathcal{T}}\}} \log \frac{e^{\frac{sim(\mathbf{z}_j, \mathbf{z}_i^{\mathcal{A}})}{\tau}}}{\sum_{\mathbf{z}_k \in \boldsymbol{Z}_i^+ \cup \boldsymbol{Z}_i^- \cup \{\mathbf{z}_i^{\mathcal{T}}\}} e^{\frac{sim(\mathbf{z}_k, \mathbf{z}_i^{\mathcal{A}})}{\tau}}} \tag{9}$$

The formulation shows the general principle of contrastive learning such that: (i) $\boldsymbol{Z}_i^+ \cup \boldsymbol{Z}_i^- \cup \{\mathbf{z}_i^{\mathcal{A}}, \mathbf{z}_i^{\mathcal{T}}\} = \{\mathbf{z}_j^{\mathcal{T}}, \mathbf{z}_j^{\mathcal{A}}\}_{j=1}^N \; \forall i \in [1, N]$ where $\boldsymbol{Z}_i^+$ and $\boldsymbol{Z}_i^-$ are positive and negative sets which are defined differently depending on self-supervised/supervised setting, (ii) without loss of generality, in Equation 8, the similarity $e^{\frac{sim(\mathbf{z}_j, \mathbf{z}_i^{\mathcal{T}})}{\tau}}$ between the anchor $\mathbf{z}_i^{\mathcal{T}}$ and a positive sample $\mathbf{z}_j \in \boldsymbol{Z}_i^+ \cup \{\mathbf{z}_i^{\mathcal{A}}\}$ has been normalized with sum of all possible pairs between the anchor and the union set of $\boldsymbol{Z}_i^+ \cup \boldsymbol{Z}_i^- \cup \{\mathbf{z}_i^{\mathcal{A}}\}$ to ensures that the log argument is not higher than 1, (iii) the contrastive loss in Equation 8 pulls anchor representation $\mathbf{z}_i^{\mathcal{T}}$ and the positives' representations $\boldsymbol{Z}_i^+ \cup \{\mathbf{z}_i^{\mathcal{A}}\}$ close together while pushes apart those of negatives $\boldsymbol{Z}_i^-$.

Explanation for our Formulation. It is worth noting that, our derivation shows the general formulation of the contrastive learning which can be adapted to SSCL Chen et al. (2020), SCL Khosla et al. (2020) or our Local ASCL by defining the positive and negative sets differently. Moreover, by using terminologies positive set $\boldsymbol{Z}_i^+$ and those sample from the same instance $\{\mathbf{z}_i^{\mathcal{T}}, \mathbf{z}_i^{\mathcal{A}}\}$ separately, we emphasize the importance of the anchor's transformation which stand out other positives. Last but not least, our derivation normalizes the contrastive loss in Equation 7 to the same scale with the cross-entropy loss and the VAT loss as in Section 3, which helps to put all terms together appropriately.

Self-Supervised Contrastive Learning. In SSCLChen et al. (2020), the positive set (excluding those samples from the same instance $\mathbf{z}_i^{\mathcal{A}}, \mathbf{z}_i^{\mathcal{T}}$) $\boldsymbol{Z}_i^+ = \emptyset$ ($\left|\boldsymbol{Z}_i^+\right| = 0$) while the negative set $\boldsymbol{Z}_i^- = \{\mathbf{z}_j^{\mathcal{T}}, \mathbf{z}_j^{\mathcal{A}} \mid j \neq i\}$ which includes all other samples except those from the same instance $\mathbf{z}_i^{\mathcal{A}}, \mathbf{z}_i^{\mathcal{T}}$. In this case, the formulation of SSCL as follow:

$$\mathcal{L}_i^{\mathcal{T},sscl} = -\log \frac{e^{\frac{sim(\mathbf{z}_j, \mathbf{z}_i^{\mathcal{T}})}{\tau}}}{\sum_{\mathbf{z}_k \in \boldsymbol{Z}_i^- \cup \{\mathbf{z}_i^{\mathcal{A}}\}} e^{\frac{sim(\mathbf{z}_k, \mathbf{z}_i^{\mathcal{T}})}{\tau}}} \tag{10}$$

and

$$\mathcal{L}_i^{\mathcal{A},sscl} = -\log \frac{e^{\frac{sim(\mathbf{z}_j, \mathbf{z}_i^{\mathcal{A}})}{\tau}}}{\sum_{\mathbf{z}_k \in \boldsymbol{Z}_i^- \cup \{\mathbf{z}_i^{\mathcal{T}}\}} e^{\frac{sim(\mathbf{z}_k, \mathbf{z}_i^{\mathcal{A}})}{\tau}}} \tag{11}$$

Supervised Contrastive Learning. The SCL framework leverages the idea of contrastive learning with the presence of label supervision to improve the regular cross-entropy loss. The positive set and the negative set are $\boldsymbol{Z}_i^+ = \{\mathbf{z}_j^{\mathcal{T}}, \mathbf{z}_j^{\mathcal{A}} \mid j \neq i, \boldsymbol{y}_j = \boldsymbol{y}_i\}$ and $\boldsymbol{Z}_i^- = \{\mathbf{z}_j^{\mathcal{T}}, \mathbf{z}_j^{\mathcal{A}} \mid j \neq i, \boldsymbol{y}_j \neq \boldsymbol{y}_i\}$, respectively. As mentioned in Khosla et al. (2020), there is a major advantage of SCL compared with SSCL in the context of regular machine learning. Unlike SSCL in which each anchor has only single positive sample, SCL takes advantages of the labels to have many positives in the same batch size N. This strategy helps to reduce the false negative cases in SSCL when two samples in the same class are pushed apart. As shown in Khosla et al. (2020), the SCL training is more stable than SSCL and also achieves a better performance.

D.1.2. Important factors for Contrastive Learning

Data augmentation. Chen et al. Chen et al. (2020) empirically found that SSCL needs stronger data augmentation than supervised learning. While the SSCL's performance experienced a huge gap of 5% with different data augmentation (Table 1 in Chen et al. (2020)), the supervised performance was not changed much with the same set of augmentation. Therefore, in our paper, to reduce the space of hyper-parameters we use only one adversarial transformation $\mathcal{A}$(e.g., PGDMadry et al. (2018) or TRADESZhang et al. (2019)) while using the identity transformation $\mathcal{T}$, $\mathbf{x}_i^{\mathcal{T}} = \mathbf{x}_i$ ($\mathbf{z}_i^{\mathcal{T}} = \mathbf{z}_i$), and let the investigation of using different data augmentations for future works.

Batch size. As shown in Figure 9 in Chen et al. (2020), the batch size is an important factor that strongly affects the performance of the contrastive learning framework. A larger batch size comes with larger positive and negative sets, which helps to generalize the contrastive correlation better and therefore improves the performance. He et al. He et al. (2020) proposed a memory bank to store the previous batch information which can lessen the batch size issue. In our framework, because of the limitation on computational resources, we only tried with a small batch size (128) which likely limits the contribution of our methods.

Projection head. Normally, the representation vector which is the output of the encoder network has very high dimensionality, e.g., the final pooling layer in ResNet-50 and ResNet-200 has 2048 dimensions. Therefore, applying contrastive learning directly on this intermediate layer is less effective. Alternatively, CL frameworks usually use a projection network $p()$ to project the normalized representation vector $\mathbf{z}$ into a lower dimensional vector $\tilde{\mathbf{z}} = p(\mathbf{z})$ which is more suitable for computing the contrastive loss. To avoid overparameterized, CL frameworks usually choose a small projection head with only one or two fully-connected layers.

D.2. Adversarial attack

Projected Gradient Decent (PGD). is an iterative version of the FGSM attack Goodfellow et al. (2015) with random initialization Madry et al. (2018). It first randomly initializes an adversarial example in a perturbation ball by adding uniform noise to a clean image, followed by multiple steps of one-step gradient ascent, at each step projecting onto the perturbation ball. The formula for the one-step update is as follows:

$$x_a^{t+1} = \text{Proj}_{B_\epsilon(x)}(x_a^t + \eta \ \text{sign}\,(\nabla \ell(x, y, \theta)) \tag{12}$$

where $B_\epsilon(x) \triangleq \left\{x' : \|x' - x\| < \epsilon\right\}$ is the perturbation ball with radius ϵ around $x$and η is the gradient scale for each step update.

Auto-Attack. Even the most popular attack, PGD can still fail in some extreme cases Croce, Rauber and Hein (2019) because of two issues: (i) fixed step size η which leads to sub-optimal solutions and (ii) the sensitivity of a gradient to the scale of logits in the standard cross-entropy loss. Auto-Attack Croce and Hein (2020) proposed two variants of PGD to deal with these potential issues by (i) automatically selecting the step size across iterations (ii) an alternative logit loss which is both shift and rescaling invariant. Moreover, to increase the diversity among the attacks used, Auto-Attack combines two new versions of PGD with the white-box attack FAB Croce and Hein (2019)and the blackbox attack Square Attack Andriushchenko, Croce, Flammarion and Hein (2020) to form a parameter-free, computationally affordable, and user-independent ensemble of complementary attacks to estimate adversarial robustness. Therefore, besides PGD, Auto-Attack is considered as the new standard evaluation for adversarial robustness.

D.3. Adversarial defense

D.3.1. Adversarial training

Adversarial training (AT) originate in Goodfellow et al. (2015), which proposed incorporating a model's adversarial examples into training data to make the model's loss surface to be smoother, thus, improve its robustness. Despite its simplicity, AT Madry et al. (2018) was among the few that were resilient against attacks other than gave a false sense of robustness because of the obfuscated gradient Athalye et al. (2018). To continue its success, many AT's variants have been proposed including (1) different types of adversarial examples (e.g., the worst-case examples Goodfellow et al. (2015) or most divergent examples Zhang et al. (2019)), (2) different searching strategies (e.g., non-iterative FGSM, Rand FGSM with a random initial point or PGD with multiple iterative gradient descent steps Madry et al. (2018)), (3) additional regularizations, e.g., adding constraints in the latent space Zhang and Wang (2019); Bui et al. (2020), (4) difference in model architecture, e.g., activation function Xie et al. (2020) or ensemble models Pang et al. (2019).

D.3.2. Defense with a latent space

Unlike an input space $\boldsymbol{X}$, a latent space $\boldsymbol{Z}$ has a lower dimensionality and a higher mutual information with the prediction space than the input one $I(Z, Y) \geq I(X, Y)$ Tishby and Zaslavsky (2015). Therefore, defense with the latent space has particular characteristics to deal with adversarial attacks notably Zhang and Wang (2019); Bui et al. (2020); Mao et al. (2019); Xie et al. (2019); Samangouei, Kabkab and Chellappa (2018). For example, DefenseGAN Samangouei et al. (2018) used a pretrained GAN which emulates the data distribution to generate a denoised version of an adversarial example. On the other hand, instead of removing noise in the input image, Xie et al. Xie et al. (2019) attempted to remove noise in the feature space by using non-local means as a denoising block. However, these works were criticized by Athalye et al. (2018) as being easy to attack by approximating the backward gradient signal.

D.3.3. Defense with contrastive learning

The idea of defense with contrastive correlation in the latent space can be traced back to Mao et al. (2019) which proposed an additional triplet regularization to adversarial training. However, the triplet loss can only handle one positive and negative at a time, moreover, requires computationally expensive hard negative mining Schroff, Kalenichenko and Philbin (2015). As discussed in Khosla et al. (2020), the triplet loss is a special case of the contrastive loss when the number of positives and negatives are each one and has lower performance in general than the contrastive loss. Recently, Jiang et al. (2020); Kim et al. (2020) integrated SSCL Chen et al. (2020) to learn unsupervised robust representations for improving robustness in unsupervised/semi-supervised setting. Specifically, both methods proposed a new kind of adversarial examples which is based on the SSCL loss instead of regular cross-entropy loss Goodfellow et al. (2015) or KL divergence Zhang et al. (2019). By adversarially pre-training with these adversarial examples, the encoder is robust against the instance-wise attack and obtains comparable robustness to supervised adversarial training as reported in Kim et al. (2020). On the other hand, Jiang et al. Jiang et al. (2020) proposed three options of pre-training. However, their best method made use of two adversarial examples that requires a much higher computational cost to generate. Although these above works have the similar general idea of using contrastive learning to improve adversarial robustness with ours, we choose to compare our methods with RoCL-AT/TRADES in Kim et al. (2020) which is most close to our problem setting. More specifically, after pre-training

phase with adversarial examples w.r.t. the contrastive loss, RoCL-AT/TRADES apply fine-tuning with standard supervised adversarial training, which requires full label. We use the reported result as in Table 1 in Kim et al. (2020) which used a larger batch size (256). It is a worth noting that the best reported version RoCL-AT-SS achieves 91.34% natural accuracy and 49.66% robust accuracy is a fine-tuned on a ImageNet pretrained model with self-supervised loss (e.g., SimCLR Chen et al. (2020)), therefore, is not as a reference for comparison.

Most closely related to our work is Bui et al. (2020) which also aims to realize the compactness in latent space to improve the robustness in supervised setting. They proposed a label weighting technique that sets the positive weight to the divergence of two examples in the same class and negative weight in any other cases. Therefore, when minimizing the divergence loss with label weighting, the divergences of those in the same class (positives) are encouraged to be close together, while those of different classes (negatives) to be distant.

References

Akhtar, N., Mian, A., 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access 6, 14410–14430.

Andriushchenko, M., Croce, F., Flammarion, N., Hein, M., 2020. Square attack: a query-efficient black-box adversarial attack via random search, in: European Conference on Computer Vision, Springer. pp. 484–501.

Athalye, A., Carlini, N., Wagner, D., 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, in: International Conference on Machine Learning, pp. 274–283.

Bui, A., Le, T., Zhao, H., Montague, P., deVel, O., Abraham, T., Phung, D., 2020. Improving adversarial robustness by enforcing local and global compactness. arXiv preprint arXiv:2007.05123 .

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., Kurakin, A., 2019. On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705 .

Carlini, N., Wagner, D., 2017. Towards evaluating the robustness of neural networks, in: 2017 ieee symposium on security and privacy (sp), IEEE. pp. 39–57.

Chapelle, O., Zien, A., 2005. Semi-supervised classification by low density separation., in: AISTATS, pp. 57–64.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 .

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., Hein, M., 2020. Robustbench: a standardized adversarial robustness benchmark. arXiv preprint arXiv:2010.09670 .

Croce, F., Hein, M., 2019. Minimally distorted adversarial examples with a fast adaptive boundary attack. arXiv preprint arXiv:1907.02044 .

Croce, F., Hein, M., 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. arXiv preprint arXiv:2003.01690 .

Croce, F., Rauber, J., Hein, M., 2019. Scaling up the randomized gradient-free adversarial attack reveals overestimation of robustness using established attacks. International Journal of Computer Vision , 1–19.

Cui, J., Liu, S., Wang, L., Jia, J., 2021. Learnable boundary guided adversarial training. International Conference on Computer Vision .

Frankle, J., Schwab, D.J., Morcos, A.S., et al., 2020. Are all negatives created equal in contrastive instance discrimination? arXiv preprint arXiv:2010.06682 .

Gidaris, S., Singh, P., Komodakis, N., 2018. Unsupervised representation learning by predicting image rotations, in: International Conference on Learning Representations.

Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples, in: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. URL: http://arxiv.org/abs/1412.6572.

Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al., 2020. Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems 33.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Jiang, Z., Chen, T., Chen, T., Wang, Z., 2020. Robust pre-training by adversarial contrastive learning, in: NeurIPS.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D., 2020. Supervised contrastive learning. arXiv preprint arXiv:2004.11362 .

Kim, M., Tack, J., Hwang, S.J., 2020. Adversarial self-supervised contrastive learning. Advances in Neural Information Processing Systems 33.

Krizhevsky, A., et al., 2009. Learning multiple layers of features from tiny images .

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., Jana, S., 2019. Certified robustness to adversarial examples with differential privacy, in: 2019 IEEE Symposium on Security and Privacy (SP), IEEE. pp. 656–672.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations.

Mao, C., Zhong, Z., Yang, J., Vondrick, C., Ray, B., 2019. Metric learning for adversarial robustness, in: Advances in Neural Information Processing Systems, pp. 480–491.

Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B., 2017. On detecting adversarial perturbations. arXiv preprint arXiv:1702.04267 .

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, pp. 3111–3119.

Miyato, T., Maeda, S., Koyama, M., Ishii, S., 2019. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 41, 1979–1993.

Pang, T., Xu, K., Du, C., Chen, N., Zhu, J., 2019. Improving adversarial robustness via promoting ensemble diversity, in: International Conference on Machine Learning, pp. 4970–4979.

Pang, T., Yang, X., Dong, Y., Su, H., Zhu, J., 2020. Bag of tricks for adversarial training, in: International Conference on Learning Representations.

Samangouei, P., Kabkab, M., Chellappa, R., 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605 .

Schroff, F., Kalenichenko, D., Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 815–823.

Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T., 2019. Adversarial training for free!, in: Advances in Neural Information Processing Systems, pp. 3353–3364.

Srinivas, A., Laskin, M., Abbeel, P., 2020. Curl: Contrastive unsupervised representations for reinforcement learning. arXiv preprint arXiv:2004.04136 .

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P., 2020. What makes for good views for contrastive learning?, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. URL: https://proceedings.neurips.cc/paper/2020/hash/

4c2e5eaae9152079b9e95845750bb9ab-Abstract.html.

Tishby, N., Zaslavsky, N., 2015. Deep learning and the information bottleneck principle, in: 2015 IEEE Information Theory Workshop (ITW), IEEE. pp. 1–5.

Xie, C., Tan, M., Gong, B., Yuille, A., Le, Q.V., 2020. Smooth adversarial training. arXiv preprint arXiv:2006.14536 .

Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K., 2019. Feature denoising for improving adversarial robustness, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 501–509.

Zagoruyko, S., Komodakis, N., 2016. Wide residual networks, in: British Machine Vision Conference 2016, British Machine Vision Association.

Zhang, H., Wang, J., 2019. Defense against adversarial attacks using feature scattering-based adversarial training, in: Advances in Neural Information Processing Systems, pp. 1829–1839.

Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I., 2019. Theoretically principled trade-off between robustness and accuracy. arXiv preprint arXiv:1901.08573 .

Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., Kankanhalli, M., 2020. Attacks which do not kill training make adversarial learning stronger, in: International Conference on Machine Learning, PMLR. pp. 11278–11287.

3.5 Concluding Remarks

In this chapter, we have introduced our novel contributions to the field, centered around enhancing the adversarial robustness of models through representation learning, as outlined in the two papers Bui et al. (2020) and Bui et al. (2021a). Our primary objective has been to address the fundamental question: "What are the key characteristics of a representation that bolster robustness?"

Our journey has led us to a crucial insight: a robust representation must encompass both local and global information within the data manifold. This synthesis is pivotal for strengthening model resilience against adversarial attacks. More importantly, we have demonstrated that measuring the divergence in the latent space, particularly through relative distance metrics like the contrastive loss, surpasses absolute distance metrics like the triplet loss in terms of effectiveness.

Moreover, it's worth highlighting that our work, has encouraged the subsequent research in Le et al. (2022) which further explored the relationships between the latent divergence and adversarial robustness. Our theory aligns with the empirical results of our previous work in Bui et al. (2021a) and Bui et al. (2020), providing further insights into the complex interplay between representation learning, adversarial attacks, and defenses.

Chapter 4

Ensemble Learning Approaches to Adversarial Robustness

4.1 Introduction

In this chapter, we present our contributions towards improving adversarial robustness through the lens of ensemble learning, as introduced in Bui et al. (2021b) and Bui et al. (2023). In the work Bui et al. (2021b), we proposed a novel collaboration strategy to enhance diversity among ensemble members, thereby reducing the transferability of adversarial examples between them. Our main principle was to prioritize the correct predictions of one model on a given adversarial example while discouraging other models from making unanimous predictions. This collaboration strategy effectively minimized the negative impact of incorrect predictions and ensured accurate predictions from the ensemble model. Through extensive experiments, we demonstrated the effectiveness of our strategy in improving the robustness of ensemble models against adversarial attacks. Furthermore, we gained a deeper understanding of the relationship between transferability and overall robustness in ensemble learning.

Previous work had shown that adversarial examples that fool all ensemble members are crucial for improving the robustness of ensemble models. Building upon this observation, we extended our investigations in Bui et al. (2023) by presenting a novel method for generating transferable adversarial examples that lie in the joint insecure region of all ensemble members. To achieve this, we considered the adversarial generation task as

a multi-objective optimization problem, aiming to find a Pareto optimality that maximizes multiple objectives simultaneously. However, directly applying multi-objective optimization to generate adversarial examples was not satisfactory due to the dominating effect of one task over other tasks.

To address this issue, we proposed a novel framework named Task Oriented Multi-Objective Optimization (TA-MOO) with multi-objective adversarial generations as the demonstrating applications. Our key principle was to favor the unsuccessful tasks while maintaining the success of the successful ones using a novel geometry-based regularization term. We conducted extensive experiments on three adversarial generation tasks and one adversarial training task to demonstrate the effectiveness of our method in generating stronger and more robust adversarial examples. Notably, our method had been demonstrated to be particularly beneficial when combined with adversarial training, offering promising avenues for strengthening the security of deep learning models. The insights provided in this work emphasized the pivotal role of considering multiple objectives in the generation of adversarial examples, ultimately contributing to the advancement of robustness and security in the realm of deep learning.

The major content of this chapter is in the following attached papers:

- **Anh Bui**, Trung Le, He Zhao, Paul Montague, Olivier de Vel, Tamas Abraham, Dinh Phung, *"Improving Ensemble Robustness by Collaboratively Promoting and Demoting Adversarial Robustness"*. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) 2021.

  The code of this paper is released at https://github.com/tuananhbui89/Crossing-Collaborative-Ensemble.

- **Anh Bui**, Trung Le, He Zhao, Quan Tran, Paul Montague, Dinh Phung, *"Generating Adversarial Examples with Task Oriented Multi-Objective Optimization"*. Accepted to the Transactions on Machine Learning Research (TMLR), 2023.

  The code of this paper is released at https://github.com/tuananhbui89/TAMOO.

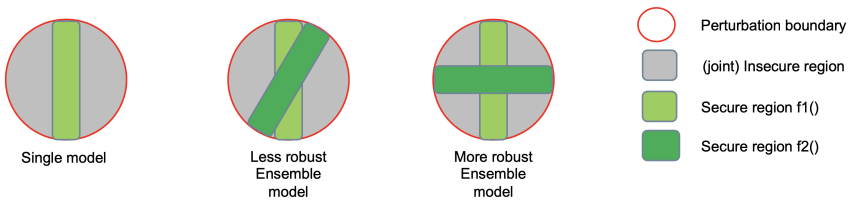


FIGURE 4.1: Principle of Ensemble-based Defenses. (a) The secure/insecure region of each single model. (b,c) The joint secure/insecure region of the ensemble with low diversity and high diversity, respectively.

4.2 Related Work

Adversarial examples have demonstrated their ability to transfer between different models (Papernot et al., 2016a,b), allowing successful attacks on models that are not directly targeted by the adversary. To delve deeper into this phenomenon, Tramèr et al. (2017) conducted the first investigation of the adversarial subspace. They proposed a method to estimate the dimensionality of the space containing adversarial examples, discovering that these examples occupy a contiguous subspace with a large number of dimensions. Furthermore, they observed the high transferability of adversarial examples across diverse models. They found that when two models achieve low error rates on the same test set but exhibit low robustness against adversarial examples, it suggests that their adversarial subspace is likely shared, and the transferability of adversarial examples between the two models is highly probable.

Given the transferability of adversarial examples, the primary research focus of ensemble-based defenses lies in reducing the transferability between ensemble members, aiming to enhance overall robustness, as illustrated in Figure 4.1.

Tramèr et al. (2018) employed perturbations generated from static pre-trained models as augmented data to decouple the generation process of adversarial examples for the target model. The idea behind this approach was that since adversarial examples can transfer between models, perturbations that challenge other models can be useful approximations for maximizing the vulnerability of the target model. However, as reported in Tramèr et al. (2018), this method was primarily designed for black-box attacks and therefore remains vulnerable to white-box attacks.

In fact, the transfer of adversarial examples from static models can have a negative impact. While it helps enhance the model's robustness against black-box attacks, it

simultaneously makes the model more susceptible to white-box attacks. This occurs because the model becomes overfitted to a specific type of adversarial examples and fails to account for adaptive attacks.

On the other hand, Kariyappa and Qureshi (2019) proposed an alignment method to reduce the shared adversarial subspace by aligning the gradients of ensemble members to be diametrically opposed. This was achieved by minimizing the cosine similarity between the gradients of pairs of ensemble members. However, attempting to achieve gradient alignment proves to be unreliable in high-dimensional input spaces and poses challenges when extending it to ensembles with more than two members. As shown in their paper, the method showcased sensitivity to the choice of architecture, particularly in scenarios with a large number of zero values in the gradient.

Furthermore, the method aimed to enhance the diversity among ensemble members by encouraging them to be uncorrelated, with each member's gradient being orthogonal to the gradients of other members. However, in the ensemble learning literature, greater diversity is achieved when ensemble members exhibit negative correlation (Liu and Yao, 1999), as emphasized in our work (Bui et al., 2021b).

Pang et al. (2019) proposed to promote the diversity of non-maximal predictions (i.e., the diversity among softmax probabilities except the highest ones) of the ensemble members as illustrated in Figure 4.2. It was done by maximizing the squared volume spanned by the non-maximal predictions of the ensemble members. Because the non-maximal predictions exclude the highest ones, this method allowed the maximal prediction of each member to be consistent with the true label, and thus will not affect ensemble accuracy. Besides, since the non-maximal predictions correspond to all potential wrong labels, a high diversity of non-maximal predictions indicates that the ensemble members are likely to have different adversarial subspaces. However, later in Tramer et al. (2020) showed that the method was vulnerable to carefully crafted adaptive attacks such as B&B attack (Brendel et al., 2019).

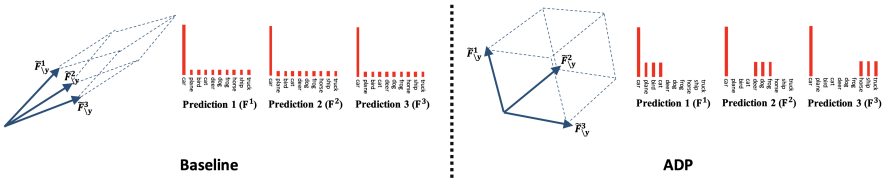


FIGURE 4.2: Illustration of promoting non-maximal diversity among ensemble members. Image source: Pang et al. (2019).

Yang et al. (2020) proposed to distill the features learned by each member corresponding to its vulnerability to adversarial examples. It was done by first learning an adversarial example that is close to a source sample but has a feature close to the input sample's feature. The adversarial example of each member was then used to train other models, hence making the adversarial subspace of each member more non-overlap.

It is worth noting that, our method (Bui et al., 2021b) shared the same motivation with Yang et al. (2020) in the use of adversarial examples to train other models. However, our work was more general that can be applied to any ensemble with untargeted/targeted attacks, while Yang et al. (2020) was only applied to an ensemble of models with the same architecture and with targeted attack only. With a specific setting, our method also could have detection capability that can be used to detect adversarial examples. Two works were developed independently and were submitted to the NeurIPS 2020 conference at the same time but ours got rejected.

Yang et al. (2021) dug further and provided a theoretical analysis of the transferability of adversarial examples between models. They showed that only promoting the orthogonality between gradients of ensemble members is not enough to ensure low transferability. This finding was consistent with our empirical results in Bui et al. (2021b).

4.3 Collaborative Ensemble for Improving Robustness

Improving Ensemble Robustness by Collaboratively Promoting and Demoting Adversarial Robustness

Anh Bui[1], **Trung Le**[1], **He Zhao**[1]
Paul Montague[2], **Olivier deVel**[2], **Tamas Abraham**[2], **Dinh Phung**[1]
[1]Monash University
[2]Defence Science and Technology Group, Australia
tuananh.bui@monash.edu

Abstract

Ensemble-based adversarial training is a principled approach to achieve robustness against adversarial attacks. An important technique of this approach is to control the transferability of adversarial examples among ensemble members. We propose in this work a simple yet effective strategy to collaborate among committee models of an ensemble model. This is achieved via the secure and insecure sets defined for each model member on a given sample, hence help us to quantify and regularize the transferability. Consequently, our proposed framework provides the flexibility to reduce the adversarial transferability as well as to promote the diversity of ensemble members, which are two crucial factors for better robustness in our ensemble approach. We conduct extensive and comprehensive experiments to demonstrate that our proposed method outperforms the state-of-the-art ensemble baselines, at the same time can detect a wide range of adversarial examples with a nearly perfect accuracy.

Introduction

Deep neural networks have experienced great success in many disciplines (I. Goodfellow, Y. Bengio, and Courville, 2016), such as computer vision (K. He et al., 2016), natural language processing and speech processing (Vaswani et al., 2017). However, even the state-of-the-art models are reported to be vulnerable to adversarial attacks (Biggio et al., 2013; I. J. Goodfellow, Shlens, and Szegedy, 2015; Szegedy et al., 2014; N. Carlini and D. Wagner, 2017; Madry et al., 2018; Athalye, Nicholas Carlini, and David Wagner, 2018), which is of significant concern given the large number of applications of deep learning in real-world scenarios. It is thus urgent to develop deep learning models that are robust against different types of adversarial attacks. To this end, several adversarial defense methods have been developed but typically addressing the robustness within a single model (e.g., Papernot, P. D. McDaniel, et al., 2016; Moosavi-Dezfooli, Fawzi, and Frossard, 2016; Madry et al., 2018; Qin et al., 2019; Shafahi et al., 2019). To cater for more diverse types of attacks, recent work, notably (W. He et al., 2017; Tramèr, Kurakin, et al., 2018; Strauss et al., 2017; X. Liu et al., 2018; Pang, Xu, et al., 2019), has shown that ensemble learning can strengthen robustness significantly.

Despite initial success, key principles for ensemble-based adversarial training (EAT) largely remain open. One crucial challenge is to achieve minimum 'transferability' between committee members to increase robustness for the overall ensemble model (Papernot, P. McDaniel, and I. Goodfellow, 2016; Yanpei Liu et al., 2016; Tramèr, Kurakin, et al., 2018; Pang, Xu, et al., 2019; Kariyappa and Qureshi, 2019). In (Kariyappa and Qureshi, 2019), robustness was achieved by aligning the gradient of committee members to be diametrically opposed, hence reducing the shared adversarial spaces (Tramèr, Papernot, et al., 2017), or the transferability. However, the method in (Kariyappa and Qureshi, 2019) was designed for black-box attacks, thus still vulnerable to white-box attacks. Furthermore, attempting to achieve gradient alignment is unreliable for high-dimensional datasets and it is difficult to extend for ensemble with more than two committee members. More recently (Pang, Xu, et al., 2019) proposed to promote the diversity of non-maximal predictions of the committee members (i.e., the diversity among softmax probabilities except the highest ones) to reduce the adversarial transferability among them. Nonetheless, the central concept of transferability has not been systematically addressed.

Our proposed work here will first make the concept of adversarial transferability concrete via the definitions of secure and insecure sets. To reduce the adversarial transferability and increase the model diversity, we aim to make the insecure sets of the committee models as disjoint as possible (i.e., lessening the overlapping of those regions) and challenge those committee members with divergent sets of adversarial examples. In addition, we observe that lessening the adversarial transferability alone is not sufficient to ensure accurate predictions of the ensemble model because the committee member that offers inaccurate predictions might dominate the final decisions. With this in mind, we propose to realize what we call a "transferring flow" by collaborating robustness promoting and demoting operations. Our key principle to coordinate the promoting and demoting operations is to promote the prediction of one model on a given adversarial example and to demote the prediction of another model on this example so as to maximally lessen the negative impact of the wrong predictions and ensure the correct predictions of the ensemble model. Moreover, different from other works (Strauss et al., 2017; Pang, Xu, et al.,

2019; Kariyappa and Qureshi, 2019) which only consider adversarial examples of the ensemble model, the committee members in our ensemble model are exposed to various divergent adversarial example sets, which inspire them to become gradually more divergent. Interestingly, by strengthening demoting operations, our method is capable to assist better detection of adversarial examples. In brief, our contributions in this work include:

- We propose a simple but efficient collaboration strategy to reduce the transferability among ensemble members.
- We propose two variants of our method: the robust oriented variant, which helps to improve the adversarial robustness and the detection oriented variant, which can detect adversarial examples with high predictive performance.
- We conduct extensive and comprehensive experiments to demonstrate the improvement of our proposed method over the state-of-the-art defense methods.
- We provide a further understanding of the relationship between the transferability and the overall robustness in ensemble learning context.

Our Proposed Method

In this section, we present our ensemble collaboration strategy, which allows us to collaborate many committee models for improving the ensemble robustness. We start with the definitions and some key properties of secure and insecure sets which later support us in devising promoting and demoting operations for collaborating the committee models to achieve the ensemble robustness. It is worth noting that our ensemble strategy is applicable for ensembling an arbitrary number of committee models; here we focus on presenting the key theories, principles, and operations for the canonical case of ensembling two models for better readability.

Secure and Insecure Sets

Consider a classification problem on a dataset $\mathcal{D}$ with M classes and a pair $(\mathbf{x}, \boldsymbol{y})$ that represents a data example $\mathbf{x}$ and its true label $\boldsymbol{y}$ which is sampled from the dataset $\mathcal{D}$. Given a model f, the crucial aim of defense is to make f robust by giving consistently accurate predictions over a ball, $\mathcal{B}(\mathbf{x}, \epsilon) \coloneqq \{\mathbf{x}' : \|\mathbf{x}' - \mathbf{x}\| \leq \epsilon\}$ around a benign data example $\mathbf{x}$, for every possible $\mathbf{x}$ in the dataset $\mathcal{D}$ and the distortion boundary ϵ. To further clarify and motivate our theory, we define

$$\mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f, \epsilon) \coloneqq \{\mathbf{x}' \in \mathcal{B}(\mathbf{x}, \epsilon) : \text{argmax}_i f_i(\mathbf{x}') = \boldsymbol{y}\},$$
$$\mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f, \epsilon) \coloneqq \{\mathbf{x}' \in \mathcal{B}(\mathbf{x}, \epsilon) : \text{argmax}_i f_i(\mathbf{x}') \neq \boldsymbol{y}\}.$$

Intuitively, we define a *secure* set $\mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f, \epsilon)$ as the set of elements in the ball $\mathcal{B}(\mathbf{x}, \epsilon)$ for which the classifier f makes the correct prediction. In addition, we define the *insecure* set $\mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f, \epsilon)$ as the set of elements in the ball $\mathcal{B}(\mathbf{x}, \epsilon)$ for which f predicts differently from the true label $\boldsymbol{y}$. By definition, the secure set is the complement of the insecure set, and $\mathcal{B}(\mathbf{x}, \varepsilon) = \mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f, \varepsilon) \bigcup \mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f, \varepsilon)$. It is clear that the aim of improving adversarial robustness is to train the classifier f in such the way that $\mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f, \epsilon)$ is either as small as possible (ideally, $\mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f, \epsilon) = \emptyset, \forall \mathbf{x} \in \mathcal{D}$) or makes an adversary hard to generate adversarial examples in it. The following simple lemma (see the proof in the supplementary material) shows the connection between those two kinds of sets and the robustness of the ensemble model and facilitates the development of our proposed method.

Lemma 1. *Let us define $f^{en}(\cdot) = \frac{1}{2}f^1(\cdot) + \frac{1}{2}f^2(\cdot)$ for two given models f^1 and f^2. If f^1 and f^2 predict an example $\mathbf{x}$ accurately, we have the following:*

i) $\mathcal{B}_{insecure}(\mathbf{x}, \boldsymbol{y}, f^{en}, \epsilon) \subset \mathcal{B}_{insecure}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon) \cup \mathcal{B}_{insecure}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon)$.

ii) $\mathcal{B}_{secure}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon) \cap \mathcal{B}_{secure}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon) \subset \mathcal{B}_{secure}(\mathbf{x}, \boldsymbol{y}, f^{en}, \epsilon)$.

Dual Collaborative Ensemble

Transferring Flow. Consider the canonical case of an ensemble consisting of two models: $f^{en}(\cdot) = \frac{1}{2}f^1(\cdot) + \frac{1}{2}f^2(\cdot)$, where f^{en} is the ensemble model and $\{f^1, f^2\}$ is the set of ensemble committee (or the committee). Based on the definitions of secure and insecure sets, an arbitrary adversarial example $\mathbf{x}_a$ must lie in one of four subsets as shown in Table 1. Let us further clarify these subsets. In the first subset $S_{11} = \mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon) \bigcap \mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon)$, the example $\mathbf{x}_a$ is predicted correctly by both models, hence also by the ensemble model f^{en} (Lemma 1 (ii)). The subsets S_{10}, S_{01} are the intersection of a secure set of one model and an insecure set of another model, hence an example of two sets is predicted correctly by one model and incorrectly by the other. Lastly, in the subset $S_{00} = \mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon) \bigcap \mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon)$, both models offer predictions other than the true label, but there is also no guarantee that their incorrect predictions are in the same class. There is still a chance that the incorrect prediction in subset S_{10}, S_{01} dominates the correct ones, which leads to the incorrect prediction on average. Therefore, the insecure region of the overall ensemble should be related to the union $S_{10} \cup S_{01} \cup S_{00}$ or the total volume (i.e., $|S_{10}| + |S_{01}| + |S_{00}|$) of the subsets S_{10}, S_{01}, S_{00}.

As the result, to obtain a robust ensemble model, we need to maintain the subset S_{00} as small as possible, which is in turn equivalent to making the insecure regions of the two models as disjoint as much as possible (i.e., concurred with Lemma 1 (i)). For the data points in either S_{10} or S_{01}, we need to increase the chance that the correct predictions dominate the incorrect ones. Our approach is to encourage adversarial examples inside S_{00} to move to the subsets S_{10}, S_{01} during the course of training, and those of S_{10}, S_{01} to move to the subset S_{11}. We term this movement as the *transferring flow*, which is described in Table 1. In what follows, we present how to implement the transferring flow for our ensemble model.

Promoting Adversarial Robustness (PO). We refer to promoting adversarial robustness as an operation to leverage the information of an example $\mathbf{x}_a^i$ (adversarial example of model f^i) for improving the robustness of a model f^j

Table 1: Four subsets of the ensemble model and the transferring flow (arrows)

	$\mathbf{x}_a \in \mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon)$		$\mathbf{x}_a \in \mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon)$
$\mathbf{x}_a \in \mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon)$	S_{11}	$\Leftarrow$	S_{01}
	$\Uparrow$		$\Uparrow$
$\mathbf{x}_a \in \mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon)$	S_{10}	$\Leftarrow$	S_{00}

$(i, j$ can be different). There are several adversarial defense methods that can be applied to promote adversarial robustness, notably (Madry et al., 2018; Hongyang Zhang et al., 2019; Qin et al., 2019). In this work, to promote the adversarial robustness of a given adversarial example $\mathbf{x}_a^i$ w.r.t the model f^j, we use adversarial training (Madry et al., 2018) by minimizing the cross-entropy loss w.r.t the true label as $\min\ \mathcal{C}\left(f^j(\mathbf{x}_a^i), \boldsymbol{y}\right)$. After undertaking this PO, $\mathbf{x}_a^i$ is expected to move to the secure set $\mathcal{B}_{\text{secure}}\left(\mathbf{x}, \boldsymbol{y}, f^j, \epsilon\right)$. We introduce two types of PO: direct PO (dPO) when $i = j$ and crossing PO (cPO) when $i \neq j$.

Demoting Adversarial Robustness (DO). In contrast to promoting adversarial robustness, we refer to demoting adversarial robustness as an operation to sacrifice the robustness of a model for an example $\mathbf{x}_a^i$ (adversarial example of model f^i). Here, we demote the adversarial robustness of a given adversarial example $\mathbf{x}_a^i$ w.r.t the model f^j by $\max\ \mathcal{H}\left(f^j(\mathbf{x}_a^i)\right)$ where $\mathcal{H}$ is the entropy. Without any further knowledge, the prediction is likely uniformly distributed, hence the example $\mathbf{x}_a^i$ likely falls into the insecure set $\mathcal{B}_{\text{insecure}}\left(\mathbf{x}, \boldsymbol{y}, f^j, \epsilon\right)$ instead of the secure set $\mathcal{B}_{\text{secure}}\left(\mathbf{x}, \boldsymbol{y}, f^j, \epsilon\right)$.

Collaboration of the Promoting and Demoting Operations. We now present how to coordinate PO/DO to enforce the transferring flow for enhancing the adversarial robustness of the ensemble model in the canonical case of a committee of two members $\{f^1, f^2\}$, parameterized by θ_1 and θ_2. Let $\mathbf{x}_a^1$ and $\mathbf{x}_a^2$ be white-box adversarial examples of f^1 and f^2 respectively. With a strong adversary, we can assume that $\mathbf{x}_a^1 \in \mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon)$ (i.e., $\mathbf{x}_a^1 \in S_{01} \cup S_{00}$) and $\mathbf{x}_a^2 \in \mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon)$ (i.e., $\mathbf{x}_a^2 \in S_{10} \cup S_{00}$). For ease of comprehensibility, we present the treatment for $\mathbf{x}_a^1$ and the same treatment is applied to $\mathbf{x}_a^2$. To strengthen model f^1, we always use $\mathbf{x}_a^1$ to promote the robustness of model f^1 by minimizing the cross-entropy loss $\mathcal{C}\left(f^1(\mathbf{x}_a^1), \boldsymbol{y}\right)$ (i.e., flow $S_{01} \Rightarrow S_{11}$ or $S_{00} \Rightarrow S_{10}$). Meanwhile, we consider two cases of $\mathbf{x}_a^1$ w.r.t model f^2: i) being correctly predicted by f^2 (i.e., $\mathbf{x}_a^1 \in S_{01}$) and ii) being incorrectly predicted by f^2 (i.e., $\mathbf{x}_a^1 \in S_{00}$). For the first case, we use $\mathbf{x}_a^1$ to promote model f^2 to make sure $\mathbf{x}_a^1$ stays in the secure set of model f^2 (i.e., $S_{11} \cup S_{01}$). For the second case, we demote $\mathbf{x}_a^1$ w.r.t f^2 by maximizing the entropy $\mathcal{H}\left(f^2(\mathbf{x}_a^1)\right)$ in order to keep $\mathbf{x}_a^1$ in the insecure set of model f^2 (i.e., $S_{10} \cup S_{00}$).

Therefore, with the collaboration of two models f^1 and f^2 on the same example $\mathbf{x}_a^1$, we deploy either flow $S_{01} \Rightarrow S_{11}$ or $S_{00} \Rightarrow S_{10}$ depending on the scenario of $\mathbf{x}_a^1$. It is worth noting that DO encourages $f^2(\mathbf{x}_a^1)$ to be close to the uniform prediction, hence causing a minimal effect on the ensemble prediction $f^{en}\left(\mathbf{x}_a^1\right)$. As a consequence, $f^{en}\left(\mathbf{x}_a^1\right) = \frac{1}{2}\left(f^1\left(\mathbf{x}_a^1\right) + f^2\left(\mathbf{x}_a^1\right)\right)$ is dominated by $f^1\left(\mathbf{x}_a^1\right)$, which likely offers a correct prediction via the corresponding PO: $\min\ \mathcal{C}\left(f^1(\mathbf{x}_a^1), \boldsymbol{y}\right)$. We summarize the PO/DO to deploy the transferring flow in Table 2.

Table 2: Promoting and demoting operations for the transferring flow

Scenario	f^1	f^2
$\mathbf{x}_a^1 \in S_{01}$	$\min \mathcal{C}\left(f^1(\mathbf{x}_a^1), \boldsymbol{y}\right)$	$\min \mathcal{C}\left(f^2(\mathbf{x}_a^1), \boldsymbol{y}\right)$
$\mathbf{x}_a^1 \in S_{00}$	$\min \mathcal{C}\left(f^1(\mathbf{x}_a^1), \boldsymbol{y}\right)$	$\max \mathcal{H}\left(f^2(\mathbf{x}_a^1)\right)$
$\mathbf{x}_a^2 \in S_{10}$	$\min \mathcal{C}\left(f^1(\mathbf{x}_a^2), \boldsymbol{y}\right)$	$\min \mathcal{C}\left(f^2(\mathbf{x}_a^2), \boldsymbol{y}\right)$
$\mathbf{x}_a^2 \in S_{00}$	$\max \mathcal{H}\left(f^1(\mathbf{x}_a^2)\right)$	$\min \mathcal{C}\left(f^2(\mathbf{x}_a^2), \boldsymbol{y}\right)$

The objective functions for model f^1 and f^2 to deploy the transferring flow are:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \boldsymbol{y}, \theta_1) &= \mathcal{C}\left(f^1(\mathbf{x}), \boldsymbol{y}\right) + \mathcal{C}\left(f^1(\mathbf{x}_a^1), \boldsymbol{y}\right) \\ &+ \lambda_{pm}\mathbb{I}\left(f^1(\mathbf{x}_a^2), \boldsymbol{y}\right)\mathcal{C}(f^1(\mathbf{x}_a^2), \boldsymbol{y}) \\ &- \lambda_{dm}\left(1 - \mathbb{I}\left(f^1(\mathbf{x}_a^2), \boldsymbol{y}\right)\right)\mathcal{H}\left(f^1(\mathbf{x}_a^2)\right), \quad (1) \\ \mathcal{L}(\mathbf{x}, \boldsymbol{y}, \theta_2) &= \mathcal{C}\left(f^2(\mathbf{x}), \boldsymbol{y}\right) + \mathcal{C}\left(f^2(\mathbf{x}_a^2), \boldsymbol{y}\right) \\ &+ \lambda_{pm}\mathbb{I}\left(f^2(\mathbf{x}_a^1), \boldsymbol{y}\right)\mathcal{C}(f^2(\mathbf{x}_a^1), \boldsymbol{y}) \\ &- \lambda_{dm}\left(1 - \mathbb{I}\left(f^2(\mathbf{x}_a^1), \boldsymbol{y}\right)\right)\mathcal{H}\left(f^2(\mathbf{x}_a^1)\right). \quad (2)\end{aligned}$$

where λ_{pm} and λ_{dm} are the hyper-parameters for promoting and demoting effects, respectively, and $\mathbb{I}\left(f^1(\mathbf{x}_a^2), \boldsymbol{y}\right)$ is the indicator to indicate whether $\mathbf{x}_a^2$ is predicted correctly (i.e., $\mathbb{I} = 1$, hence $\mathbf{x}_a^2 \in S_{10}$) or incorrectly (i.e., $\mathbb{I} = 0$, hence $\mathbf{x}_a^2 \in S_{00}$) by f^1, which helps to switch on/off the cPO/DO for model f^1.

For the final objective function, we approximate the hard indicator $\mathbb{I}\left(f^1(\mathbf{x}_a^2), \boldsymbol{y}\right)$ by the soft version $f^1_{\boldsymbol{y}}(\mathbf{x}_a^2) = p\left(\boldsymbol{y} \mid \mathbf{x}_a^2, f^1\right)$, which represents the probability the model f^1 assigning $\mathbf{x}_a^2$ to the label $\boldsymbol{y}$. We hence arrive at the following objective functions for both f^1 and f^2, respectively.

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \boldsymbol{y}, \theta_1) &= \mathcal{C}\left(f^1(\mathbf{x}), \boldsymbol{y}\right) + \mathcal{C}\left(f^1(\mathbf{x}_a^1), \boldsymbol{y}\right) \\ &+ \lambda_{pm} f^1_y(\mathbf{x}_a^2)\mathcal{C}(f^1(\mathbf{x}_a^2), \boldsymbol{y}) \\ &- \lambda_{dm}\left(1 - f^1_y(\mathbf{x}_a^2)\right)\mathcal{H}\left(f^1(\mathbf{x}_a^2)\right), \quad (3) \\ \mathcal{L}(\mathbf{x}, \boldsymbol{y}, \theta_2) &= \mathcal{C}\left(f^2(\mathbf{x}), \boldsymbol{y}\right) + \mathcal{C}\left(f^2(\mathbf{x}_a^2), \boldsymbol{y}\right) \\ &+ \lambda_{pm} f^2_y(\mathbf{x}_a^1)\mathcal{C}(f^2(\mathbf{x}_a^1), \boldsymbol{y}) \\ &- \lambda_{dm}\left(1 - f^2_y(\mathbf{x}_a^1)\right)\mathcal{H}\left(f^2(\mathbf{x}_a^1)\right). \quad (4)\end{aligned}$$

We note that in our implementation, the soft indicators $f^1_y(\mathbf{x}_a^2)$ and $f^2_y(\mathbf{x}_a^1)$ are used as values by performing a stopping gradient to prevent the back-propagation process to go inside them for further updating f^1 and f^2.

Crossing Collaborative Ensemble

We now extend our collaboration strategy to enable us to ensemble many individual members, which we term as a *Crossing Collaborative Ensemble (CCE)*. Specifically, given an ensemble of N members $f^{en}(\cdot) = \frac{1}{N}\sum_{n=1}^{N} f^n(\cdot)$ parameterized by θ_n, the loss function for a model $f^n, n \in [1, N]$ as follow:

$$\mathcal{L}^n(\mathbf{x}, \boldsymbol{y}, \theta_n) = \mathcal{C}\left(f^n(\mathbf{x}), \boldsymbol{y}\right) + \mathcal{C}\left(f^n(\mathbf{x}_a^n), \boldsymbol{y}\right) + \frac{1}{N-1}\sum_{i \neq n}\Bigg(\lambda_{pm} f_y^n(\mathbf{x}_a^i)\mathcal{C}(f^n(\mathbf{x}_a^i), \boldsymbol{y}) - \lambda_{dm}\left(1 - f_y^n(\mathbf{x}_a^i)\right)\mathcal{H}\left(f^n(\mathbf{x}_a^i)\right)\Bigg). \quad (5)$$

It appears from the above loss that we encourage each individual model to (i) minimize the loss of the adversarial example itself for improving its robustness (dPO) and (ii) promoting or demoting its robustness (cPO/DO) with other adversarial examples depending on the soft indicator.

Connections to Traditional Ensemble Learning. Firstly, in our method, N members $\{f^n\}$ are reinforced with the joint of $N+1$ data sources: clean data $\{\mathbf{x}\}$ and N adversarial examples $\{\mathbf{x}_a^n\}_{n=1}^N$. However, depending on different scenarios, they have the same task (PO-PO) or opposite tasks (PO-DO) on the same adversarial set $\{\mathbf{x}_a^n\}$. Our approach can be linked to the bagging technique in the literature, in which each classifier was trained on different sets of data. Secondly, by assigning opposite tasks for ensemble members, our method produces a negative correlation which was described in (Yong Liu and Yao, 1999; Kuncheva and Whitaker, 2003; Bagnall, Bunescu, and Stewart, 2017). It has been claimed that negative relationship among ensemble members can further improve the ensemble accuracy better than the independent correlation.

Experiments

In this section, we first introduce the experimental setting for adversarial defenses and attackers followed by an extensive evaluation to compare our method with state-of-the-art adversarial defenses. We show that our method surpasses these methods for common benchmark datasets. Next, we provide an ablation study to understand the transferability among ensemble members of adversarial examples. Finally, we show that our method not only detects adversarial examples accurately and consistently but also predicts benign examples with a significant improvement.

Experimental Setting

General Setting. We use CIFAR10 and CIFAR100 as the benchmark datasets in our experiment.[1] Both datasets have 50,000 training images and 10,000 test images. The inputs were normalized to $[0, 1]$. We apply random horizontal flips and random shifts with scale 10% for data augmentation as used in (Pang, Xu, et al., 2019). We use both standard CNN architecture and ResNet architecture (K. He et al., 2016) in our experiment. The architecture and training setting for each dataset are provided in our supplementary material.

Crafting Adversarial Examples for Defenders. In our experiments, we use PGD $\{k, \epsilon, \eta, l_\infty\}$ as the common adversary to generate adversarial examples for the adversarial training of all defenders where k is the iteration steps, ϵ is the distortion bound and η is the step size. Specifically, the configuration for the CIFAR10 dataset is $k = 10, \epsilon = 8/255, \eta = 2/255$ and that for the CIFAR100 dataset is $k = 10, \epsilon = 0.01, \eta = 0.001$. For the CIFAR10 dataset with ResNet architecture, we use the same setting in (Pang, Xu, et al., 2019) which is $k = 10, \epsilon \sim U(0.01, 0.05), \eta = \epsilon/10$.

Baseline Methods. Because the model capacity has significant impact on the inference performance, therefore, for a fair comparison, we compare our method with the start-of-the-art ensemble-based method, i.e., ADV-EN (Madry et al., 2018) and ADP (Pang, Xu, et al., 2019), which have the same number of committee members and also the member's architecture. More specifically, ADV-EN is the variant of PGD adversarial training method (ADV) in the context of ensemble learning, in which the entire ensemble model is treated as one unified model applied with adversarial training. We also compare with the ADV method which is adversarial training on a single model. For ADP, we choose the best setting $ADP_{2,0.5}$ with adversarial version, which was reported in the paper (Pang, Xu, et al., 2019), and use the official code.[2]

Throughout our experiments, we use two variants of our method: (i) Robustness Mode (i.e., CCE-RM) for which we set $\lambda_{pm} = \lambda_{dm} = 1$ and (ii) Detection Mode (i.e., CCE-DM) for which we disable cPO ($\lambda_{pm} = 0$) and strengthen DO (i.e., $\lambda_{dm} = 5$).

Attack Setting. We use different state-of-the-art attacks to evaluate the defense methods including:

(i) **Gradient based attacks** (with *cleverhans*[3] lib). We use PGD (Madry et al., 2018), the Basic Iterative Method (BIM) (Kurakin, I. J. Goodfellow, and S. Bengio, 2017) and the Momentum Iterative Method (MIM) (Dong et al., 2018). They share the same hyper-parameters configuration, i.e., $\{k, \epsilon, \eta\}$, which is described in each individual experiment.

(ii) **B&B attack** (Brendel et al., 2019) (with *foolbox*[4] lib) which is a decision based attack. We argue that the B&B attack setting in the paper of (Tramèr and Boneh, 2019) may not be appropriate to evaluate the ADP method. It is because the ADP method used PGD ($\epsilon \sim U(0.01, 0.05), k = 10$) for its adversarial training, while B&B attack used PGD ($\epsilon = 0.15, k = 20$) as an initialized attack which is much stronger than the defense capacity. More specifically, the initialized PGD attack alone can reduce the accuracy to 0.1%. Therefore, B&B attack contributes very little to the final attack performance. To have a fair evaluation, we use two initialized attacks with lower strength: PGD1 ($\epsilon = 8/255, \eta = 2/255, k = 20$) and PGD2 ($\epsilon = 16/255, \eta = 2/255, k = 20$) then apply B&B attack with 100 steps and repeat for three times. It is worth noting that, PGD2 is still much stronger than the defense capacity, however, we use this set-

[1] Recently, (Tsipras et al., 2020) found the labeling issue in the ImageNet dataset, which highly affects the fairness of robustness evaluation on this dataset.

[2] https://github.com/P2333/Adaptive-Diversity-Promoting

[3] https://github.com/tensorflow/cleverhans

[4] https://foolbox.readthedocs.io/en/stable/

Table 3: Robustness evaluation on the CIFAR10 dataset with ResNet architecture. For the PGD attack, we use $\epsilon = 8/255, \eta = 2/255$. (*) The low robust accuracies (even with standard method ADV) because the attack strength of PGD2 is double of the defense capacity, which makes the adversarial examples to be recognizable.

Attack	ADV_1	$ADV\text{-}EN_2$	ADP_2	$CCE\text{-}RM_2$	$ADV\text{-}EN_3$	ADP_3	$CCE\text{-}RM_3$
Non-att (Nat. acc.)	83.9	85.3	**85.3**	84.5	86.1	**86.2**	84.9
PGD $k = 250$	41.4	42.8	44.2	**45.8**	43.8	45.1	**48.6**
BIM $k = 250$	41.5	42.9	44.1	**45.8**	44.0	45.2	**48.8**
MIM $k = 250$	41.9	43.3	44.8	**46.3**	44.5	45.7	**49.1**
B&B (wPGD1)	37.0	38.3	37.3	**42.2**	39.3	38.3	**44.2**
B&B (wPGD2)*	4.9	2.9	3.9	**6.0**	4.2	4.3	**7.1**
SPSA	50.0	53.5	52.8	**56.2**	53.8	53.9	**56.6**
Auto-Attack	16.1	18.5	17.3	**18.8**	18.4	17.6	**20.8**

ting to mimic the evaluation in the paper of (Tramèr and Boneh, 2019).

(iii) **Auto-Attack** (Croce and Hein, 2020) (with the official implementation[5]) which is an ensemble based attack. We use $\epsilon = 8/255$ for the CIFAR10 dataset and $\epsilon = 0.01$ for the CIFAR100 dataset, both with standard version which is an ensemble of four different attacks.

(iv) **SPSA attack** (Uesato et al., 2018) (with *cleverhans* lib) which is a gradient-free optimization method. We use $\epsilon = 8/255$ for the CIFAR10 dataset and $\epsilon = 0.01$ for the CIFAR100 dataset, both with 50 steps**.**

The distortion metric we use in our experiments is l_∞ for all measures. We use the full test set for the attacks (i) and 1000 test samples for the attacks (ii-iv).

Robustness Evaluation

We conduct extensive experiments on the CIFAR10 and CIFAR100 datasets to compare our method with the other methods. We consider the ensemble of both two and three committee members (denoted by a subscript number in each method). It can be observed from the experimental results in Table [3, 4, 5] that:

(i) There is a gap of 2%$\sim$3% when comparing $ADV\text{-}EN_3$ with ADV_1 showing that increasing model capacity (by increasing number of ensemble member) can improve the robustness of the model.

(ii) There is a gap of 3%$\sim$4% between ADP_3 and ADV_1, and especially, a gap of 7%$\sim$8% when comparing our $CCE\text{-}RM_3$ with ADV_1, which shows the potential of the ensemble learning to tackle with the adversarial attacks.

(iii) With the same model capacity, our CCE-RM is consistently the best with all attacks and in some attacks, ours surpasses other baselines in a large margin (4%$\sim$5%).

(iv) There is a gap of 3% between $CCE\text{-}RM_3$ and $CCE\text{-}RM_2$, which is larger than the gap of 1% between ADP_3 and ADP_2 or that of $ADV\text{-}EN_3$ and $ADV\text{-}EN_2$, showing that our method collaborates members better and gets more benefit from ensembling more committee members.

The effectiveness of adversarial training method depends on the diversity (or the hardness) of the adversarial examples (Madry et al., 2018). Fort et al. (2019) found that differently initializing members' parameters, even with the same training data, can end up with different local optimal in the solution space. Therefore, the potential of ensemble learning (in the remark ii) can be explained by the fact that the adversarial space of an ensemble model $\mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f^{en}, \epsilon)$ is more diverse than that of a single model $\mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f, \epsilon)$.

[5]https://github.com/fra31/auto-attack

Table 4: Robustness evaluation on the CIFAR10 dataset with standard CNN architecture. We use $\epsilon = 8/255, \eta = 2/255$. Note that $mul\mathcal{A}$ represents for multiple-targeted attack by adversary $\mathcal{A}$.

Attack	ADV_1	$ADV\text{-}EN_2$	ADP_2	$CCE\text{-}RM_2$	$ADV\text{-}EN_3$	ADP_3	$CCE\text{-}RM_3$
Non-att (Nat. acc.)	75.7	76.0	75.9	**76.0**	**76.7**	76.6	75.7
PGD $k = 100$	38.0	39.7	42.2	**44.7**	40.8	43.9	**46.8**
BIM $k = 100$	38.2	39.7	42.2	**44.9**	40.8	43.8	**46.8**
MIM $k = 100$	38.5	40.5	42.4	**45.4**	41.3	44.2	**47.2**
mul-PGD $k = 20$	26.0	27.7	27.8	**31.9**	28.3	32.4	**36.9**
mul-BIM $k = 20$	25.9	27.2	27.2	**31.6**	27.7	29.8	**34.1**
mul-MIM $k = 20$	26.2	28.1	28.3	**32.3**	29.0	30.7	**34.6**
SPSA	40.6	44.3	41.5	**45.2**	45.1	46.1	**47.5**
Auto-Attack	25.1	25.0	24.4	**29.9**	25.5	28.1	**31.9**

Table 5: Robustness evaluation on the CIFAR100 dataset with standard CNN architecture. We use $\epsilon = 0.01, \eta = 0.001$. Note that $mul\mathcal{A}$ represents for multiple-targeted attack by adversary $\mathcal{A}$.

Attack	ADV_1	$ADV\text{-}EN_2$	ADP_2	$CCE\text{-}RM_2$	$ADV\text{-}EN_3$	ADP_3	$CCE\text{-}RM_3$
Non-att (Nat. acc.)	40.8	41.4	48.0	**53.4**	40.8	52.6	**54.4**
PGD $k = 100$	26.8	29.7	30.9	**35.3**	32.8	36.2	**39.5**
BIM $k = 100$	26.9	29.1	31.0	**35.2**	32.8	36.2	**39.4**
MIM $k = 100$	27.0	29.0	30.8	**35.3**	32.9	36.1	**39.6**
mul-PGD $k = 20$	16.4	15.8	20.1	**24.2**	16.6	24.8	**28.4**
mul-BIM $k = 20$	15.9	15.5	19.4	**23.7**	16.3	24.5	**28.1**
mul-MIM $k = 20$	16.7	16.1	20.3	**24.1**	16.8	25.1	**28.6**
SPSA	25.6	25.5	24.1	**31.8**	26.0	32.5	**35.0**
Auto-Attack	15.3	15.1	14.8	**21.9**	15.8	23.0	**25.9**

Our advantages over others (in the remark iii, iv) can be explained by the fact that our proposed method encourages the diversity of its committee members. Specifically it can be elaborated on with the following three key points. Firstly, while other ensemble-based defenses use the adversarial examples of the entire ensemble $\mathbf{x}_a^{en} \sim \mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f^{en}, \epsilon)$, our method makes use of the broader joint adversarial space $\mathbf{x}_a^i \sim \mathcal{B}_{\text{insecure}}(\mathbf{x}, \boldsymbol{y}, f^i, \epsilon)$ (Lemma 1 (i)). Secondly, each member has different loss landscape (Fort, Hu, and Lakshminarayanan, 2019), in addition with the randomness of an adversary (e.g., random starting points in PGD), each member has its individual adversarial set (partly collapsed as shown in the next experiment). Therefore, similar with the bagging technique, by promoting each member with its adversarial examples independently, we can increase the diversity of the joint adversarial space. Last but not least, inspired from traditional ensemble learning (Yong Liu and Yao, 1999), by elegantly collaborating PO and DO, we encourage the negative correlation among ensemble members, therefore, further improve the diversity of the joint adversarial space.

Transferability among Ensemble Members

The transferability is a phenomenon when adversarial examples generated to attack a specific model also mislead other models trained for the same task. In the ensemble learning context, adversarial examples which are transferred well among members will likely fool the entire ensemble. There-

fore, reducing the transferability among members is a principled approach to achieve better robustness as claimed in the previous works (Pang, Xu, et al., 2019; Kariyappa and Qureshi, 2019). In this sub-section, we provide a further understanding of the transferability to the overall robustness and show the impact of the transferring flow.

We first summarize the experiments setting. The experiments are conducted on the CIFAR10 dataset with an ensemble of two members under PGD attack with $k = 20, \epsilon = 8/255, \eta = 2/255$. The results are reported in Table 6. CCE-Base is our model which disables the crossing PO and DO by setting $\lambda_{pm} = \lambda_{dm} = 0$. $a^{(i,j)}$ represents for the robust accuracy when adversarial examples $\{x_a^i\}$ attack model f^j. $|S|$ shows the cardinality of a subset S, i.e., the percentage of the images that go into the subset S, which can be one of $\{S_{11}, S_{01}, S_{10}, S_{00}\}$. From the definition of the transferability as mentioned above, to measure the transferability of adversarial examples $\{\mathbf{x}_a^i\}$, we can compute the accuracy difference of model f^i and $f^j, j \neq i$ against the same attack $\{\mathbf{x}_a^i\}$. The smaller gap implies that adversarial examples $\{\mathbf{x}_a^i\}$ are more transferable. The overall transferability of an ensemble method can be evaluated by the sum the accuracy differences over all its members, i.e., $T = a^{(1,2)} - a^{(1,1)} + a^{(2,1)} - a^{(2,2)}$.

We would like to emphasize some following important empirical observations (Table 6):

1) **The impact of the transferring flow.** It can be observed that the cardinality $|S_{11}|$ in CCE-RM (39.9%) is larger than that in CCE-Base (36.1%), while the cardinality $|S_{01}|, |S_{10}|, |S_{00}|$ is smaller than those in CCE-Base which serves as evidence that the adversarial examples are successfully transferred from subsets S_{10}, S_{01}, S_{00} to subset S_{11} as we expect. This helps improve the overall robustness of the ensemble model from 43.3% for CCE-Base to 45.5% for CCE-RM.

2) **The transferable space is just a subset of the adversarial space.** By definition, the subset S_{00} consists of adversarial examples which fools both models f^1, f^2, therefore, S_{00} represents for the transferable space of the ensemble model f^{en}. In fact, the cardinality of $|S_{00}|$ is smaller than the insecure region of the ensemble model f^{en} (i.e., the total classification error $100\% - a^{(en,en)}$) in all methods showing that the transferable space cannot represent for the insecure region of the ensemble model f^{en}, and the former is just the subset of the latter.

3) **Reducing transferability among ensemble members is not enough to improve adversarial robustness.** In fact, the transferability metric T for CCE-RM is 33.7% which is much smaller than those for ADP and ADV-EN (59.3% and 65.5%, respectively). The smaller value of T shows that the adversarial examples $\{\mathbf{x}_a^1\}, \{\mathbf{x}_a^2\}$ in our method are more transferable than those in ADV-EN and ADP. However, the fact that the overall robustness of our method is significantly better evidently shows that *transferability is not the only factor for improving the robustness.* This is because the robustness of each individual member under a direct attack (i.e., $a^{(1,1)}$ or $a^{(2,2)}$) is much lower than our method. In addition, the cardinality $|S_{11}|$ in our method is 39.9% which is much bigger than those in ADV-EN (24.0%) and ADP (25.7%).

Table 6: Evaluation on the transferability among ensemble members on the CIFAR10 dataset. $\{T, nT, a_{single}\}$ are the metrics of interest.

Model	$a^{(en,en)}$	$a^{(1,1)}$	$a^{(2,2)}$	$\lvert S_{11}\rvert$	$\lvert S_{01}\rvert$	$\lvert S_{10}\rvert$	$\lvert S_{00}\rvert$	T	nT	a_{single}
ADV-EN	40.7	31.1	33.2	24.0	17.0	13.0	46.0	65.5	13.3	16.7
ADP	42.9	31.0	33.1	25.7	13.1	11.7	49.5	59.3	7.6	17.2
CCE-RM	45.5	41.7	41.4	39.9	5.2	5.5	49.5	33.7	5.0	5.6
CCE-Base	43.3	40.3	40.5	36.1	6.5	7.2	50.3	36.1	6.4	7.2

We provide two additional metrics which are (i) $nT = 100\% - a^{(en,en)} - |S_{00}|$ to measure the cardinality of *adversarial examples set which successful attack model f^{en} but non transferable among f^1, f^2* and (ii) $a_{single} = a^{(en,en)} - |S_{11}|$ to measure the cardinality of *adversarial examples set which are correctly predicted by only one model either f^1 or f^2 but still being correctly predicted by model f^{en}*. The comparison on the metric nT in Table 6 shows that most of successful adversarial examples in our method are predicted incorrectly by both members. While the comparison on the metric a_{single} shows that most of unsuccessful adversarial examples in our method are predicted correctly by both members. The two comparisons demonstrate that our method have better robustness than other methods because (i) the adversarial examples have to fool both ensemble members for a successful attack and (ii) our ensemble model can predict correctly by both members which explains the higher performance.

The remarks (2, 3) further imply that:

An ensemble model cannot be secure against white-box attacks unless its members are robust against direct attacks (even they are secure against transferred attacks).

This hypothesis provides more understanding of the correlation between the transferability and the overall robustness of an ensemble model.

Improving Natural Accuracy and Adversarial Detectability

The parameter $\lambda_{pm}(\lambda_{dm})$ controls the level of the agreement (disagreement) of models $\{f^i\}, i \in [1, N]$ and model $f^j, j \neq i$ on the same adversarial example x_a^j. By disabling the crossing PO ($\lambda_{pm} = 0$) and strengthening DO (i.e., $\lambda_{dm} = 5$), our method encourages the disagreement among members on the same data example, therefore, increases the negative correlation among them. This setting of CCE-DM leads to two important properties, which are empirically proved by the experiments below.

Improving Natural Accuracy. We compare natural accuracies of two variants: CCE-RM and CCE-DM against the baselines. Table 7 shows that CCE-DM significantly improves natural accuracy of the ensemble model by a large margin. In traditional ensemble learning, the key ingredient to improve natural performance is making ensemble members more diverse (Kuncheva and Whitaker, 2003). By disabling the crossing PO and strengthening DO, CCE-DM variant enforces the diversity more strictly, which explains

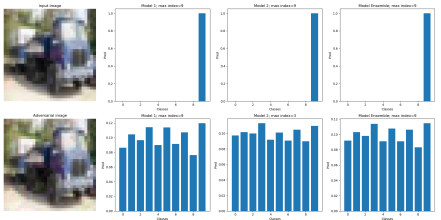


Figure 1: Prediction example in the detection mode. Top/bottom images are benign/adversarial images. Next columns are outputs from f^1, f^2, f^{en}

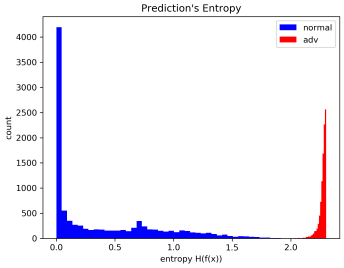


Figure 2: Histogram of prediction entropy in the detection mode

the improvement of the natural performance. This result demonstrates the promising usage of adversarial examples to improve the traditional ensemble learning.

Table 7: Comparison of the natural performance on the CIFAR10 dataset (the subscript number denotes the number members).

Model	ADV-EN	ADP	CCE-RM	CCE-DM
CNN_2	76.0	75.9	76.0	86.0
CNN_3	76.7	76.6	75.7	87.2
$ResNet_2$	85.3	85.3	84.5	91.0
$ResNet_3$	86.1	86.2	84.9	91.6

Adversarial Detectability. CCE-DM can distinguish between benign and adversarial examples more easily. It is because the committee members produce a uniform prediction for adversarial examples, while yielding a very high confident prediction for benign examples. For example, as shown in Figure 1, the committee members are highly certain when predicting benign examples, while they provide highly uncertain predictions with high entropy for adversarial examples. The histogram for all images in the test set and their adversarial examples in Figure 2 demonstrate the consistency of this observation over the data distribution.

These results further inspire us to develop a simple yet effective method to detect adversarial examples based on the entropy of the model prediction. Following the evaluation in (Pang, Du, et al., 2018; Pang, Xu, et al., 2019), we try with different thresholds to distinguish the benign and adversarial examples and report the AUC score of each adversarial attack. It is worth noting that, we do not intend to compete

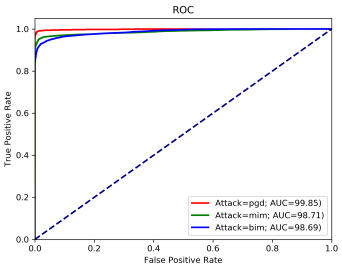


Figure 3: ROC of CCE-RM under multiple types of attack

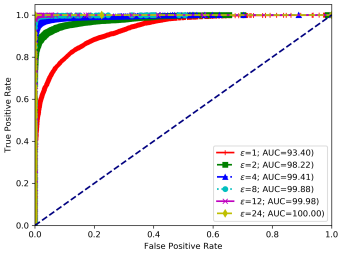


Figure 4: ROC of CCE-RM under multiple attack strengths

with other adversarial detectors but just to show the advantage and flexibility of our CCE. The experiment is on the CIFAR10 dataset with an ensemble of two members. We conduct two evaluations to justify our understanding. First, we study our detection method against three different attacks: PGD, BIM and MIM with the same hyper-parameter setting $k = 20, \epsilon = 8/255, \eta = 1/255$. The result in Figure 3 shows that our method can accurately and consistently detect all three kind of attacks. Secondly, we study our detection method on different attack strengths. We use the PGD attack $k = 20, \eta = 1/255$ and vary the distortion bound ϵ from $1/255$ to $24/255$. The result in Figure 4 shows that our method can perform well on a wide range of attack strengths. The adversary is obviously less distinguishable when decreasing its strength. However, our method still obtains a very high AUC score $(93.4/100)$ even under a very weak attack $(\epsilon = 1/255)$, in which adversarial images look nearly identical to the original ones.

Conclusion

In this paper, we explore the use of ensemble-based learning to improve adversarial robustness. In particular, we propose a cross-collaborative strategy by means of enforcing the transferring flow of adversarial examples, thereby implicitly increasing the diversity of adversarial space and improving the robustness of the ensemble. Moreover, our proposed method can be performed in both detection and robustness modes. We conduct extensive and comprehensive experiments to show the improvement of our proposed method on state-of-the-art baselines. We also provide the detailed understanding of the relationship between the transferability and the overall robustness in the ensemble learning context.

Acknowledgement. This work was partially supported by the Australian Defence Science and Technology (DST) Group under the Next Generation Technology Fund (NTGF) scheme.

References

Athalye, Anish, Nicholas Carlini, and David Wagner (2018). "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples". In: *International Conference on Machine Learning*, pp. 274–283.

Bagnall, Alexander, Razvan Bunescu, and Gordon Stewart (2017). "Training ensembles to detect adversarial examples". In: *arXiv preprint arXiv:1712.04006*.

Biggio, Battista et al. (2013). "Evasion attacks against machine learning at test time". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer, pp. 387–402.

Brendel, Wieland et al. (2019). "Accurate, reliable and fast robustness evaluation". In: *Advances in Neural Information Processing Systems*, pp. 12861–12871.

Bui, Anh et al. (2020). "Improving Adversarial Robustness by Enforcing Local and Global Compactness". In: *arXiv preprint arXiv:2007.05123*.

Carlini, N. and D. Wagner (2017). "Towards evaluating the robustness of neural networks". In: *2017 ieee symposium on security and privacy (sp)*. IEEE, pp. 39–57.

Croce, Francesco and Matthias Hein (2020). "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks". In: *arXiv preprint arXiv:2003.01690*.

Dong, Yinpeng et al. (2018). "Boosting adversarial attacks with momentum". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193.

Fort, Stanislav, Huiyi Hu, and Balaji Lakshminarayanan (2019). "Deep ensembles: A loss landscape perspective". In: *arXiv preprint arXiv:1912.02757*.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy (2015). "Explaining and Harnessing Adversarial Examples". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6572.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, Warren et al. (2017). "Adversarial example defense: Ensembles of weak defenses are not strong". In: *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*.

Kariyappa, Sanjay and Moinuddin K Qureshi (2019). "Improving adversarial robustness of ensembles with diversity training". In: *arXiv preprint arXiv:1901.09981*.

Kuncheva, Ludmila I and Christopher J Whitaker (2003). "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy". In: *Machine learning* 51.2, pp. 181–207.

Kurakin, Alexey, Ian J. Goodfellow, and Samy Bengio (2017). "Adversarial examples in the physical world". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=HJGU3Rodl.

Liu, Xuanqing et al. (2018). "Towards robust neural networks via random self-ensemble". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 369–385.

Liu, Yanpei et al. (2016). "Delving into transferable adversarial examples and black-box attacks". In: *arXiv preprint arXiv:1611.02770*.

Liu, Yong and Xin Yao (1999). "Ensemble learning via negative correlation". In: *Neural networks* 12.10, pp. 1399–1404.

Madry, Aleksander et al. (2018). "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*.

Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard (2016). "Deepfool: a simple and accurate method to fool deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582.

Pang, Tianyu, Chao Du, et al. (2018). "Towards robust detection of adversarial examples". In: *Advances in Neural Information Processing Systems*, pp. 4579–4589.

Pang, Tianyu, Kun Xu, et al. (2019). "Improving Adversarial Robustness via Promoting Ensemble Diversity". In: *International Conference on Machine Learning*, pp. 4970–4979.

Papernot, Nicolas, Patrick D. McDaniel, et al. (2016). "The Limitations of Deep Learning in Adversarial Settings". In: *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*. IEEE, pp. 372–387. DOI: 10.1109/EuroSP.2016.36. URL: https://doi.org/10.1109/EuroSP.2016.36.

Papernot, Nicolas, Patrick McDaniel, and Ian Goodfellow (2016). "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples". In: *arXiv preprint arXiv:1605.07277*.

Qin, Chongli et al. (2019). "Adversarial robustness through local linearization". In: *Advances in Neural Information Processing Systems*, pp. 13824–13833.

Shafahi, A. et al. (2019). "Adversarial training for free!" In: *Advances in Neural Information Processing Systems*, pp. 3353–3364.

Strauss, Thilo et al. (2017). "Ensemble methods as a defense to adversarial perturbations against deep neural networks". In: *arXiv preprint arXiv:1709.03423*.

Szegedy, Christian et al. (2014). "Intriguing properties of neural networks". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceed-*

ings. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1312.6199.

Tramèr, Florian and Dan Boneh (2019). "Adversarial training and robustness for multiple perturbations". In: *Advances in Neural Information Processing Systems*, pp. 5858–5868.

Tramèr, Florian, Alexey Kurakin, et al. (2018). "Ensemble adversarial training: Attacks and defenses". In: *6th International Conference on Learning Representations, ICLR 2018*.

Tramèr, Florian, Nicolas Papernot, et al. (2017). "The space of transferable adversarial examples". In: *arXiv preprint arXiv:1704.03453*.

Tsipras, Dimitris et al. (2020). "From ImageNet to Image Classification: Contextualizing Progress on Benchmarks". In: *arXiv preprint arXiv:2005.11295*.

Uesato, Jonathan et al. (2018). "Adversarial Risk and the Dangers of Evaluating Against Weak Attacks". In: *International Conference on Machine Learning*, pp. 5025–5034.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008.

Xie, Cihang et al. (2020). "Smooth adversarial training". In: *arXiv preprint arXiv:2006.14536*.

Zhang, Haichao and Jianyu Wang (2019). "Defense against adversarial attacks using feature scattering-based adversarial training". In: *Advances in Neural Information Processing Systems*, pp. 1829–1839.

Zhang, Hongyang et al. (2019). "Theoretically Principled Trade-off between Robustness and Accuracy". In: *arXiv preprint arXiv:1901.08573*.

Supplementary materials for "Improving Ensemble Robustness by Collaboratively Promoting and Demoting Adversarial Robustness"

Proof

Lemma 2. *Let us define* $f^{ens}(\cdot) = \frac{1}{2}f^1(\cdot) + \frac{1}{2}f^2(\cdot)$ *for two given models* f^1 *and* f^2. *If* f^1 *and* f^2 *predict an example* $\mathbf{x}$ *accurately, we have the following:*

i) $\mathcal{B}_{insecure}(\mathbf{x}, \boldsymbol{y}, f^{ens}, \epsilon) \subset \mathcal{B}_{insecure}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon) \cup \mathcal{B}_{insecure}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon)$.

ii) $\mathcal{B}_{secure}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon) \cap \mathcal{B}_{secure}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon) \subset \mathcal{B}_{secure}(\mathbf{x}, \boldsymbol{y}, f^{ens}, \epsilon)$.

Proof. It is obvious that Lemma 2 (i) and (ii) are equivalent. We hence need to prove only Lemma 2 (ii). Consider a classification problem on a dataset $\mathcal{D}$ with M classes, the true label of $\mathbf{x}$ is $\boldsymbol{y} \in \{1, 2, ..., M\}$ and let $\mathbf{x}' \in \mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f^1, \epsilon) \cap \mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f^2, \epsilon)$. Since f^1 and f^2 predict $\mathbf{x}'$ correctly with the label $\boldsymbol{y}$, we then have:

$$f_y^1(\mathbf{x}') \geq f_j^1(\mathbf{x}'), \forall j \in \{1, 2, ..., M\},$$
$$f_y^2(\mathbf{x}') \geq f_j^2(\mathbf{x}'), \forall j \in \{1, 2, ..., M\}.$$

This follows that

$$f_y^{ens}(\mathbf{x}') \geq f_j^{ens}(\mathbf{x}'), \forall j \in \{1, 2, ..., M\},$$

which means

$$\mathbf{x}' \in \mathcal{B}_{\text{secure}}(\mathbf{x}, \boldsymbol{y}, f^{ens}, \epsilon).$$

□

Related works

In this section we introduce the most related works to our approach including adversarial training and ensemble-based methods.

Adversarial Training.

Adversarial training (ADV) can be traced back to (I. J. Goodfellow, Shlens, and Szegedy, 2015), in which a model becomes more robust by incorporating its adversarial examples into training data. Given a model f, a benign example pair $(\mathbf{x}, \boldsymbol{y})$ and an adversarial example $\mathbf{x}_a$, the objective function of ADV as:

$$\mathcal{L}_{AT}(\mathbf{x}, \mathbf{x}_a, \boldsymbol{y}) = \mathcal{L}(f(\mathbf{x}), \boldsymbol{y}) + \mathcal{L}(f(\mathbf{x}_a), y)$$

Although many defense models were broken by (Athalye, Nicholas Carlini, and David Wagner, 2018) or gave a false sense of robustness because of the obfuscated gradient, the adversarial training (Madry et al., 2018) was among the few that were resilient against attacks. Many ADV's variants have been developed including but not limited to: (1) difference in the choice of adversarial examples, e.g., the worst-case examples (I. J. Goodfellow, Shlens, and Szegedy, 2015) or most divergent examples (Hongyang Zhang et al., 2019), (2) difference in the searching of adversarial examples, e.g., non-iterative FGSM, Rand FGSM with random initial point or PGD with multiple iterative gradient descent

steps (Madry et al., 2018; Shafahi et al., 2019), (3) difference in additional regularizations, e.g., adding constraints in the latent space (Haichao Zhang and Wang, 2019; Bui et al., 2020), (4) difference in model architecture, e.g., activation function (Xie et al., 2020) or ensemble models (Pang, Xu, et al., 2019).

Ensemble-based Defenses.

Recent works (Tramèr, Kurakin, et al., 2018; Kariyappa and Qureshi, 2019) shows that ensemble adversarial trained models can reduce the dimensionality of adversarial subspace (Tramèr, Papernot, et al., 2017). There are different approaches, however, the key ingredient of their stories is reducing the transferability of adversarial examples between members. In (Tramèr, Kurakin, et al., 2018), the authors used the crafted perturbations from static pretrained models as augmented data to decouple the generation process of adversarial examples of target model. However, as reported in (Tramèr, Kurakin, et al., 2018), this method was designed for black-box attacks, thus still vulnerable to white-box attacks. In (Kariyappa and Qureshi, 2019), robustness was achieved by aligning the gradient of committee members to be diametrically opposed, hence reducing the shared adversarial spaces, or the transferability. However, attempting to achieve gradient alignment is unreliable for high-dimensional datasets and it is difficult to extend for ensemble with more than two committee members. More recently, (Pang, Xu, et al., 2019) proposed to promote the diversity of non-maximal predictions of the committee members (i.e., the diversity among softmax probabilities except the highest ones) to reduce the adversarial transferability among them. The adaptive diversity promoting (ADP) regularizer as: $ADP_{\alpha,\beta}(\mathbf{x}, \boldsymbol{y}) = \alpha\ \mathcal{H}(\mathcal{F}) + \beta\ log(\mathbb{ED})$, where $\mathcal{H}(\mathcal{F})$ is the Shannon entropy of the ensemble prediction and $log(\mathbb{ED})$ is the logarithm of the ensemble diversity. As reported in their paper, ADP can cooperate with adversarial training to increase the robustness. In this case, the objective function of ADV as:

$$\mathcal{L}_{ADP}(\mathbf{x}, \mathbf{x}_a, \boldsymbol{y}) = \mathcal{L}_{AT}(\mathbf{x}, \mathbf{x}_a, \boldsymbol{y}) - ADP_{\alpha,\beta}(\mathbf{x}, \boldsymbol{y}) - ADP_{\alpha,\beta}(\mathbf{x}_a, \boldsymbol{y})$$

Model architecture and training setting

We use both standard CNN architecture and ResNet architecture in our experiment. For ResNet architecture, we use the same architecture and training setting as in (Pang, Xu, et al., 2019). More specifically, we use ResNet-20 and Adam optimizer, with initialized learning rate 0.001 and reduce it by a factor 0.1 at epoch 80, 120, and 160. Table 8 summarizes the standard CNN architecture for each ensemble member in our experiments. The architectures for the MNIST and CIFAR10 datasets are identical with those in (N. Carlini and D. Wagner, 2017). We use Adam optimization with learning rate 0.001 for all datasets. Conv(k) represents for the Convolutional layer with k output filters and ReLU activation. Kernel size 3 and stride 1 for every convolution layer. FC(k) represents for the Fully Connected layers with k output filters without ReLU activation. Dropout rate is 0.5. We train models in 180 epochs for both CIFAR10 and CIFAR100 datasets and in 100 epochs for the MNIST dataset.

Table 8: Model architectures for experimental section

MNIST	**CIFAR10**	**CIFAR100**
2 x Conv(32)	2 x Conv(64)	3 x Conv(64)
MaxPool	MaxPool	MaxPool
2 x Conv(64)	2 x Conv(128)	3 x Conv(128)
MaxPool	MaxPool	MaxPool
FC(200), ReLU	FC(256), ReLU	FC(256), ReLU
Dropout(0.5)	Dropout(0.5)	Dropout(0.5)
FC(200), ReLU	FC(256), ReLU	2 x (FC(256), ReLU)
FC(10)	FC(10)	FC(100)
Softmax	Softmax	Softmax

Table 9: Comparison on the training time on the CIFAR10 dataset using ResNet architecture

Model	N=2	N=3
ADV (N=1)	109s	109s
ADV-EN	205s	319s
ADP	210s	328s
Ours	356s	546s

Comparison on the training time. Our method requires to find the adversarial examples of each member and do cross inference, therefore, it takes a longer training process. We measured the training time (per epoch) on our machine with Nvidia RTX Titan GPU, using ResNet architecture (N=2,3) with batch size 64 on the CIFAR10 dataset and summarize as in Table 9.

White-box attacks evaluation

In addition to the result in the experimental section, we provide further results on the evaluation of adversarial robustness under white-box attacks. Firstly, we explain in detail the metrics of interest in our experiments. Secondly, we provide an ablation study to show the impact of the transferring flow to the improvement.

Robustness evaluation metrics

Static attack and Adaptive attack. There are two scenarios of attacks on an ensemble model (W. He et al., 2017). The first scenario is *static attack*, in which the attacker is not aware of the ensemble method (i.e., how to do the ensemble for making the final prediction). The other scenario is *adaptive attack*, where the attacker has full access to the ensemble method and adapts attacks accordingly. In our experiments, we make use of the adaptive attack, which is a considerably stronger attack.

Non-targeted attack and Multiple-targeted attack. We use both non-targeted attack ($\mathcal{A}$) and multiple-targeted attack ($mul\mathcal{A}$) in our evaluation. The non-targeted attack obtains adversarial examples by maximizing the loss w.r.t its true label, resulting in any non-true label prediction. The

Table 10: Ablation study on the impact of the transferring flow. Note that $mul\mathcal{A}$ represents for the multiple-targeted attack by adversary $\mathcal{A}$.

(a) Evaluation on CIFAR10 dataset. We commonly use $\epsilon = 8/255, \eta = 2/255$

	ADP_2	$CCE\text{-}Base_2$	$CCE\text{-}RM_2$	ADP_3	$CCE\text{-}Base_3$	$CCE\text{-}RM_3$
Non-att (Nat. acc.)	75.9	75.8	**76.0**	76.6	76.4	75.7
PGD $k = 100$	42.2	43.4	**44.7**	43.9	44.5	**46.8**
BIM $k = 100$	42.2	43.4	**44.9**	43.8	44.5	**46.8**
MIM $k = 100$	42.4	44.1	**45.4**	44.2	45.0	**47.2**
mul-PGD $k = 20$	27.8	31.1	**31.9**	32.4	32.4	**36.9**
mul-BIM $k = 20$	27.2	30.8	**31.6**	29.8	32.2	**34.1**
mul-MIM $k = 20$	28.3	31.5	**32.3**	30.7	32.9	**34.6**
SPSA	41.5	44.3	**45.2**	46.1	47.2	**47.5**
Auto-Attack	24.4	29.2	**29.9**	28.1	31.5	**31.9**

(b) Evaluation on CIFAR100 dataset. We commonly use $\epsilon = 0.01, \eta = 0.001$

	ADP_2	$CCE\text{-}Base_2$	$CCE\text{-}RM_2$	ADP_3	$CCE\text{-}Base_3$	$CCE\text{-}RM_3$
Non-att (Nat. acc.)	48.0	51.1	**53.4**	52.6	54.2	**54.4**
PGD $k = 100$	30.9	33.6	**35.3**	36.2	37.0	**39.5**
BIM $k = 100$	31.0	33.7	**35.2**	36.2	37.1	**39.4**
MIM $k = 100$	30.8	33.5	**35.3**	36.1	37.2	**39.6**
mul-PGD $k = 20$	20.1	23.0	**24.2**	24.8	26.2	**28.4**
mul-BIM $k = 20$	19.4	22.6	**23.7**	24.5	25.9	**28.1**
mul-MIM $k = 20$	20.3	23.1	**24.1**	25.1	26.4	**28.6**
SPSA	24.1	**32.1**	31.8	32.5	**35.1**	35.0
Auto-Attack	14.8	**22.0**	21.9	23.0	**26.1**	25.9

multiple-targeted attack is undertaken by performing simultaneously targeted attacks for all possible data labels (10 for CIFAR10 and 100 for CIFAR100) and being counted if any individual targeted-attack is successful. While the non-targeted attack considers only one direction of the gradient, the multiple-targeted attack takes many directions into account, therefore, being considered as a much stronger attack.

Ablation study

We provide an ablation study to compare CCE-RM with CCE-Base (which disables promoting and demoting operations by setting $\lambda_{pm} = \lambda_{dm} = 0$). Firstly, the comparison in Table 10 shows that even CCE-Base variant can beat ADP method on both CIFAR10 and CIFAR100 datasets. This surpassness can be explained from the fact our proposed method encourages the diversity of its committee members. More specifically, each member is reinforced with two data sources: clean data $\{\mathbf{x}\}$ and adversarial examples $\{\mathbf{x}_a^n\}$, which becomes more diverge due to the gradually more divergence of the committee models and the random initialization of PGD at the step 0. From this point of view, our method can be linked to the bagging technique in traditional ensemble learning, which is a well-known method to produce the diversity in the ensemble. Secondly, CCE-RM shows a huge improvement over CCE-Base in both CIFAR10 and CIFAR100 datasets. This result demonstrates the impact of the transferring flow, which offers better collaboration among members.

In addition, we study the impact of each PO and DO to the final performance by evaluating them separately. Table 11 shows the comparison when disabling one of these operations while varying the other. It can be observed that: (i) the ensemble tends to be detection mode (i.e., increasing natural performance and adversarial detectability while sacrificing its robustness) when increasing DO's strength ($\lambda_{dm} \geq 2$),

Table 11: Ablation study on the impact of each operation PO/DO. We commonly use $\epsilon = 8/255, \eta = 2/255$. Note that $mul\mathcal{A}$ represents for the multiple-targeted attack by adversary $\mathcal{A}$.

(a) Using DO only by disabling cPO ($\lambda_{pm} = 0$)

	$\lambda_{dm} = 0$	$\lambda_{dm} = 1$	$\lambda_{dm} = 2$	$\lambda_{dm} = 5$
Non-att (Nat. acc.)	75.8	76.6	83.2	86.0
PGD $k = 100$	43.4	43.3	23.9	26.1
BIM $k = 100$	43.4	43.2	24.0	26.2
MIM $k = 100$	44.1	43.6	31.1	21.7
mul-PGD $k = 20$	31.1	29.9	14.9	20.1
mul-BIM $k = 20$	30.8	29.8	13.1	19.8
mul-MIM $k = 20$	31.5	30.3	28.6	21.7
SPSA	44.3	44.4	30.4	5.3
Auto-Attack	29.2	29.8	1.6	0.2

(b) Using PO only by disabling DO ($\lambda_{dm} = 0$)

	$\lambda_{pm} = 0$	$\lambda_{pm} = 1$	$\lambda_{pm} = 2$	$\lambda_{pm} = 5$
Non-att (Nat. acc.)	75.8	76.4	76.2	77.1
PGD $k = 100$	43.4	42.7	42.7	41.3
BIM $k = 100$	43.4	42.8	42.7	41.5
MIM $k = 100$	44.1	43.0	43.2	41.9
mul-PGD $k = 20$	31.1	30.4	30.0	29.7
mul-BIM $k = 20$	30.8	30.3	29.9	29.6
mul-MIM $k = 20$	31.5	30.8	30.4	30.1
SPSA	44.3	44.3	45.5	46.2
Auto-Attack	29.2	29.7	28.8	30.0

(ii) the ensemble tends to reduce its robustness slightly when increasing PO's strength, (iii) neither PO nor DO can improve the robustness alone, which shows the important of the transferring flow. These observations are inline with the properties of PO and DO which have been mentioned in the main paper. The parameter $\lambda_{pm}(\lambda_{dm})$ controls the level of the agreement (disagreement) of models $\{f^i\}, i \in [1, N]$ and model $f^j, j \neq i$ on the same adversarial example x_a^j. Therefore the observation (i) can be explained by the fact that by disabling cPO ($\lambda_{pm} = 0$) and strengthening DO, our method encourages the disagreement among members on the same data example, therefore, increases the negative correlation among them. In contrast, by disabling DO and increasing cPO's strength, our method increases the agreement among members, therefore, increases the positive correlation. The increasing of the positive correlation among members reduces the diversity of adversarial space, therefore, explains the observation (ii).

Black-box attacks evaluation

We investigate the transferability of adversarial examples among models and evaluate the robustness under black-box attacks. The experiment is conducted on the CIFAR10 and CIFAR100 datasets, with ensemble of two members. We use PGD to challenge each ensemble model to generate adversarial examples then transfer these adversarial examples to attack other models. The PGD configuration for the CIFAR10 dataset is $k = 20, \eta = 2/255, \epsilon \in \{8/255, 12/255\}$, while that for the CIFAR100 dataset is $k = 20, \eta = 0.001, \epsilon \in \{0.01, 0.02\}$. The result is shown in Figure **??**.

Table 12: Blackbox attack evaluation on CIFAR10 dataset

(a) PGD attack with $k = 20, \epsilon = 8/255, \eta = 2/255$

	NAT	ADV-EN	ADP	CCE-RM
NAT	9.2	76.2	75.6	74.5
ADV-EN	59.5	40.7	58.5	58.4
ADP	58.4	61.8	42.9	59.9
CCE-RM	57.3	58.9	57.4	45.5

(b) PGD attack with $k = 20, \epsilon = 12/255, \eta = 2/255$

	NAT	ADV-EN	ADP	CCE-RM
NAT	8.0	75.5	75.1	74.0
ADV-EN	40.1	23.6	44.5	45.2
ADP	38.4	48.5	24.9	47.1
CCE-RM	35.9	42.9	41.3	27.8

Table 13: Blackbox attack evaluation on CIFAR100 dataset

(a) PGD attack with $k = 20, \epsilon = 0.01, \eta = 0.001$

	NAT	ADV-EN	ADP	CCE-RM
NAT	14.7	43.6	49.3	52.9
ADV-EN	42.9	31.1	48.1	51.6
ADP	41.8	41.6	32.8	50.9
CCE-RM	39.4	40.2	45.8	36.1

(b) PGD attack with $k = 20, \epsilon = 0.02, \eta = 0.001$

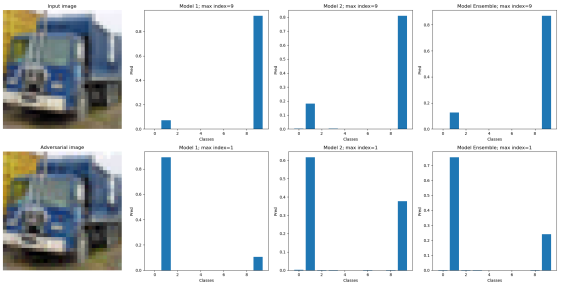
	NAT	ADV-EN	ADP	CCE-RM
NAT	13.0	43.6	49.2	52.9
ADV-EN	39.9	24.3	46.9	50.4
ADP	38.3	40.6	25.2	49.4
CCE-RM	34.5	38.2	43.0	27.5

The element $a^{(i,j)}$ in each sub-table represents the robust accuracy when adversarial examples from model i attack model j. *NAT* represents for the natural model which does not engage with any defense method. It is worth noting that, the diagonal in each sub-table represents robust accuracies against the white-box attacks, which has been discussed in the section above.

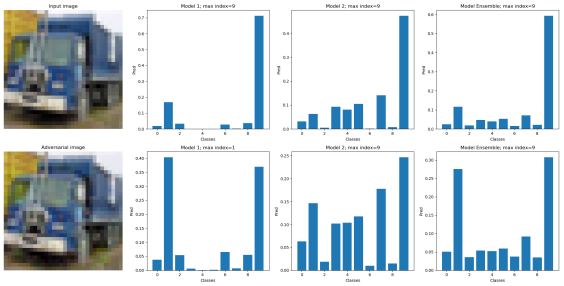
Firstly, the first row in each sub-table shows the robust accuracy against adversarial examples which are transferred from the natural model (*NAT*). This result shows that our method outperforms baseline methods on the CIFAR100 dataset, but to be weaker than other on the CIFAR10 dataset. Secondly, each column in each sub-table compares the attack strength of different models on the same defense model. The comparison on these columns shows that adversarial examples which are crafted from our CCE-RM attack better than those crafted from other methods (i.e., by giving lower robust accuracy). This result indicates that our method generates stronger adversarial examples than other methods.

Loss surface visualization

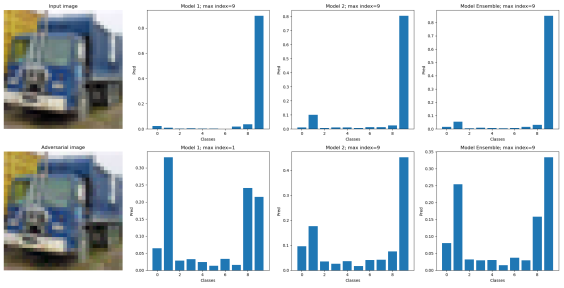
In addition to the quantitative evaluation on the adversarial robustness, we would like to provide two additional visualizations which further demonstrate our improvement. The visualizations are conducted on the same image from CIFAR10 dataset with the ensemble of two models. First, we visualize the prediction probability of each ensemble member and the entire ensemble with the same two types of input which are a benign example and adversarial example of the benign one. The visualization as Figure 5 shows that our method can produce a high confident prediction unlike ADP which has a less confident prediction because of its diversity promoting method. Secondly, we visualize the loss surface around the adversarial example $\mathbf{x}_a$ w.r.t three different of model: f^1, f^2 and f^{en}. We generate a grid of neighborhood images $\{\mathbf{x}_a + i * u + j * v\}$ where $u = \nabla_{\mathbf{x}} \mathcal{C}\left(f(\mathbf{x}_a), \boldsymbol{y}\right)$ is the gradient of the prediction loss w.r.t the input and v is the random perpendicular vector to u. In each sub-figure, the left image is the adversarial example of interest while the middle and the right image depict the loss surface and the predicted labels corresponding with the neighbor grid. Our method can produce correct labels in entire the neighborhood region, unlike other methods that still have an incorrect prediction region. Therefore, our method can produce a smoother surface around the adversarial example which further explains the better robustness in our method.



(a) ADV-EN.

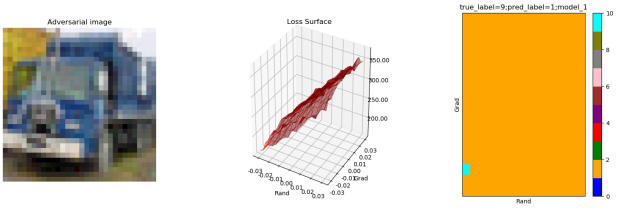


(b) ADP.

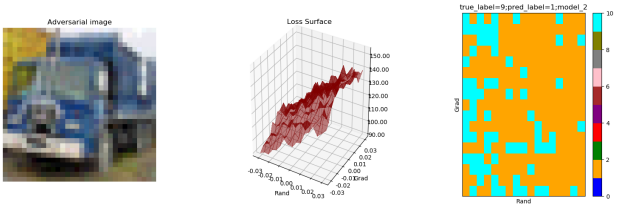


(c) CCE-RM.

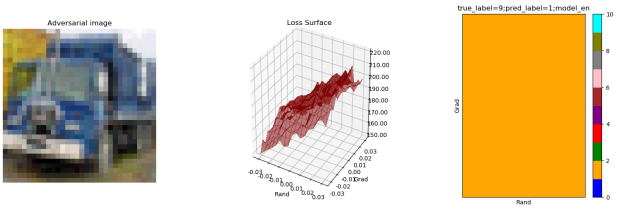
Figure 5: Prediction example. Top/bottom images are benign/adversarial images. Next columns are outputs from f^1, f^2, f^{en}.



(a) Prediction surface of model f^1.

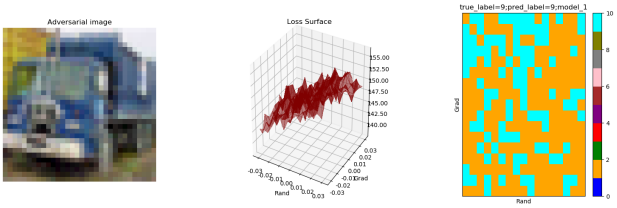


(b) Prediction surface of model f^2.

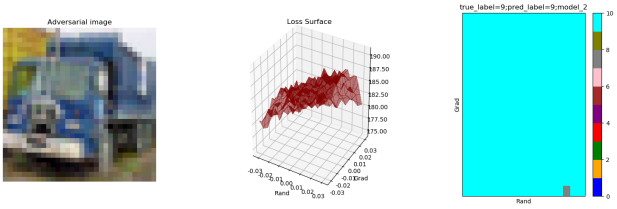


(c) Prediction surface of model f^{en}.

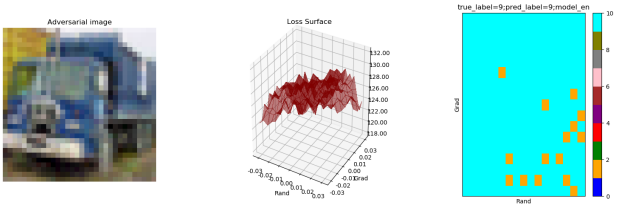
Figure 6: Loss surface around adversarial example of ADV-EN method. Left: Adversarial input. Middle: Loss surface. Right: Predicted labels.



(a) Prediction surface of model f^1.

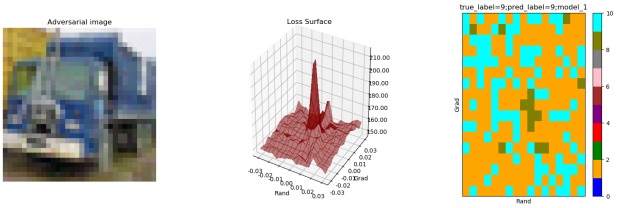


(b) Prediction surface of model f^2.

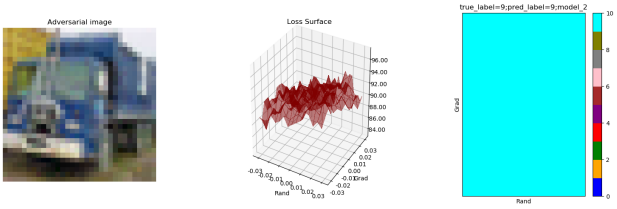


(c) Prediction surface of model f^{en}.

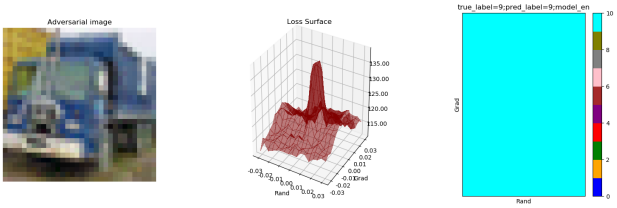
Figure 7: Loss surface around adversarial example of ADP method. Left: Adversarial input. Middle: Loss surface. Right: Predicted labels.



(a) Prediction surface of model f^1.



(b) Prediction surface of model f^2.



(c) Prediction surface of model f^{en}.

Figure 8: Loss surface around adversarial example of CCE-RM method. Left: Adversarial input. Middle: Loss surface. Right: Predicted labels.

4.4 Multi-Objective Optimization for Generating Adversarial Examples

Generating Adversarial Examples with Task Oriented Multi-Objective Optimization

Anh Bui *tuananh.bui@monash.edu*

Monash University

Trung Le *trunglm@monash.edu*
Monash University

He Zhao *he.zhao@ieee.org*
CSIRO's Data61, Australia

Quan Tran *qtran@adobe.com*
Adobe Research

Paul Montague *paul.montague@dst.defence.gov.au*
Defence Science and Technology Group, Australia

Dinh Phung *dinh.phung@monash.edu*
Monash University, VinAI Research

Reviewed on OpenReview: *https://openreview.net/forum?id=XXXX*

Abstract

Deep learning models, even the-state-of-the-art ones, are highly vulnerable to adversarial examples. Adversarial training is one of the most efficient methods to improve the model's robustness. The key factor for the success of adversarial training is the capability to generate qualified and divergent adversarial examples which satisfy some objectives/goals (e.g., finding adversarial examples that maximize the model losses for simultaneously attacking multiple models). Therefore, multi-objective optimization (MOO) is a natural tool for adversarial example generation to achieve multiple objectives/goals simultaneously. However, we observe that a naive application of MOO tends to maximize all objectives/goals equally, without caring if an objective/goal has been achieved yet. This leads to useless effort to further improve the goal-achieved tasks, while putting less focus on the goal-unachieved tasks. In this paper, we propose Task Oriented MOO to address this issue, in the context where we can explicitly define the goal achievement for a task. Our principle is to only maintain the goal-achieved tasks, while letting the optimizer spend more effort on improving the goal-unachieved tasks. We conduct comprehensive experiments for our Task Oriented MOO on various adversarial example generation schemes. The experimental results firmly demonstrate the merit of our proposed approach. Our code is available at `https://github.com/tuananhbui89/TAMOO`.

1 Introduction

Deep neural networks are powerful models that achieve impressive performance across various domains such as bioinformatics (Spencer et al., 2015), speech recognition (Hinton et al., 2012), computer vision (He et al., 2016), and natural language processing (Vaswani et al., 2017). Despite achieving state-of-the-art performance, these models are extremely fragile, as one can easily craft small and imperceptible adversarial perturbations of input data to fool them, hence resulting in high misclassifications (Szegedy et al., 2014; Goodfellow et al., 2015). Accordingly, adversarial training (AT) (Madry et al., 2018; Zhang et al., 2019) has been proven to

be one of the most efficient approaches to strengthen model robustness (Athalye et al., 2018). AT requires challenging models with divergent and qualified adversarial examples (Madry et al., 2018; Zhang et al., 2019; Bui et al., 2021b) so that the robustified models can defend against adversarial examples. Therefore, generating adversarial examples is an important research topic in Adversarial Machine Learning (AML). Several perturbation based attacks have been proposed, notably PGD (Madry et al., 2018), CW (Carlini & Wagner, 2017), and AutoAttack (Croce & Hein, 2020). Most of them aim to optimize a single objective/goal, e.g., maximizing the cross-entropy (CE) loss w.r.t. the ground-truth label (Goodfellow et al., 2015; Madry et al., 2018), maximizing the Kullback-Leibler (KL) divergence w.r.t. the predicted probabilities of a benign example (Zhang et al., 2019), or minimizing a combination of perturbation size and predicted loss to a targeted class as in Carlini & Wagner (2017).

However, in many contexts, we need to find qualified adversarial examples satisfying multiple objectives/goals, e.g., finding an adversarial example that can *attack simultaneously multiple models* in an ensemble model (Pang et al., 2019; Bui et al., 2021b), finding an universal perturbation that can *attack simultaneously multiple benign examples* (Moosavi-Dezfooli et al., 2017). Obviously, these adversarial generations have a nature of multi-objective problem rather than a single-objective one. Consequently, using *single-objective* adversarial examples leads to a much less adversarial robustness in ensemble learning as discussed in Section 4.2 and Appendix D.2.

Multi-Objective Optimization (MOO) (Désidéri, 2012) is an optimization problem to find a Pareto optimality that aims to optimize multiple objective functions. In a nutshell, MOO is a natural tool for the aforementioned multi-objective adversarial generations. However, a direct and naive application of MOO to generating robust adversarial examples for multiple models or ensemble of transformations does not work satisfactorily (cf. Appendix E). Concretely, it can be observed that the tasks are not optimized equally. The optimizing process focuses too much on one dominating task and can be trapped easily by it, hence leading to downgraded attack performances.

Intuitively, for multi-objective adversarial generations, we can explicitly investigate if an objective or a task achieves or fails to achieve its goal (e.g., the current adversarial example can fool a model successfully or unsuccessfully in multiple models). To avoid some tasks dominating others during the optimization process, we can favour more the tasks that are failing and pay less attention to the tasks that are performing well. For example, in the context of attacking multiple models, we update an adversarial example x^a to favor the models that x^a has not attacked successfully yet, while trying to maintain the attack capability of x^a on the already successful models. In this way, we expect that no task really dominates others and all tasks can be updated equally to fulfill their goals.

Bearing this in mind, we propose a new framework named **TA**sk Oriented **M**ulti-**O**bjective **O**ptimization (TA-MOO) with multi-objective adversarial generations as the demonstrating applications. Specifically, we learn a weight vector (i.e., each dimension is the weight for a task) lying on a simplex corresponding to all tasks. To favor the unsuccessful tasks while maintaining the success of the successful ones, we propose a geometry-based regularization term that represents the distance between the original simplex and a reduced simplex which involves the weight vectors for the currently unsuccessful tasks only. Furthermore, along with the original quadratic term of the standard MOO helping to improve all tasks, minimizing our geometry-based regularization term encourages the weights of the goal-achieved tasks to be as small as possible, while inspiring those for the goal-unachieved ones to have a sum close to 1. By doing so, we aim to focus more on improving the goal-unachieved tasks, while still maintain the performance of goal-achieved tasks.

Most related work to ours is Wang et al. (2021), which considers the worst-case performance across all tasks. However, this original principle reduces the generalizability to other tasks. To mitigate this issue, a specific regularization was proposed to balance all tasks' weights. Our work, which casts an adversarial generation task as a multi-objective optimization problem, is conceptually different from that work, although both methods can be applied to similar tasks. Further discussion about relate work can be found in Appendix A.

To summarize, our contributions in this work include:

(C1) We propose a novel framework called TA-MOO, which addresses the shortcomings of the original MOO when applied to multi-objective adversarial generation. Specifically, the TA-MOO framework incorporates

a geometry-based regularization term that favors unsuccessful tasks, while simultaneously maintaining the performance of successful tasks. This innovative approach improves the efficiency and efficacy of adversarial generation by promoting a more balanced exploration of the solution space.

(C2) We conduct comprehensive experiments for three adversarial generation tasks and one adversarial training task including attacking multiple models, learning universal perturbation, attacking over many data transformations, and adversarial training on ensemble learning setting. The experimental results show that our TA-MOO outperforms the baselines by a wide margin on the three aforementioned adversarial generation tasks. More importantly, our adversary brings a great benefit on improving adversarial robustness, highlighting the potential of our TA-MOO framework in adversarial machine learning.

(C3) Additionally, we provide a comprehensive analysis on different aspects of applying MOO and TA-MOO to adversarial generation tasks, such as the impact of the dominating issue in Appendix E.1, the importance of the Task-Oriented regularization in Appendix E.2, the impact of initialization of MOO in Appendix subsec:optimal-init-moo, and the limitations of MOO solver in Appendix sec:sup-gradient-des-discuss. We believe that our analysis would be beneficial for future research in this area.

2 Background

We revisit the background of multi-objective optimization (MOO), which lays the foundation for our task-oriented MOO in the sequel. Given multiple objective functions $f(\delta) := [f_1(\delta), ..., f_m(\delta)]$ where each $f_i : \mathbb{R}^d \to \mathbb{R}$, we aim to find the Pareto optimal solution that simultaneously maximizes all objective functions:

$$\max_{\delta} f(\delta) := [f_1(\delta), ..., f_m(\delta)]. \tag{1}$$

While there are a variety of MOO solvers (Miettinen, 2012; Ehrgott, 2005), in this paper, we adapt from the multi-gradient descent algorithm (MGDA) that was proposed suitably for end-to-end learning by Désidéri (2012). Specifically, MGDA combines the gradients of individual objectives to a single optimal direction that increases all objectives simultaneously. The optimal direction corresponds to the minimum-norm point that can be found by solving the quadratic programming problem:

$$w^* = \text{argmin}_{w \in \Delta_m} w^T Q w, \tag{2}$$

where $\Delta_m = \left\{\pi \in \mathbb{R}_+^m : \|\pi\|_1 = 1\right\}$ is the m-simplex and $Q \in \mathbb{R}^{m \times m}$ is the matrix with $Q_{ij} = \nabla_\delta f_i(\delta)^T \nabla_\delta f_j(\delta)$. Finally, the solution of the problem 1 can be found iteratively with each update step $\delta = \delta + \eta g$ where g is the combined gradient $g = \sum_{i=1}^m w_i^* \nabla_\delta f_i(\delta)$ and $\eta > 0$ is a sufficiently small learning rate. Furthermore, Désidéri (2012) also proved that by using an appropriate learning rate at each step, we reach the Pareto optimality point δ^* at which there exist $w \in \Delta_m$ such that $\sum_{i=1}^m w_i \nabla_\delta f_i(\delta^*) = \mathbf{0}$.

3 Our Proposed Method

3.1 Task Oriented Multi-Objective Optimization

We now present our **TA**sk Oriented **M**ulti-**O**bjective **O**ptimization (TA-MOO). We consider the MOO problem in (1) where each task $\mathcal{T}_i (i = 1, ..., m)$ corresponds to the objective function $f_i(\delta)$ $(i = 1, ..., m)$. Additionally, assume that given a task $\mathcal{T}_i$, we can explicitly observe if this task has currently achieved its goal (e.g., the current adversarial example x can fool successfully the model f_i), which is named a goal-achieved task. We also name a task that has not achieved its goal a goal-unachieved task. Different from the standard MOO, which equally pays equal attention to all tasks, our TA-MOO focuses on improving the currently goal-unachieved tasks, while trying to maintain the performance of the goal-achieved tasks. By this principle, we expect all tasks would be equally improved to simultaneously achieve their goals.

To be more precise, we depart from δ_0 and consecutively update in L steps to obtain the sequence $\delta_1, \delta_2, ..., \delta_L$ that approaches the optimal solution. Considering the t-th step (i.e., $1 \le t \le L$), we currently have δ_t and need to update it to obtain δ_{t+1}. We examine the tasks that have achieved their goals already and denote them as $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_s$ without the loss of generalization. Here we note that the list of goal-achieved tasks is

empty if $s = 0$ and the list of goal-unachieved tasks is empty if $s = m$. Specifically, to find δ_{t+1}, we first solve the following optimization problem (OP):

$$w^* = \text{argmin}_{w \in \Delta_m} \left\{ w^T Q w + \lambda \Omega(w) \right\}, \tag{3}$$

where $Q \in \mathbb{R}^{m \times m}$ with $Q_{ij} = \nabla_\delta f_i(\delta_t)^T \nabla_\delta f_j(\delta_t)$, $\lambda > 0$ is a trade-off parameter, and $\Omega(w)$ is a regularization term to let the weights focus more on the goal-unachieved tasks. We next compute the combined gradient g_t and update δ_t as:

$$g_t = \sum_{i=1}^{m} w_i^* \nabla_\delta f_i(\delta_t) \text{ and } \delta_{t+1} = \delta_t + \eta g_t.$$

The OP in (3) consists of two terms. The first term $w^T Q w$ ensures that all tasks are improving, while the second term $\Omega(w)$ serves as the regularization to restrict the goal-achieved tasks $\mathcal{T}_1, ..., \mathcal{T}_s$ by setting the corresponding weights $w_1, ..., w_s$ as small as possible.

Before getting into the details of the regularization, we emphasize that to impose the constraint $w \in \Delta_m$, we parameterize $w = \text{softmax}(\alpha)$ with $\alpha \in \mathbb{R}^m$ and solve the OP in (3) using gradient descent. In what follows, we discuss our proposed geometry-based regularization term $\Omega(w)$.

Simplex-based regularization. Let $\mathcal{S}_u = \left\{ \beta = [\beta_i]_{i=s+1}^m \in \mathbb{R}_+^{m-s} : \sum_{i=s+1}^m \beta_i = 1 \right\}$ be a simplex w.r.t. the goal-unachieved tasks and $\mathcal{S} = \{\mathbf{0}_s\} \times \mathcal{S}_u$ be the extended simplex, where $\mathbf{0}_s$ is the s-dimensional vector of all zeros. We define the regularization term $\Omega(w)$ as the distance from w to the extended simplex $\mathcal{S}$:

$$\Omega(w) = d(w, \mathcal{S}) = \min_{\pi \in \mathcal{S}} \|w - \pi\|_2^2. \tag{4}$$

Because $\mathcal{S}$ is a compact and convex set and $\|w - \pi\|_2^2$ is a differentiable and convex function, the optimization problem in (4) has a unique global minimizer $\Omega(w) = \|w - \text{proj}_{\mathcal{S}}(w)\|_2^2$, where the projection $\text{proj}_{\mathcal{S}}(w)$ is defined as

$$\text{proj}_{\mathcal{S}}(w) = \text{argmin}_{\pi \in \mathcal{S}} \|w - \pi\|_2^2.$$

The following lemma shows us how to find the projection $\text{proj}_{\mathcal{S}}(w)$ and evaluate $\Omega(w)$.

Lemma 1. *Sorting $w_{s+1:m}$ into $u_{s+1:m}$ such that $u_{s+1} \geq u_{s+2} \geq ... \geq u_m$. Defining $\rho = \max \left\{ s+1 \leq i \leq m : u_i + \frac{1}{i-s} \left(1 - \sum_{j=s+1}^{i} u_j\right) > 0 \right\}$. Denoting $\gamma = \frac{1}{\rho}\left(1 - \sum_{i=s+1}^{\rho} u_i\right)$, the projection $proj_{\mathcal{S}}(w)$ can be computed as*

$$proj_{\mathcal{S}}(w)_i = \begin{cases} 0 & 1 \leq i \leq s \\ \max\{w_i + \gamma, 0\} & otherwise \end{cases}$$

Furthermore, the regularization $\Omega(w)$ has the form:

$$\Omega(w) = \sum_{i=1}^{s} w_i^2 + \sum_{i=s+1}^{m} \left(w_i - \max\{w_i + \gamma, 0\}\right)^2. \tag{5}$$

With further algebraic manipulations, $\Omega(w)$ can be significantly simplified as shown in Theorem 1.

Theorem 1. *The regularization $\Omega(w)$ has the following closed-form:*

$$\Omega(w) = \sum_{i=1}^{s} w_i^2 + \frac{1}{m-s} \left(1 - \sum_{i=s+1}^{m} w_i\right)^2. \tag{6}$$

The proof of Lemma 1 and Theorem 1 can be found in Appendix B.1. Evidently, the regularization term in Eq. (6) in Theorem 1 encourages the weights $w_{1:s}$ associated with the goal-achieved tasks to be as small as possible and the weights $w_{s+1:m}$associated with the goal-unachieved tasks to move closer to the simplex $\mathcal{S}_u$ (i.e., $\sum_{i=s+1}^{m} w_i$ is closer to 1).

Parameterized TA-MOO. Algorithm 1 summarizes the key steps of our TA-MOO. We use gradient descent to find solution δ for the OP 1 in L steps and at each iteration we solve the OP in 3 in K steps using gradient descent solver with the parameterization $w = \text{softmax}(\alpha)$. To reduce computational cost, at each iteration we reuse the previous solution α and use a few steps K (i.e., $K \leq 10$) to get new solution. We then compute the combined gradient g_t and finally update δ_t to δ_{t+1} using the combined gradient g_t (or $\text{sign}(g_t)$ in the case of L_∞ norm). The projecting operation in step 13 is to project δ to a valid space specifying to applications that we introduce hereon.

Algorithm 1 Pseudocode for Parameterized TA-MOO.

Input: Multi-objective functions $f_{1:m}(\delta)$. δ's solver with L update steps and learning rate η_δ. w's Gradient Descent Solver (GD) with K update steps and learning rate η_w and variable α. The softmax function denotes by σ. Tradeoff parameter λ.

Output: The optimal solution δ^*.

1: Initialize δ_0 (e.g., $\delta_0 \sim \mathcal{U}(-\epsilon, \epsilon)$).
2: Initialize $\alpha_0 = [\alpha_0^i]_{i=1}^m$ with $\alpha_0^i = 1/m$.
3: **for** $t = 0$ to $L - 1$ **do**
4: Collect list of tasks' gradients $\{\nabla_\delta f_i(\delta_t)\}_{i=1}^m$.
5: Compute Q with $Q_{ij} = \nabla_\delta f_i(\delta_t)^T \nabla_\delta f_j(\delta_t)$.
6: Initialize $\alpha_{t+1} = \alpha_t$
7: **for** $k = 0$ to $K - 1$ **do**
8: Compute $\mathcal{L}(\alpha_{t+1}) = \sigma(\alpha_{t+1})^T Q \sigma(\alpha_{t+1}) + \lambda\Omega(\sigma(\alpha_{t+1}))$.
9: Update $\alpha_{t+1} = \alpha_{t+1} - \eta_w \nabla_\alpha \mathcal{L}(\alpha_{t+1})$.
10: **end for**
11: Compute the combined gradient $g_t = \sum_{i=1}^m \sigma(\alpha_{t+1,i}) \nabla_\delta f_i(\delta_t)$.
12: Update $\delta_{t+1} = \delta_t + \eta_\delta g_t$.
13: Project δ_{t+1} to a valid space (specific to domain, e.g., $\|\delta\| \leq \epsilon$).
14: **end for**
15: Output $\delta^* = \delta_L$.

3.2 Applications in Adversarial Generation

Although TA-MOO is a general framework, we in this paper focus on its applications in adversarial generation. Following Wang et al. (2021), we consider three tasks of generating adversarial examples.

Generating adversarial examples for an ensemble model. Considering an ensemble classifier with multiple classification models $h_1, h_2, ..., h_m$, where $h_i(x) \in \Delta_M = \left\{\pi \in \mathbb{R}_+^M : \|\pi\|_1 = 1\right\}$ with the number of classes M. Given a data sample x, our aim is to find an adversarial example $x^a = x + \delta$ that can successfully attack all the models. Specifically, we consider a set of tasks each of which, $\mathcal{T}_i$, is about whether $x + \delta$ can successfully attack model h_i, defined as:

$$\mathbb{I}\left\{\text{argmax}_{1 \leq k \leq M} h_i(x + \delta, k) \neq y\right\},$$

where y is the ground truth label of x, $\mathbb{I}$ is the indicator function and $h_i(x, k)$ returns the probability to predict x to the class k. To find a perturbation δ that can attack successfully all models, we solve the following multi-objective optimization problem:

$$\max_{\delta: \|\delta\| \leq \epsilon} \left[f_1(\delta), ..., f_m(\delta)\right],$$

where $f_i(\delta) = \ell(h_i(x + \delta), y)$ with the loss function ℓ which could be the cross-entropy (CE) loss (Madry et al., 2018), the Kullback-Leibler (KL) loss (Zhang et al., 2019), or the Carlini-Wagner (CW) loss (Carlini & Wagner, 2017).

Generating universal perturbations. Considering a single classification model h with $h(x) \in \Delta_M$ and a batch of data samples $x_1, x_2, ..., x_B$, we would like to find a perturbation δ with $\|\delta\| \leq \epsilon$ such that

$x_i^a = x_i + \delta, i = 1, ..., B$, are adversarial examples. We define the task $\mathcal{T}_i$ as finding the adversarial example $x_i^a = x_i + \delta$ for data sample x_i. For each task $\mathcal{T}_i$, we can define its goal as finding successfully the adversarial example x_i^a:

$$\mathbb{I}\left\{\text{argmax}_{1\leq k\leq M} h\left(x_i^a, k\right) \neq \text{argmax}_{1\leq k\leq M} h\left(x_i, k\right)\right\}.$$

To find the perturbation δ, we solve the following multi-objective optimization problem:

$$\max_{\delta:\|\delta\|\leq\epsilon} \left[f_1\left(\delta\right), ..., f_m\left(\delta\right)\right],$$

where $f_i\left(\delta\right) = \ell\left(h\left(x_i^a\right), y_i\right) = \ell\left(h\left(x_i + \delta\right), y_i\right)$ with y_i the ground-truth label of x_i.

Generating adversarial examples against transformations. Considering a single classification model h and m categories of data transformation $\mathcal{P}_{1:m}$ (e.g., rotation, lighting, and translation). Our goal is to find an adversarial attack that is robust to these data transformations. Specifically, given a benign example x, we would like to learn a perturbation δ with $\|\delta\| \leq \epsilon$ that can successfully attack the model after any transformation $t_i \sim \mathcal{P}_i$ is applied. To formulate as an MOO problem, we consider the task $\mathcal{T}_i$ as finding the adversarial example $x_i^a = t_i\left(x + \delta\right)$ with $t_i \sim \mathcal{P}_i$. For each task $\mathcal{T}_i$, we can define the goal as finding successfully the adversarial example x_i^a:

$$\mathbb{I}\left\{\text{argmax}_{1\leq k\leq M} h\left(x_i^a, k\right) \neq \text{argmax}_{1\leq k\leq M} h\left(x, k\right)\right\}.$$

To find the perturbation δ, we solve the following multi-objective optimization problem:

$$\max_{\delta:\|\delta\|\leq\epsilon} \left[f_1\left(\delta\right), ..., f_m\left(\delta\right)\right],$$

where $f_i\left(\delta\right) = \mathbb{E}_{t_i\sim\mathcal{P}_i}\left[\ell\left(h\left(t_i\left(x + \delta\right)\right), y\right)\right]$ with y the ground-truth label of x.

4 Experiments

In this section, we provide extensive experiments across four settings: (i) generating adversarial examples for ensemble of models (ENS, Sec 4.1), (ii) generating universal perturbation (UNI, Sec 4.3) , (iii) generating robust adversarial examples against Ensemble of Transformations (EoT, Sec 4.4), and (iv) adversarial training for ensemble of models (AT, Sec 4.2). The details of each setting can be found in Appendix C.

General settings. Through our experiments, we use six common architectures for the classifier including ResNet18 (He et al., 2016), VGG16 (Simonyan & Zisserman, 2014), GoogLeNet (Szegedy et al., 2015), EfficientNet (Tan & Le, 2019), MobileNet Howard et al. (2017), and WideResNet Zagoruyko & Komodakis (2016) with the implementation [1]. We evaluate on the full testing set of two benchmark datasets which are CIFAR10 and CIFAR100 (Krizhevsky et al., 2009). We observed that the attack performance is saturated with standard training models. Therefore, to make the job of adversaries more challenging, we use Adversarial Training with PGD-AT (Madry et al., 2018) to robustify the models and use these robust models as the victim models in our experiments.

Evaluation metrics. We use three metrics to evaluate the attack performance including (i) A-All: the Attack Success Rate (ASR) when an adversarial example can achieve goals in all tasks. This is considered as the most important metric to indicate how well one method can achieve in all tasks; (ii)A-Avg: the average Attack Success Rate over all tasks which indicate the average attacking performance; (iii)$\{\text{A-i}\}_{i=1}^K$: Attack Success Rate in each individual task. For reading comprehension purposes, if necessary the highest/second highest performance in each experimental setting is highlighted in **Bold**/Underline and the most important metric(s) is emphasized in blue color.

Baseline methods. We compare our method with the **Uniform** strategy which assigns the same weight for all tasks and the **MinMax** method (Wang et al., 2021) which examines only the worst-case performance across all tasks. To increase the generality to other tasks, MinMax requires a regularization to balance

[1] https://github.com/kuangliu/pytorch-cifar

Table 1: Evaluation of Attacking Ensemble model on the CIFAR10 and CIFAR100 datasets.

		CW		CE		KL	
		A-All	A-Avg	A-All	A-Avg	A-All	A-Avg
CIFAR10	Uniform	26.37	**41.13**	28.21	48.34	17.44	32.85
	MinMax	27.53	41.20	35.75	**51.56**	19.97	**33.13**
	MOO	18.87	34.24	25.16	44.76	15.69	29.54
	TA-MOO	**30.65**	40.41	**38.01**	51.10	**20.56**	31.42
CIFAR100	Uniform	52.82	**67.39**	55.86	72.62	38.57	**54.88**
	MinMax	54.96	66.92	63.70	75.44	40.67	53.83
	MOO	51.16	65.87	58.17	73.19	39.18	53.44
	TA-MOO	**55.73**	67.02	**64.89**	**75.85**	**41.97**	53.76

between the average and the worst-case performance. We use the same attack setting for all methods: the attack is the L_∞ untargeted attack with 100 steps, step size $\eta_\delta = 2/255$ and perturbation limitation $\epsilon = 8/255$. The GD solver in TA-MOO uses 10 steps with learning rate $\eta_w = 0.005$. Further detail can be found in Appendix C.

4.1 Adversarial Examples for Ensemble of Models (ENS)

Experimental setting. In our experiment, we use an ensemble of four adversarially trained models: ResNet18, VGG16, GoogLeNet, and EfficientNet. The architecture is the same for both the CIFAR10 and CIFAR100 datasets except for the last layer which corresponds with the number of classes in each dataset. The final output of the ensemble is an average of the probability outputs (i.e., output of the softmax layer). We use three different losses as an object for generating adversarial examples including CE (Madry et al., 2018), KL (Zhang et al., 2019), and CW (Carlini & Wagner, 2017).

Results 1: TA-MOO achieves the best performance. Table 1 shows the results of attacking the ensemble model on the CIFAR10 and CIFAR100 datasets. It can be seen that TA-MOO significantly outperforms the baselines and achieves the best performance in all the settings. For example, the improvement over the Uniform strategy is around 10% on both datasets with the CE loss. Comparing to the MinMax method, the biggest improvement is around 3% for CIFAR10 with CW loss and the lowest one is around 0.6% with the KL loss. The improvement can be observed in all the settings, showing the generality of the proposed method.

Results 2: When does not MOO work? It can be observed that MOO falls behind all other methods, even compared with the Uniform strategy. Our hypothesis for the failure of MOO is that in the original setting with an ensemble of 4 diverse architectures (i.e., ResNet18, VGG16, GoogLeNet, and EfficientNet) there is one task that dominates the others and makes MOO become trapped (i.e., focusing on improving the dominant task). To verify our hypothesis, we measure the gradient norm $\|\nabla_\delta f_i(\delta)\|$ corresponding to each model and the final weight w of 1000 samples and report the results in Table 2. It can be seen that the EfficientNet has a much lower gradient strength, therefore, it has a much higher weight. This explains the highest ASR observed in EfficientNet and the large gap of 19% (56.11% in EfficientNet and 37.05% in GoogLeNet). To further confirm our hypothesis, we provide an additional experiment on a non-diverse ensemble model which consists of 4 individual ResNet18 models. It can be observed that in the non-diverse setting, the gradient strengths are more balanced across models, indicating that no task dominates others. As a result, MOO shows its effectiveness by outperforming the Uniform strategy by 4.3% in A-All.

Results 3: The importance of the Task-Oriented regularization. It can be observed from Table 2 that in the diverse setting, TA-MOO has a much lower gap (4%) between the highest ASR (53.4% at EfficientNet) and the lowest one (49.29% at GoogLeNet) compared to MOO (19%). Moreover, while the ASR of EfficientNet is lower by 2.7%, the ASRs of all other architectures have been improved considerably (i.e., 12% in GoogLeNet). This improvement shows the importance of the Task-Oriented regularization, which helps to avoid being trapped by one dominating task, as happened in MOO. For the non-diverse

Table 2: Attacking Ensemble model with a diverse set D={R-ResNet18, V-VGG16, G-GoogLeNet, E-EfficientNet} and non-diverse set ND={4 ResNets}. w represents the final w of MOO (mean ± std). $\|\nabla_\delta f_i(\delta)\|$ represents the gradient norm of each model (mean ± std).

		A-All	A-Avg	R/R1	V/R2	G/R3	E/R4
D	$\|\nabla_\delta f_i(\delta)\|$	-	-	7.15 ± 6.87	4.29 ± 4.64	7.35 ± 7.21	0.98 ± 0.72
	w	-	-	0.15 ± 0.14	0.17 ±0.13	0.15 ± 0.14	0.53 ± 0.29
	Uniform	28.21	48.34	48.89	49.08	48.38	47.03
	MOO	25.16	44.76	39.06	46.83	37.05	56.11
	TA-MOO	38.01	51.10	49.55	52.15	49.29	53.40
ND	$\|\nabla_\delta f_i(\delta)\|$	-	-	8.41 ± 8.22	6.68± 6.95	7.36 ± 6.03	5.67 ± 6.09
	w	-	-	0.23 ± 0.21	0.24±0.17	0.23 ± 0.19	0.30 ± 0.21
	Uniform	28.17	48.75	51.94	45.55	54.15	43.34
	MOO	32.50	52.21	53.25	49.05	56.80	49.76
	TA-MOO	41.01	57.33	58.88	55.32	60.81	54.29

Table 3: Evaluation on the Transferability of adversarial examples. Each cell (row-ith, column-jth) reports SAR (higher is better) of adversarial examples from the same source architecture (RME) with an adversary at row-ith to attack an ensemble at column-jth. Each architecture has been denoted by symbols such as R: ResNet18, M: MobileNet, E: EfficientNet, V: VGG16, W: WideResNet. For examples, RME represents for an ensemble of ResNet18, MobileNet and EfficientNet.

	RME	RVW	EVW	MVW	REV	MEV	RMEV	RMEVW
Uniform	31.73	25.03	22.13	22.73	29.50	28.44	26.95	20.50
MinMax	40.01	23.75	22.39	23.34	32.57	32.75	31.85	21.99
MOO	35.20	24.25	22.94	23.76	30.65	32.28	29.49	21.77
TA-MOO	**40.97**	**25.13**	**23.59**	**24.38**	**33.00**	**33.05**	**32.14**	**23.04**

setting, when no task dominates others, TA-MOO still shows its effectiveness when improving the ASR in all tasks by around 5%. The significant improvement can be observed in all settings (except the setting on EfficientNet with the CIFAR10 dataset) as shown in Table 1, and demonstrates the generality of the Task-Oriented regularization.

Results 4: TA-MOO achieves the best transferability on a diverse set of ensembles.

Table 3 reports the SAR-All metric of transferred adversarial examples crafted from a source ensemble (RME) on attacking target ensembles (e.g., RMEVW is an ensemble of 5 models). A higher number indicates a higher success rate of attacking a target model, therefore, also implies a higher transferability of adversarial examples. It can be seen that our TA-MOO adversary achieves the highest attacking performance on the whitebox attack setting, with a huge gap of 9.24% success rate over the Uniform strategy. Our method also achieves the highest transferability regardless diversity of a target ensemble. More specifically, on target models such as REV, MEV, and RMEV, where members in the source ensemble (RME) are also in the target ensemble, our TA-MOO significantly outperforms the Uniform strategy, with the highest improvement is 5.19% observed on target model RMEV. On the target models EVW and MVW which are less similar to the source model, our method still outperforms the Uniform strategy by 1.46% and 1.65%. The superior performance of our adversary on the transferability shows another benefit of using multi-objective optimization in generating adversarial examples. By reaching the intersection of all members' adversarial regions, our adversary is capable to generate a common vulnerable pattern on an input image shared across architectures, therefore, increasing the transferability of adversarial examples. More discussion can be found in Appendix D.1.

Table 4: Robustness evaluation of Adversarial Training methods on the CIFAR10 dataset. RME represents an ensemble of ResNet18 (R), MobileNet (M) and EfficientNet E), while MobiX3 represents an ensemble of three MobileNets. NAT and ADV measure the natural accuracy and the robust accuracy against PGD-Linf attack (↑the higher the better). Other metrics measure the success attack rate (SAR) of adversarial examples generated by the same PGD-Linf attack on fooling each single member and all members of the ensemble (↓the lower the better).

	MobiX3				RME			
	NAT↑	ADV↑	A-All↓	A-Avg↓	NAT↑	ADV↑	A-All↓	A-Avg↓
PGD-AT	**80.43**	32.78	54.34	73.89	**86.52**	37.36	49.01	69.75
MinMax-AT	79.01	37.28	50.28	**66.77**	83.16	40.40	46.91	65.73
MOO-AT	79.38	33.04	**46.28**	74.36	82.04	37.48	45.24	70.11
TA-MOO-AT	79.22	**38.22**	48.21	67.83	82.59	**41.32**	**43.68**	**65.09**

4.2 Adversarial Training with TA-MOO for Ensemble of Models (ENS)

We conduct adversarial training with adversarial examples generated by MOO and TA-MOO attacks to verify the quality of these adversarial examples and report results on Table 4. The detailed setting and more experimental results can be found in Appendix D.2. **Result 1: Reducing transferability.** It can be seen that the SAR-All of MOO-AT and TA-MOO-AT are much lower than that on other methods. More specifically, the gap of SAR-All between PGD-AT and TA-MOO-AT is (5.33%) 6.13% on the (non) diverse setting. The lower SAR-All indicating that adversarial examples are harder to transfer among ensemble members on the TA-MOO-AT model than on the PGD-AT model. **Result 2: Producing more robust single members.** The comparison of average SAR shows that adversarial training with TA-MOO produces more robust single models than PGD-AT does. More specifically, the average robust accuracy (measured by 100% - *A-Avg*) of TA-MOO-AT is 32.17%, an improvement of 6.06% over PGD-AT in the non-diverse setting, while there is an improvement of 4.66% in the diverse setting. **Result 3: Adversarial training with TA-MOO achieves the best robustness.** More specifically, on the non-divese setting, TA-MOO-AT achives 38.22% robust accuracy, an improvement of 1% over MinMax-AT and 5.44% over standard PGD-AT. On the diverse setting, the improvement over MinMax-AT and PGD-AT are 0.9% and 4%, respectively. The root of the improvement is the ability to generate stronger adversarial examples in the the sense that they can challenge not only the entire ensemble model but also all single members. These adversarial examples lie in the joint insecure region of members (i.e., the low confidence region of multiple classes), therefore, making the decision boundaries more separate. As a result, adversarial training with TA-MOO produces more robust single models (i.e., lower SAR-Avg) and significantly reduces the transferability of adversarial examples among members (i.e., lower SAR-All). These two conditions explain the best ensemble adversarial robustness achieved by TA-MOO.

4.3 Universal Perturbation (UNI)

Experimental setting. We follow the experimental setup in Wang et al. (2021), where the full test set (10k images) is randomly divided into equal-size groups (K images per group). The comparison has been conducted on the CIFAR10 and CIFAR100 datasets, with an adversarially trained ResNet18 model and CW loss. We observed that the ASR-All was mostly zero, indicating that it is difficult to generate a general perturbation for all data points. Therefore, in Table 5 we use ASR-Avg to compare the performances of the methods. More experiments on VGG16 and EfficientNet models can be found in Appendix D.3.

Results. Table 5 shows the evaluation of generating universal perturbations on the CIFAR10 and CIFAR100 datasets, respectively. K represents the number of images that are using the same perturbation. The larger the value of K, the harder it is to generate a universal perturbation that can be applied successfully to all images. It can be seen that with a small number of tasks (i.e., K=4), MOO and TA-MOO achieve lower performance than the MinMax method. However, with a large number of tasks (i.e, $K \geq 8$), MOO and TA-MOO show their effectiveness and achieve the best performance. More specifically, on the CIFAR10 dataset, the improvements of MOO over the Uniform strategy are 5.6%, 4%, 3.2%, and 2.5% with $K = 8$,

Table 5: Evaluation of generating Universal Perturbation on the CIFAR10 and CIFAR100 datasets.

	CIFAR10					CIFAR100				
	K=4	K=8	K=12	K=16	K=20	K=4	K=8	K=12	K=16	K=20
Uniform	37.52	30.34	27.41	25.52	24.31	65.40	58.99	55.33	53.02	51.49
MinMax	**50.13**	33.68	20.46	15.74	14.73	**74.73**	62.29	52.05	45.26	42.33
MOO	43.80	35.92	31.41	28.75	26.83	69.35	62.72	57.72	54.12	52.25
TA-MOO	48.00	**39.31**	**34.96**	**31.84**	**30.12**	72.74	**68.06**	**62.33**	**57.48**	**54.12**

Table 6: Robust adversarial examples against transformations evaluation. I: Identity, H: Horizontal flip, V: Vertical flip, C: Center crop, G: Adjust gamma, B: Adjust brightness, R: Rotation.

		A-All	A-Avg	I	H	V	C	G	B	R
C10	Uniform	25.98	**55.33**	**44.85**	41.58	82.90	**72.56**	**45.92**	**49.59**	**49.93**
	MinMax	30.54	52.20	43.31	41.59	78.80	64.83	44.38	46.53	45.97
	MOO	21.25	49.81	36.23	33.93	**87.47**	71.05	37.68	40.21	42.12
	TA-MOO	**31.10**	55.26	44.15	**41.86**	85.19	71.86	45.53	48.70	49.54
C100	Uniform	56.19	76.23	**70.43**	69.01	87.66	87.36	71.40	74.25	73.47
	MinMax	59.75	75.72	70.13	69.26	87.45	86.03	71.54	73.30	72.32
	MOO	53.17	74.21	66.96	65.68	**89.16**	87.03	68.49	71.11	71.06
	TA-MOO	**60.88**	**76.71**	**70.43**	**69.37**	89.11	**87.95**	**71.70**	**74.73**	**73.69**

$K = 12$, $K = 16$, and $K = 20$, respectively. On the same setting, TA-MOO significantly improves MOO by around 4% in all the K settings and consistently achieves the best performance. Unlike the ENS setting, in the UNI setting, MOO consistently achieves better performance than the Uniform strategy . This improvement can be explained by the fact that in the UNI setting with the same architecture and data transformation, no task dominates the others. There will be a case (a group) when one sample is extremely close to/far from the decision boundary, and hence easier/harder to fool. However, in the entire test set with a large number of groups, the issue of dominating tasks is lessened.

4.4 Robust Adversarial Examples against Transformations (EoT)

Results. Table 6 shows the evaluation on the CIFAR10 and CIFAR100 datasets with 7 common data transformations. It can be observed that (i) MOO has a lower performance than the baselines, (ii) the Task Oriented regularization significantly boosts the performance, and (iii) our TA-MOO method achieves the best performance on both settings and outperforms the MinMax method 0.6% and 1.1% in the CIFAR10 and the CIFAR100 experiments, respectively. The low performance of MOO in observation (i) is again caused by the issue of one task dominating others. In the EoT setting, it is because of the V-vertical flip transformation as shown in Table 6. Observation (ii) provides another piece of evidence to support the effectiveness of the Task-Oriented regularization for MOO. This regularization boosts the ASRs in all the tasks (except V - the dominant one), increases the average ASR by 5.45% and 2.5% in the CIFAR10 and CIFAR100 experiments, respectively.

4.5 Additional Experiments with Multi-Task Learning Methods

In this section we would like to provide additional experiments with recent multi-task learning methods to explore how better constrained approaches can improve over the naive MOO. We applied three recent multi-task learning methods including PCGrad Yu et al. (2020), CAGrad Liu et al. (2021a), and HVM Albuquerque et al. (2019) with implementation from their official repositories into our adversarial generation task. We apply the best practice in Albuquerque et al. (2019) which is adaptively updated the Nadir point based on the current tasks' losses. For PCGrad we use the *mean* as the reduction mode. For CAGrad we use parameter $\alpha = 0.5$ and rescale $= 1$ as in their default setting. We experiment on attacking ensemble

Table 7: Attacking Ensemble model with a diverse set D={R-ResNet18, V-VGG16, G-GoogLeNet, E-EfficientNet} and non-diverse set ND={4 ResNets}.

		A-All	A-Avg	R/R1	V/R2	G/R3	E/R4
D	Uniform	28.21	48.34	48.89	49.08	48.38	47.03
	HVM	29.88	46.98	48.97	48.10	46.88	43.96
	PCGrad	28.25	48.28	48.81	49.03	48.13	47.14
	CAGrad	30.23	48.34	47.03	48.22	45.92	52.20
	MOO	25.16	44.76	39.06	46.83	37.05	56.11
	TA-MOO	38.01	51.10	49.55	52.15	49.29	53.40
ND	Uniform	28.17	48.75	51.94	45.55	54.15	43.34
	HVM	28.46	49.87	51.64	50.03	50.72	47.10
	PCGrad	28.30	48.75	52.02	45.42	54.35	43.21
	CAGrad	35.22	51.07	54.22	47.84	55.24	46.97
	MOO	32.50	52.21	53.25	49.05	56.80	49.76
	TA-MOO	41.01	57.33	58.88	55.32	60.81	54.29

of models setting with two settings, a diverse set D with 4 different architectures including R-ResNet18, V-VGG16, G-GoogLeNet, E-EfficientNet and a non-diverse set ND with 4 ResNet18 models.

It can be seen from the Table 7 that in the diverse ensemble setting, the three additional methods HVM, PCGrad and CAGrad significantly outperform the standard MOO method with the improvement gaps of SAR-All around 4.7%, 3% and 5%, respectively. In the non-diverse ensemble setting, while HVM and PCGrad achieve lower performances than the standard MOO method, CAGrad can outperform the MOO method with a 2.7% improvement. On comparison to the naive uniform method, the three methods also achieve better performance in both settings.

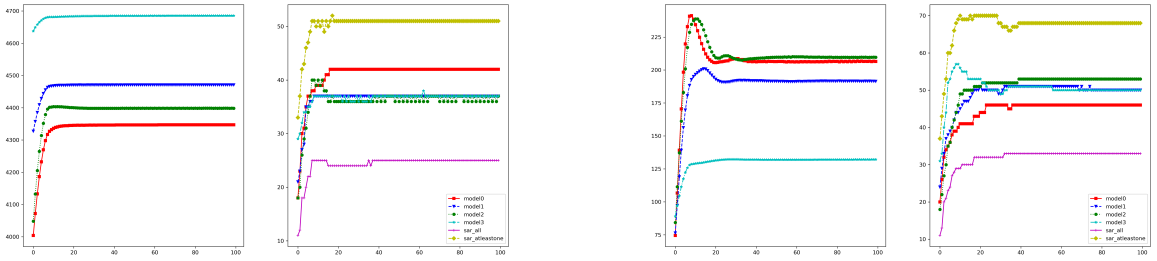
The improvement on the diverse set of HVM, PCGrad and CAGrad over the standard MOO method is more noticeable than on the non-diverse set. It can be explained by the fact that on the diverse set of model architectures, there is a huge difference in gradients among architectures, therefore, requires a better multi-task learning method to handle the constraint between tasks.

On the other hand, on both ensemble settings, our TA-MOO still achieves the best performance, with a huge gap of (5.8%) 7.8% compared to the second best method on the (non) diverse setting. It is because our method can leverage a supervised signal from knowing whether a task is achieved or not to focus on improving unsuccessful tasks. It is a huge advantage compared to unsupervised multi-task learning methods as MOO, HVM, PCGrad, and CAGrad.

5 Additional Discussion

In this section, we would like to summarize some important observations through all experiments while the complete discussion with detail can be found in Appendix E.

Correlation between the objective loss and attack performance. It is broadly accepted that to fool a model, a feasible approach is maximizing the objective loss (i.e., CE, KL, or CW loss), and the higher the loss, the higher the attack success rate. While it is true when observing the same architecture, we found that it is not necessarily true when comparing different architectures. As shown in Figure 1, with CW loss as the adversarial objective, it can be observed that there is a positive correlation between the loss value and the ASR, i.e., the higher the loss, the higher the ASR. However, there is no clear correlation observed when using CE and KL loss. Therefore, the higher weighted loss does not directly imply a higher success rate for attacking an ensemble of different architectures. The MinMax method (Wang et al., 2021) which solely



(a) CW (b) CE

Figure 1: Loss (left fig) and ASR (right fig) of each task over all attack iterations with the MinMax method. model0/1/2/3 represents R/V/G/E architecture, respectively.

weighs tasks' losses, therefore, is not always appropriate to achieve a good performance in all tasks. More discussion can be found in Appendix E.4.

When does MOO work? On the one hand, the dominating issue is observed in all three settings (ENS, UNI, EoT). The issue can be recognized by the gap of attack performance among tasks or by observing the dominating of one task's weight over others which is caused by a significant small gradient strength of one task on comparison with other tasks' strength as discussed in Section 4.1. The root of the dominating issue can be the natural of the setting (i.e., as in EoT setting, when the large gap can be observed in all methods) or the MOO solver.

On the other hand, if overcoming this issue, MOO can outperform the Uniform strategy as shown in Section 4.1. As discussed in Appendix 4.4, a simple memory can helps to overcome the infinite gradient issue and significantly boosts the performance of MOO or TA-MOO. Therefore, we believe that developing a technique to lessen the dominating issue might be a potential extension.

More efficient MOO solvers. Inspired by Sener & Koltun (2018), in this paper we use multi-gradient descent algorithm (Deb, 2011) as a MOO solver which casts the multi-objective problem to a single-objective problem. However, while Sener & Koltun (2018) used Frank-Wolfe algorithm to project the weight into the desired simplex, we use parameterization with softmax to do the job. While this technique is much faster than Frank-Wolfe algorithm, it has some weaknesses that might be target for future works. First, it cannot handle well the edge case which is the root of the dominating issue. Second, it does not work well in the case of a non-convex objective space as similar as other MOO scalarizing methods (Deb, 2011).

6 Conclusion

In this paper, we propose Task Oriented Multi-Objective Optimization (TA-MOO), with specific applications to adversarial generation tasks. We develop a geometry-based regularization term to favor the goal-unachieved tasks, while trying to maintain the the goal-achieved tasks. We conduct comprehensive experiments to showcase the merit of our proposed approach on generating adversarial examples and adversarial training. On the other hand, there are acknowledged limitations of our method such as weaknesses of the gradient-based solver and lacking theory on algorithm's convergence which might be target for future works.

Acknowledgements

This work was partially supported by the Australian Defence Science and Technology (DST) Group under the Next Generation Technology Fund (NGTF) scheme. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

Isabela Albuquerque, Joao Monteiro, Thang Doan, Breandan Considine, Tiago Falk, and Ioannis Mitliagkas. Multi-objective training of generative adversarial networks with multiple discriminators. In International Conference on Machine Learning, pp. 202–211. PMLR, 2019.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In International Conference on Machine Learning, pp. 274–283, 2018.

Emil Björnson and Eduard Jorswieck. Optimal resource allocation in coordinated multi-cell systems. Now Publishers Inc, 2013.

Emil Bjornson, Eduard Axel Jorswieck, Mérouane Debbah, and Bjorn Ottersten. Multiobjective signal processing optimization: The way to balance conflicting metrics in 5g systems. IEEE Signal Processing Magazine, 31(6):14–23, 2014.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.

Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. In Advances in Neural Information Processing Systems, pp. 12861–12871, 2019.

Anh Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving adversarial robustness by enforcing local and global compactness. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII, pp. 209–223. Springer, 2020.

Anh Bui, Trung Le, He Zhao, Paul Montague, Seyit Camtepe, and Dinh Phung. Understanding and achieving efficient robustness with adversarial supervised contrastive learning. arXiv preprint arXiv:2101.10027, 2021a.

Anh Tuan Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving ensemble robustness by collaboratively promoting and demoting adversarial robustness. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pp. 6831–6839, 2021b.

Anh Tuan Bui, Trung Le, Quan Hung Tran, He Zhao, and Dinh Phung. A unified wasserstein distributional robustness framework for adversarial training. In International Conference on Learning Representations, 2022.

Rafael Caballero, Lourdes Rey, Francisco Ruiz, and Mercedes González. An algorithmic package for the resolution and analysis of convex multiple objective problems. In Multiple criteria decision making, pp. 275–284. Springer, 1997.

N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pp. 39–57. IEEE, 2017.

Carlos A Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. Knowledge and Information systems, 1(3):269–308, 1999.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. arXiv preprint arXiv:2003.01690, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2021. URL https://openreview.net/forum?id=SSKZPJCt7B.

Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In Multi-objective evolutionary optimisation for product design and manufacturing, pp. 3–34. Springer, 2011.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. Comptes Rendus Mathematique, 350(5-6):313–318, 2012.

Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. Advances in Neural Information Processing Systems, 33, 2020.

Matthias Ehrgott. Multicriteria optimization, volume 491. Springer Science & Business Media, 2005.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL `http://arxiv.org/abs/1412.6572`.

Pengxin Guo, Yuancheng Xu, Baijiong Lin, and Yu Zhang. Multi-task adversarial attack. arXiv preprint arXiv:2011.09824, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6):82–97, 2012.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In Artificial intelligence safety and security, pp. 99–112. Chapman and Hall/CRC, 2018.

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. Advances in neural information processing systems, 32, 2019.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. Advances in Neural Information Processing Systems, 34:18878–18890, 2021a.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. Advances in neural information processing systems, 29, 2016.

Xingchao Liu, Xin Tong, and Qiang Liu. Profiling pareto front with multi-objective stein variational gradient descent. Advances in Neural Information Processing Systems, 34, 2021b.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations, 2018.

Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In International Conference on Machine Learning, pp. 6597–6607. PMLR, 2020.

Kaisa Miettinen. Nonlinear multiobjective optimization, volume 12. Springer Science & Business Media, 2012.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1765–1773, 2017.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pp. 4970–4979. PMLR, 09–15 Jun 2019.

Haoxuan Qiu, Yanhui Du, and Tianliang Lu. The framework of cross-domain and model adversarial attack against deepfake. Future Internet, 14(2):46, 2022.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? Advances in Neural Information Processing Systems, 33:3533–3545, 2020.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. Advances in neural information processing systems, 31, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

M. Spencer, J. Eickholt, and J. Cheng. A deep learning network approach to ab initio protein secondary structure prediction. IEEE/ACM Trans. Comput. Biol. Bioinformatics, 12(1):103–112, January 2015. ISSN 1545-5963.

Takahiro Suzuki, Shingo Takeshita, and Satoshi Ono. Adversarial example generation using evolutionary multi-objective optimization. In 2019 IEEE Congress on evolutionary computation (CEC), pp. 2136–2144. IEEE, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014. URL http://arxiv.org/abs/1312.6199.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9, 2015.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pp. 6105–6114. PMLR, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pp. 5998–6008, 2017.

Jingkang Wang, Tianyun Zhang, Sijia Liu, Pin-Yu Chen, Jiacen Xu, Makan Fardad, and Bo Li. Adversarial attack generation empowered by min-max optimization. Advances in Neural Information Processing Systems, 34, 2021.

Weiran Wang and Miguel A Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. arXiv preprint arXiv:1309.1541, 2013.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In International Conference on Learning Representations, 2019.

Feiyang Ye, Baijiong Lin, Zhixiong Yue, Pengxin Guo, Qiao Xiao, and Yu Zhang. Multi-objective meta learning. Advances in Neural Information Processing Systems, 34, 2021.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. Advances in Neural Information Processing Systems, 33:5824–5836, 2020.

S. Zagoruyko and N. Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pp. 7472–7482, 2019.

APPENDIX

The Appendix provides technical and experimental details as well as auxiliary aspects to complement the main paper. Briefly, it contains the following:

- Appendix A: Discussion on related work.
- Appendix B: Detailed proof and an illustration of our methods.
- Appendix C: Detailed description of experimental settings.
- Appendix D.1: Additional experiments on transferability of adversarial examples in the ENS setting.
- Appendix D.2: Additional experiments on adversarial training with our methods.
- Appendix D.3: Additional experiments on the UNI setting.
- Appendix D.4: Additional experiments on the EoT setting.
- Appendix D.5: Additional comparison on speed of generating adversarial examples.
- Appendix D.6: Additional experiments on sensitivity to hyper-parameters.
- Appendix D.7: Additional comparison with standard attacks on attacking performance.
- Appendix D.8: Additional experiments on attacking the ImageNet dataset.
- Appendix E.1: Additional discussions on the dominating issue and when MOO can work.
- Appendix E.2: A summary on the importance of Task-Oriented regularization.
- Appendix E.3: Discussion on the limitation of MOO solver.
- Appendix E.4: Discussion on correlation between the objective loss and attack performance.
- Appendix E.5: Discussion on the conflicting between gradients in the adversarial generation task.
- Appendix E.6: Discussion on the convergence of our methods.
- Appendix E.7: Additional experiments with MOO with different initializations.

A Related Work

Multi-Objective Optimization for multi-task learning. (Désidéri, 2012) proposed a multi-gradient descent algorithm for multi-objective optimization (MOO) which opens the door for the applications of MOO in machine learning and deep learning. Inspired by Désidéri (2012), MOO has been applied in multi-task learning (MTL) (Sener & Koltun, 2018; Mahapatra & Rajan, 2020), few-shot learning (Ye et al., 2021), and knowledge distillation (Du et al., 2020). Specifically, the work of Sener & Koltun (2018) viewed multi-task learning as a multi-objective optimization problem, where a task network consists of a shared feature extractor and a task-specific predictor. The work of Mahapatra & Rajan (2020) developed a gradient-based multi-objective MTL algorithm to find a solution that satisfies the user preferences. The work of Lin et al. (2019) proposed a Pareto MTL to find a set of well-distributed Pareto solutions which can represent different trade-offs among different tasks. Recently, the work of Liu et al. (2021b) leveraged MOO with Stein Variational Gradient Descent (Liu & Wang, 2016) to diversify the solutions of MOO. Additionally, the work of Ye et al. (2021) proposed a bi-level MOO which can be applied to few-shot learning. Finally, the work of Du et al. (2020) applied MOO to enable knowledge distillation from multiple teachers.

Generating adversarial examples with single-objective and multi-objective optimizations. Generating qualified adversarial examples is crucial for adversarial training (Madry et al., 2018; Zhang et al., 2019; Bui et al., 2021a; 2022). Many perturbation based attacks have been proposed, notably FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2018), TRADES (Zhang et al., 2019), CW (Carlini & Wagner, 2017), BIM (Kurakin et al., 2018), and AutoAttack (Croce & Hein, 2020). Most adversarial attacks aim to maximize a single objective, e.g., maximizing the cross-entropy (CE) loss w.r.t. the ground-truth label (Madry et al., 2018), maximizing the Kullback-Leibler (KL) divergence w.r.t. the predicted probabilities of a benign example (Zhang et al., 2019), or maximizing the CW loss (Carlini & Wagner, 2017). However, in some contexts, we need to generate adversarial examples maximizing multiple objectives or goals, e.g., attacking multiple models (Pang et al., 2019; Bui et al., 2020) or finding universal perturbations (Moosavi-Dezfooli et al., 2017).

The work of Suzuki et al. (2019) was a pioneering attempt to consider the generation of adversarial examples as a multi-objective optimization problem. The authors proposed a non-adaptive method based on Evolutionary Multi-Objective Optimization (EMOO) Deb (2011) to generate sets of adversarial examples. However, the EMOO method is computationally expensive and requires a large number of evaluations, which limits its practicality. Additionally, the authors applied MOO without conducting an extensive study on the behavior of the algorithm, which could limit the effectiveness of the proposed method. Furthermore, the experimental results presented in the work are limited, which could weaken the evidence for the effectiveness of the proposed method.

To this end, the work of Wang et al. (2021) examined the worst-case scenario by casting the problem of interest as a min-max problem for finding the weight of each task. However, this principle leads to a problem of lacking generality in other tasks. To mitigate the issue, Wang et al. (2021) proposed a regularization to strike a balance between the average and the worst-case performance. The final optimization was formulated as follow:

$$\max_{\delta:\|\delta\|\leq\epsilon} \min_{w\in\Delta_m} \sum_{i=1}^{K} w_i f_i(\delta) + \frac{\gamma}{2}\|w - 1/K\|_2^2,$$

Where $f_i(v)$ is the victim model's loss (i.e., cross entropy loss or KL divergence) and $\gamma > 0$ is the regularization parameter. The authors used the bisection method (Boyd et al., 2004) with project gradient descent for the inner minimization and project gradient ascent for the outer maximization. There are several major differences in comparison to MOO and TA-MOO methods: (i) In principle, MinMax considers the worst-case performance only while our methods improve performance of all tasks simultaneously. (ii) MinMax weighs the tasks' losses to find the minimal weighted sum loss in its inner minimization, however, as discussed in Section E.4 the higher weighted loss does not directly imply the higher success rate in attacking multi-tasks simultaneously. In contrast, our methods use multi-gradient descent algorithm (Deb, 2011) in order to increase losses of all tasks simultaneously. (iii) The original principle of MinMax leads to the biasing problem to the worst-case task. The above regularization has been used to mitigate the issue, however, it considers all tasks equally. In contrast, our TA-MOO takes goal-achievement status of each task into account and focuses more on the goal-unachieved tasks.

Recently, Guo et al. (2020) proposed a multi-task adversarial attack and demonstrated on the universal perturbation problem. However, while Wang et al. (2021) and ours can be classified as an iterative optimization-based attack, Guo et al. (2020) requires a generative model in order to generate adversarial examples. While this line of attack is faster than optimization-based attacks at the inference phase, it requires to train a generator on several tasks beforehand. Due to the difference in setting, we do not compare with that work in this paper.

More recently, Qiu et al. (2022) proposed a framework to attack a generative Deepfake model using the multi-gradient descent algorithm in their backpropagation step. While their method also use the multi-objective optimization for generating adversarial examples, there are several major differences to ours. Firstly, their method aims for a generative Deepfake model while our method aims for the standard classification problem which is the most common and important setting in AML. Secondly, we conduct comprehensive experiments to show that a direct and naive application of MOO to adversarial generation tasks does not work satisfactorily because of the gradient dominating problem. Most importantly, we propose the TA-MOO method which employs a geometry-based regularization term to favor the unsuccessful tasks, while trying to

maintain the performance of the already successful tasks. We have conducted extensive experiments to show that our TA-MOO consistently achieves the best attacking performance across different settings. We also conducted additional experiments with SOTA multi-task learning methods which are PCGrad (Yu et al., 2020) and CAGrad (Liu et al., 2021a) in Section 4.5. Compared to these methods, our TA-MOO still achieves the best attack performance thanks to the Task Oriented regularization.

B Further Details of the Proposed Method

B.1 Proofs

Lemma 1. *Sorting $w_{s+1:m}$ into $u_{s+1:m}$ such that $u_{s+1} \geq u_{s+2} \geq ... \geq u_m$. Defining $\rho = \max\left\{s+1 \leq i \leq m : u_i + \frac{1}{i-s}\left(1 - \sum_{j=s+1}^{i} u_j\right) > 0\right\}$. Denoting $\gamma = \frac{1}{\rho}\left(1 - \sum_{i=s+1}^{\rho} u_i\right)$, the projection $proj_{\mathcal{S}}(w)$ can be computed as*

$$proj_{\mathcal{S}}(w)_i = \begin{cases} 0 & 1 \leq i \leq s \\ \max\{w_i + \gamma, 0\} & otherwise \end{cases}$$

Furthermore, the regularization $\Omega(w)$ has the form:

$$\Omega(w) = \sum_{i=1}^{s} w_i^2 + \sum_{i=s+1}^{m} \left(w_i - \max\{w_i + \gamma, 0\}\right)^2. \tag{5}$$

Proof. The proof is based on Wang & Carreira-Perpinán (2013) with modifications. We need to solve the following OP:

$$\begin{aligned} &\min_{\pi} \frac{1}{2}\|w - \pi\|_2^2 \\ \text{s.t. } &: \pi \geq \mathbf{0} \\ &\|\pi\|_1 = 1. \end{aligned}$$

We note that $\pi_1 = ... = \pi_s = 0$. The OP of interest reduces to

$$\begin{aligned} &\min_{\pi_{s+1:m}} \frac{1}{2}\sum_{i=s+1}^{m} (\pi_i - w_i)^2 \\ \text{s.t. } &: \pi_{s+1:m} \geq \mathbf{0} \\ &\sum_{i=s+1}^{m} \pi_i = 1. \end{aligned}$$

Using the Karush-Kuhn-Tucker (KKT) theorem, we construct the following Lagrange function:

$$\mathcal{L}(\pi, \gamma, \beta) = \frac{1}{2}\sum_{i=s+1}^{m} (\pi_i - w_i)^2 - \gamma\left(\sum_{i=s+1}^{m} \pi_i - 1\right) - \sum_{i=s+1}^{m} \beta_i \pi_i.$$

Setting the derivative w.r.t. π_i to zeros and using the KKT conditions, we obtain:

$$\begin{aligned} \pi_i - w_i - \gamma - \beta_i = 0, \forall i = s+1, ..., m \\ \sum_{i=s+1}^{m} \pi_i = 1 \\ \beta_i \geq 0, \pi_i \geq 0, \beta_i \pi_i = 0, \forall i = s+1, ..., m. \end{aligned}$$

If $\pi_i > 0$, $\beta_i = 0$, hence $\pi_i = w_i + \gamma > 0$. Otherwise, if $\pi_i = 0$, $w_i + \gamma = -\beta_i \leq 0$. Therefore, $w_{s+1:m}$ has the same order as $\pi_{s+1:m}$ and we can arrange them as:

$$\pi_{s+1} \geq \pi_{s+2} \geq ... \geq \pi_\rho > \pi_{\rho-1} = ... = \pi_m = 0.$$

$$u_{s+1} = w_{s+1} \geq u_{s+2} = w_{s+2} \geq .... \geq u_p = w_p \geq u_{\rho-1} = w_{\rho-1} \geq ... \geq u_m = w_m \geq 0.$$

It appears that $1 = \sum_{i=s+1}^{m} \pi_i = \sum_{i=s+1}^{\rho} \pi_i = \sum_{i=s+1}^{\rho} (w_i + \gamma) = \sum_{i=s+1}^{\rho} w_i + (\rho - s)\gamma$. Hence, we gain $\gamma = \frac{1}{\rho - s}\left[1 - \sum_{i=s+1}^{\rho} w_i\right] = \frac{1}{\rho-s}\left[1 - \sum_{i=s+1}^{\rho} u_i\right]$. We now prove that $\rho = \max\left\{s+1 \leq i \leq m : u_i + \frac{1}{i-s}\left(1 - \sum_{j=s+1}^{i} u_j\right) > 0\right\}$. □

- For $i = \rho$, we have

$$u_\rho + \frac{1}{\rho - s}\left(1 - \sum_{j=s+1}^{\rho} u_j\right) = u_\rho + \gamma = w_\rho + \gamma > 0.$$

- For $i < \rho$, we have

$$\begin{aligned}
u_i + \frac{1}{i-s}\left(1 - \sum_{j=s+1}^{i} u_j\right) &= \frac{1}{i-s}\left((i-s)u_i + 1 - \sum_{j=s+1}^{i} u_j\right) \\
&= \frac{1}{i-s}\left[(i-s)w_i + \sum_{j=s+1}^{\rho-1} \pi_j - \sum_{j=s+1}^{i} w_j\right] \\
&= \frac{1}{i-s}\left[(i-s)w_i + \sum_{j=i+1}^{\rho-1} \pi_j + \sum_{j=s+1}^{i} (\pi_j - w_j)\right] \\
&= \frac{1}{i-s}\left[(i-s)(w_i + \gamma) + \sum_{j=i+1}^{\rho-1} \pi_j\right] \\
&= \frac{1}{i-s}\left[(i-s)\pi_i + \sum_{j=i+1}^{\rho-1} \pi_j\right] > 0.
\end{aligned}$$

- For $i > \rho$, we have

$$\begin{aligned}
u_i + \frac{1}{i-s}\left(1 - \sum_{j=s+1}^{i} u_j\right) &= \frac{1}{i-s}\left((i-s)u_i + 1 - \sum_{j=s+1}^{i} u_j\right) \\
&= \frac{1}{i-s}\left((i-s)w_i + \sum_{j=s+1}^{\rho-1} \pi_j - \sum_{j=s+1}^{i} w_j\right) \\
&= \frac{1}{i-s}\left((i-s)w_i + \sum_{j=s+1}^{\rho-1} (\pi_j - w_j) - \sum_{j=\rho}^{i} w_j\right) \\
&= \frac{1}{i-s}\left((i-s)w_i + (\rho - s - 1)\gamma - \sum_{j=\rho}^{i} w_j\right) \\
&= \frac{1}{i-s}\left((\rho - s - 1)(w_i + \gamma) + \sum_{j=\rho}^{i}(w_i - w_j)\right) \leq 0.
\end{aligned}$$

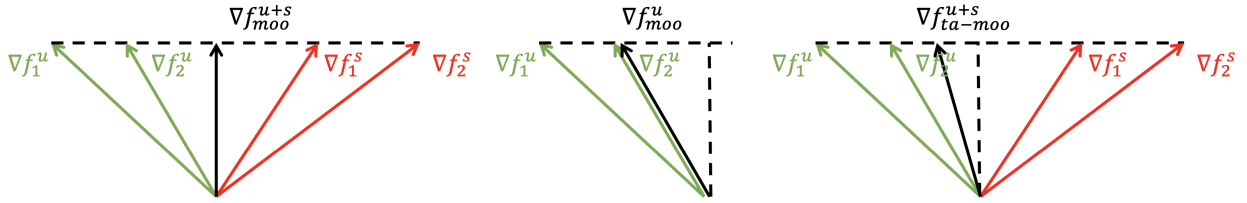


Figure 2: Visualization of standard MOO and TA-MOO solutions in a scenario of 2 goal-achieved tasks ($\nabla f_{1,2}^s$) and 2 goal-unachieved tasks ($\nabla f_{1,2}^u$). (left) MOO; (middle) MOO on the set of goal-unachieved tasks only; (right) TA-MOO with a solution focuses more on the goal-unachieved tasks.

Therefore, $\rho = \max\left\{s+1 \leq i \leq m : u_i + \frac{1}{i-s}\left(1 - \sum_{j=s+1}^{i} u_j\right) > 0\right\}$. Finally, we also have $\pi_i = \max\{w_i + \gamma, 0\}, i = s+1, ..., m$ and $\pi_i = 0, i = 1, ..., s$.

Theorem 1. *The regularization $\Omega(w)$ has the following closed-form:*

$$\Omega(w) = \sum_{i=1}^{s} w_i^2 + \frac{1}{m-s}\left(1 - \sum_{i=s+1}^{m} w_i\right)^2. \tag{6}$$

Proof. Recall $\rho = \max\left\{s+1 \leq i \leq m : u_i + \frac{1}{i-s}\left(1 - \sum_{j=s+1}^{i} u_j\right) > 0\right\}$. Therefore, $\rho = m$ because we have

$$u_m + \frac{1}{m-s}\left(1 - \sum_{j=s+1}^{m} u_j\right) = w_m + \frac{1}{m-s}\left(1 - \sum_{j=s+1}^{m} w_j\right) = w_m + \frac{\sum_{j=1}^{s} w_j}{m-s} > 0.$$

It follows that

$$\gamma = \frac{1}{m-s}\left(1 - \sum_{i=s+1}^{m} u_i\right) = \frac{1}{m-s}\left(1 - \sum_{i=s+1}^{m} w_i\right) \geq 0.$$

$$\operatorname{proj}_{\mathcal{S}}(w)_i = \begin{cases} 0 & 1 \leq i \leq s \\ \max\{w_i + \gamma, 0\} = w_i + \gamma & \text{otherwise} \end{cases}$$

$$\begin{aligned} \Omega(w) &= \sum_{i=1}^{s} w_i^2 + \sum_{i=s+1}^{m} \left(w_i - \max\{w_i + \gamma, 0\}\right)^2 \\ &= \sum_{i=1}^{s} w_i^2 + \sum_{i=s+1}^{m} \gamma^2 = \sum_{i=1}^{s} w_i^2 + (m-s)\gamma^2 \\ &= \sum_{i=1}^{s} w_i^2 + \frac{1}{m-s}\left(1 - \sum_{i=s+1}^{m} w_i\right)^2. \end{aligned}$$

□

B.2 Illustrations of How MOO and TA-MOO Work

Figure 2 illustrates solutions of MOO and TA-MOO in a scenario of 2 goal-achieved tasks (with corresponding gradients $\nabla f_{1,2}^s$) and 2 goal-unachieved tasks (with corresponding gradients $\nabla f_{1,2}^u$). As illustrated in the left figure, with standard MOO method, where all the tasks' gradients have been considered regardless their status and the solution associated with the minimal norm is the perpendicular vector as suggested by geometry (Sener & Koltun, 2018). If considering the goal-unachieved tasks only as in the middle case, the MOO solution

Table 8: Robustness performance of models in the experiments

	CIFAR10		CIFAR100	
	Nat-Acc	Adv-Acc	Nat-Acc	Adv-Acc
ResNet18	86.47	42.14	59.64	18.62
VGG16	84.24	40.88	55.27	16.41
GoogLeNet	88.26	41.26	63.10	19.16
EfficientNet	74.52	41.36	57.67	19.90
MobileNet	76.52	31.12	-	-
WideResNet	88.13	48.62	-	-

is the edge case. However, this extreme strategy ignores all the goal-achieved tasks which might lead to the instability. The Task Oriented regularization strikes a balance between the two aforementioned strategies as illustrated in the right figure. The method focuses more on improving the goal-unachieved tasks while spend less effort to maintain the goal-achieved tasks. With $\lambda = 0$ the TA-MOO optimal solution is equivalent to the standard MOO optimal solution while it becomes the MOO solution in the case of the goal-unachieved tasks only when $\lambda \to \infty$.

C Experimental settings

General settings. Through our experiments, we use six common architectures including ResNet18 (He et al., 2016), VGG16 (Simonyan & Zisserman, 2014), GoogLeNet (Szegedy et al., 2015), EfficientNet (B0) (Tan & Le, 2019), MobileNet Howard et al. (2017), and WideResNet (with depth 34 and widen factor 10) Zagoruyko & Komodakis (2016) with the implementation of `https://github.com/kuangliu/pytorch-cifar`. We evaluate on the full testing set (10k) of two benchmark datasets which are CIFAR10 and CIFAR100 (Krizhevsky et al., 2009). More specifically, the two datasets have 50k training images and 10k testing images, respectively, with the same image resolution of $32 \times 32 \times 3$. However, while the CIFAR10 dataset has 10 classes, the CIFAR100 dataset has 100 classes and fewer images per class. Therefore, in general, an adversary is easier to attack a CIFAR100 model than a CIFAR10 one as shown in Table 8. We observed that the attack performance is saturated with standard training models. Therefore, to make the job of adversaries more challenging, we use Adversarial Training with PGD-AT (Madry et al., 2018) to robustify the models and use these robust models as victim models in our experiments. Specifically, we use the SGD optimizer (momentum 0.9 and weight decay 5×10^{-4}) and Cosine Annealing Scheduler to adjust the learning rate with an initial value of 0.1 and train a model in 200 epochs as suggested in the implementation above. We use PGD-AT L_∞ (Madry et al., 2018) with the same setting for both CIFAR10 and CIFAR100 datasets, i.e., perturbation limitation $\epsilon = 8/255$, $k = 20$ steps, and step size $\eta = 2/255$.

Method settings. In this work, we evaluate all the methods in the untargeted attack setting with L_∞ norm. The attack parameters are the same among methods, i.e., number of attack steps 100, attack budget $\epsilon = 8/255$ and step size $\eta_\delta = 2/255$. In our method, we use K=10 to update the weight in each step with learning rate $\eta_w = 0.005$. Tradeoff parameter $\lambda = 100$ in all experiments. In MinMax (Wang et al., 2021), we use the same $\gamma = 3$ for all settings and use the authors' implementation [2].

Attacking ensemble model settings. In our experiment, we use an ensemble of four adversarially trained models: ResNet18, VGG16, GoogLeNet, and EfficientNet. The architecture is the same for both the CIFAR10 and CIFAR100 datasets except for the last layer which corresponds with the number of classes in each dataset. The final output of the ensemble is an average of the probability outputs (i.e., output of the softmax layer). We use three different losses as an object for generating adversarial examples including Cross Entropy (CE) (Madry et al., 2018), Kullback-Leibler divergence (KL) (Zhang et al., 2019), and CW loss (Carlini & Wagner, 2017).

[2] https://github.com/wangjksjtu/minmax-adv

Table 9: Data transformation setting. U represents uniform sampling function and p represents probability to excuse a transformation (e.g., flipping).

	Deterministic	Stochastic
Identity	Identity	Identity
Horizontal flip	$p = 1$	$p = 0.5$
Vertical flip	$p = 1$	$p = 0.5$
Center crop	scale $= 0.6$	scale $= U(0.6, 1.0)$
Adjust brightness	factor $= 1.3$	factor $= U(1.0, 1.3)$
Rotation	angle $= 10$	angle $= U(-10\ , 10\ )$
Adjust gamma	gamma $= 1.3$	$gamma = U(0.7, 1.3)$

Universal perturbation settings. We follow the experimental setup in Wang et al. (2021), such that the full test set (10k images) is randomly divided into equal-size groups (K images per group). The comparison has been conducted on the CIFAR10 and CIFAR100 datasets, and CW loss. We use adversarial trained ResNet18, VGG16 and EfficientNet as base models. We observed that the ASR-All was mostly zero, indicating that it is difficult to generate a general perturbation for all data points. Therefore, we use ASR-Avg to compare the performances of the methods.

Robust adversarial examples against transformations settings. In our experiment, we use 7 common data transformations including I-Identity, H-Horizontal flip, V-Vertical flip, C-Center crop, B-Adjust brightness, R-Rotation, and G-Adjust gamma. The parameter setting for each transformation has been shown in Table 9. In the deterministic setting, a transformation has been fixed with one specific parameter, e.g., center cropping with a scale of 0.6 or adjusting brightness with a factor of 1.3. While in the stochastic setting, a transformation has been uniformly sampled from its family, e.g., center cropping with a random scale in range (0.6, 1.0) or adjusting brightness with a random factor in range (1.0, 1.3). The experiment has been conducted on adversarially trained ResNet18 model with the CW loss.

D Additional Experiments

D.1 Transferability of adversarial examples in the ENS setting

We conduct an additional experiment to evaluate the transferability of our adversarial examples. We use an ensemble (RME) of three models: ResNet18, MobileNet, and EfficientNet as a source model and apply different adversaries to generate adversarial examples to this ensemble. We then use these adversarial examples to attack other ensemble architectures (target models), for example, RMEVW is an ensemble of 5 models including ResNet18, MobileNet, EfficientNet, VGG16 and WideResNet. Table 10 reports the SAR-All metric of transferred adversarial examples, where a higher number indicates a higher success rate of attacking a target model, therefore, also implies a higher transferability of adversarial examples. The first column (heading RME) shows SAR-All when adversarial examples attack the source model (i.e., the whitebox attack setting).

The Uniform strategy achieves the lowest transferability. It can be observed from Table 10 that the Uniform strategy achieves the lowest SAR in the whitebox attack setting. This strategy also has the lowest transferability in attacking other ensembles (except an ensemble RVW).

MinMax's transferability drops on dissimilar target models. While MinMax achieves the second-best performance in the whitebox attack setting, its adversarial examples have a low transferability when target models are different from the source model. For example, in the target model RVW where there is only one member of the target model from the source model (RME) (i.e., R or ResNet18), MinMax achieves a 23.75% success rate which is lower than the Uniform strategy by 1.28%. Similar observation can be observed on target models EVW and MVW, where MinMax outperforms the Uniform strategy by just 0.2% and 0.6%, respectively.

Table 10: Evaluation on the Transferability of adversarial examples. Each cell (row-ith, column-jth) reports SAR (higher is better) of adversarial examples from the same source architecture (RME) with an adversary at row-ith to attack an ensemble at column-jth. Each architecture has been denoted by symbols such as R: ResNet18, M: MobileNet, E: EfficientNet, V: VGG16, W: WideResNet. For examples, RME represents for an ensemble of ResNet18, MobileNet and EfficientNet. The highest/second highest performance is highlighted in **Bold**/Underline. The table is copied from Table 3 in the main paper for reading comprehension purpose.

	RME	RVW	EVW	MVW	REV	MEV	RMEV	RMEVW
Uniform	31.73	25.03	22.13	22.73	29.50	28.44	26.95	20.50
MinMax	40.01	23.75	22.39	23.34	32.57	32.75	31.85	21.99
MOO	35.20	24.25	22.94	23.76	30.65	32.28	29.49	21.77
TA-MOO	**40.97**	**25.13**	**23.59**	**24.38**	**33.00**	**33.05**	**32.14**	**23.04**

TA-MOO achieves the highest transferability on a diverse set of ensembles . Our TA-MOO adversary achieves the highest attacking performance on the whitebox attack setting, with a huge gap of 9.24% success rate over the Uniform strategy. Our method also achieves the highest transferability regardless diversity of a target ensemble. More specifically, on target models such as REV, MEV, and RMEV, where members in the source ensemble (RME) are also in the target ensemble, our TA-MOO significantly outperforms the Uniform strategy, with the highest improvement is 5.19% observed on target model RMEV. On the target models EVW and MVW which are less similar to the source model, our method still outperforms the Uniform strategy by 1.46% and 1.65%. The superior performance of our adversary on the transferability shows another benefit of using multi-objective optimization in generating adversarial examples. By reaching the intersection of all members' adversarial regions, our adversary is capable to generate a common vulnerable pattern on an input image shared across architectures, therefore, increasing the transferability of adversarial examples.

D.2 Adversarial Training with TA-MOO

Setting. We conduct adversarial training with adversarial examples generated by MOO and TA-MOO attacks to verify the quality of these adversarial examples. We choose an ensemble of 3 MobileNet architectures (non-diverse set) and ensemble of 3 different architectures including ResNet18, MobileNet and EfficientNet (diverse set). To evaluate the adversarial robustness, we compare natural accuracy (NAT) and robust accuracy (ADV) against PGD-Linf attack of these adversarial training methods (the higher the better). We also measure the success attack rate (SAR) of adversarial examples generated by the same PGD-Linf attack on fooling each single member and all members of the ensemble (the lower the better). We use $k = 10, \epsilon = 8/255, \eta = 2/255$ for adversarial training and PGD-Linf with $k = 20, \epsilon = 8/255, \eta = 2/255$ for robustness evaluation. We use SGD optimizer with momentum 0.9 and weight decay 5e-4. Initial learning rate is 0.1 with Cosine Annealing scheduler and train on 100 epochs.

Result 1. Reducing transferability. It can be seen that the SAR-All of MOO-AT and TA-MOO-AT are much lower than that on other methods. More specifically, the gap of SAR-All between PGD-AT and TA-MOO-AT is (5.33%) 6.13% on the (non) diverse setting. The lower SAR-All indicating that adversarial examples are harder to transfer among ensemble members on the TA-MOO-AT model than on the PGD-AT model.

Result 2. Producing more robust single members. The comparison of average SAR shows that adversarial training with TA-MOO produces more robust single models than PGD-AT does. More specifically, the average robust accuracy (measured by 100% - *A-Avg*) of TA-MOO-AT is 32.17%, an improvement of 6.06% over PGD-AT in the non-diverse setting, while there is an improvement of 4.66% in the diverse setting.

Result 3. Adversarial training with TA-MOO achieves the best robustness. More specifically, on the non-divese setting, TA-MOO-AT achives 38.22% robust accuracy, an improvement of 1% over MinMax-

Table 11: Robustness evaluation of Adversarial Training methods on the CIFAR10 dataset. RME represents an ensemble of ResNet18 (R), MobileNet (M) and EfficientNet E), while MobiX3 represents an ensemble of three MobileNets. NAT and ADV measure the natural accuracy and the robust accuracy against PGD-Linf attack (↑the higher the better). Other metrics measure the success attack rate (SAR) of adversarial examples generated by the same PGD-Linf attack on fooling each single member and all members of the ensemble (↓the lower the better). The highest/second highest **robustness** is highlighted in **Bold**/Underline. The most important metric is emphasized in blue.

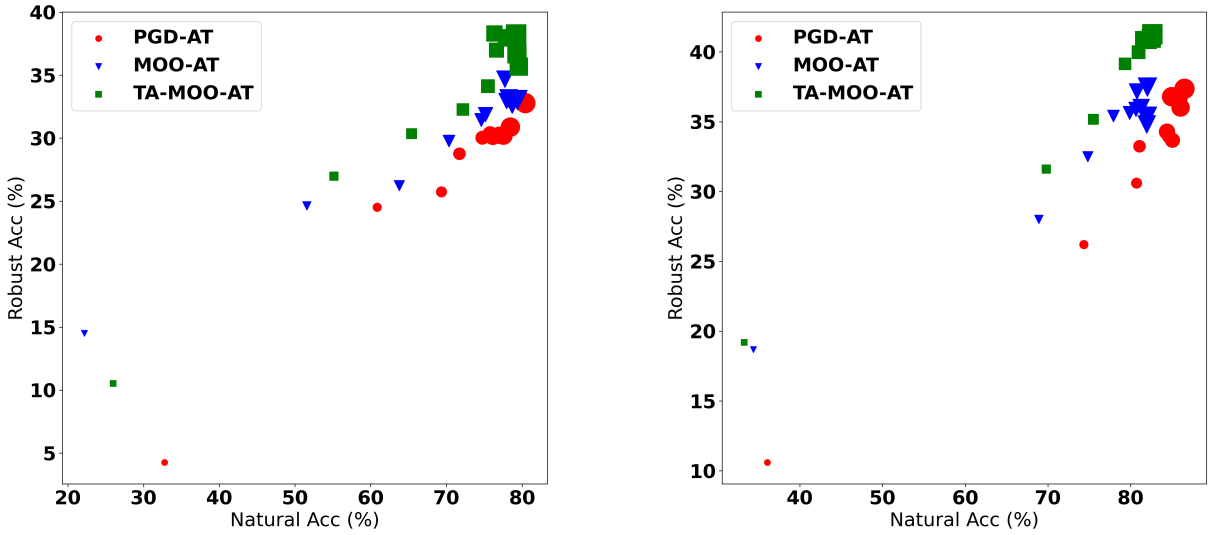
	Arch	NAT↑	ADV↑	A-All↓	A-Avg↓	R/M1↓	M/M2↓	E/M3↓
PGD-AT	MobiX3	**80.43**	32.78	54.34	73.89	76.17	74.35	71.14
MinMax-AT	MobiX3	79.01	37.28	50.28	**66.77**	**65.27**	70.27	**64.78**
MOO-AT	MobiX3	79.38	33.04	**46.28**	74.36	71.25	74.53	77.29
TA-MOO-AT	MobiX3	79.22	**38.22**	48.21	67.83	68.04	**67.37**	68.07
PGD-AT	RME	**86.52**	37.36	49.01	69.75	65.81	75.24	68.21
MinMax-AT	RME	83.16	40.40	46.91	65.73	65.22	**68.28**	63.70
MOO -AT	RME	82.04	37.48	45.24	70.11	69.00	75.43	65.90
TA-MOO-AT	RME	82.59	**41.32**	**43.68**	**65.09**	**63.77**	68.98	**62.51**

AT and 5.44% over standard PGD-AT. On the diverse setting, the improvement over MinMax-AT and PGD-AT are 0.9% and 4%, respectively. The root of the improvement is the ability to generate stronger adversarial examples in the the sense that they can challenge not only the entire ensemble model but also all single members. These adversarial examples lie in the joint insecure region of members (i.e., the low confidence region of multiple classes), therefore, making the decision boundaries more separate. As a result, adversarial training with TA-MOO produces more robust single models (i.e., lower SAR-Avg) and significantly reduces the transferability of adversarial examples among members (i.e., lower SAR-All). These two conditions explain the best ensemble adversarial robustness achieved by TA-MOO.

D.3 Universal Perturbation (UNI)

Additional experimental results. In addition to the experiments on ResNet18 as reported in Table 5, we would like to provide additional experimental results on two other adversarial trained models VGG16 and EfficientNet as shown in Table 12. It can be seen that, TA-MOO consistently achieves the best attacking performance on ResNet18 and VGG16, on both CIFAR10 and CIFAR100 datasets, with $K \geq 8$.

Why does MOO work? As shown in Table 12, MOO consistently achieves better performance than the Uniform strategy (except for the setting with EfficientNet on the CIFAR100 dataset). To find out the reason for the improvement, we investigate the gradient norm $\|\nabla_\delta f(\delta)\|$ and weight w for the first, and second groups (as an example) and the average over 100 groups of the testset as shown in Table 13. It can be seen that in the first and second groups, there are some tasks that have significantly low gradient strengths than other tasks. The gap of the strongest/weakest gradient strength can be a magnitude of 10^6 indicating the domination of one task over others. While this issue can cause the failure as in the ENS setting, however, in the UNI setting, the lowest gradient strengths in each group correspond to unsuccessful tasks (unsuccessful adversarial examples) and vice versa. Recall that we use the multi-gradient descent algorithm to solve MOO, which in principle assigns a higher weight for a weaker gradient vector. Therefore, in the UNI setting, while the dominating issue still exists, fortunately, the result still fits our desired weighting strategy (i.e., higher weight for an unsuccessful task and vice versa). Moreover, when there are a large number of groups (i.e., 100 groups), the issue of dominating tasks is alleviated. The average gradient strength is more balanced as shown in Table 13. This explains the improvement of MOO over the Uniform strategy in the UNI setting.



(a) MobiX3 (b) RME

Figure 3: Comparison progress of three adversarial training methods. The bigger marker size represents the later epoch. Each point represents the natural accuracy and robust accuracy against PGD-Linf attack on the testing set.

Table 12: Evaluation of generating Universal Perturbation on the CIFAR10 and CIFAR100 datasets. R: ResNet18, V: VGG16, E: EfficientNet.

		CIFAR10					CIFAR100				
		K=4	K=8	K=12	K=16	K=20	K=4	K=8	K=12	K=16	K=20
R	Uniform	37.52	30.34	27.41	25.52	24.31	65.40	58.99	55.33	53.02	51.49
	MinMax	**50.13**	33.68	20.46	15.74	14.73	**74.73**	62.29	52.05	45.26	42.33
	MOO	43.80	35.92	31.41	28.75	26.83	69.35	62.72	57.72	54.12	52.25
	TA-MOO	48.00	**39.31**	**34.96**	**31.84**	**30.12**	72.74	**68.06**	**62.33**	**57.48**	**54.12**
V	Uniform	37.76	30.81	27.49	25.94	24.46	66.87	61.49	58.53	56.29	54.98
	MinMax	**47.96**	30.88	20.20	16.93	16.25	**78.58**	69.14	58.85	51.81	48.09
	MOO	43.04	34.56	30.07	27.43	25.42	73.46	66.51	61.28	57.88	56.09
	TA-MOO	46.58	**38.33**	**32.32**	**29.16**	**26.56**	75.57	**71.86**	**67.22**	**62.99**	**59.19**
E	Uniform	44.86	39.03	36.37	34.65	33.49	67.55	60.99	57.35	**54.84**	**53.57**
	MinMax	44.47	32.96	28.86	27.01	26.47	69.69	57.99	50.93	45.59	43.87
	MOO	45.31	**39.28**	**36.44**	**34.72**	**33.51**	66.68	59.69	54.95	53.20	51.43
	TA-MOO	**46.74**	37.95	33.95	31.71	30.41	**70.40**	**63.78**	**58.17**	53.26	50.66

Table 13: Evaluation of generating Universal Perturbation (K=8) on the CIFAR10 dataset with ResNet18 architecture and MOO method. $\{T_i\}_{i=1}^K$ represents value for each task (i.e., a sample in a group). w_1/w_2 represents the weight of the first/second group of K samples, while w represents the the statistic of weight over all groups (mean±std). $\|\nabla_{\delta_1} f_i(\delta_1)\|$ / $\|\nabla_{\delta_2} f_i(\delta_2)\|$ represents the gradient norm of the first/second group of K samples, while $\|\nabla_{\delta} f_i(\delta)\|$ represents the statistic of gradient norm over all groups (mean±std). $\mathbb{I}_0/\mathbb{I}_1$ represents the indicator function for a successful (1) or unsuccessful (0) task, while $\mathbb{I}$ represents the the statistic of successful rate over all groups.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
$\|\nabla_{\delta_1} f_i(\delta_1)\|$	1.15e1	3.45e-5	1.97e-2	1.26e-4	1.27e0	1.04e-1	1.04e1	9.91e0
w_0	0.0238	0.1861	0.1859	0.1861	0.1763	0.1862	0.0257	0.0299
$\mathbb{I}_0$	1	0	0	0	0	0	1	1
$\|\nabla_{\delta_2} f_i(\delta_2)\|$	9.70e0	1.59e1	4.32e-4	4.27e-4	1.25e1	6.23e-5	2.91e-5	6.17e-6
w_1	0.0341	0.0167	0.1854	0.1854	0.0222	0.1854	0.1854	0.1854
$\mathbb{I}_1$	1	1	0	0	1	0	0	0
$\|\nabla_{\delta} f_i(\delta)\|$	4.93±6.63	4.23±6.97	5.18±7.42	3.84±5.83	4.39±6.04	6.66±7.64	4.82±7.48	5.25±7.17
w	0.12±0.08	0.14±0.09	0.12±0.08	0.13±0.08	0.12±0.09	0.10±0.08	0.14±0.10	0.11±0.08
$\mathbb{I}$	0.38±0.49	0.28±0.46	0.36±0.48	0.32±0.48	0.38±0.48	0.48±0.50	0.32±0.47	0.40±0.49

D.4 Robust Adversarial Examples against Transformations (EoT)

We observed that in EoT with the stochastic setting, adjusting gamma sometimes has the overflow issue resulting in an infinite gradient. Recall that our method using MGDA to solve MOO which relies on the stability of gradient strengths. Therefore, in the case of having infinite gradients, learning weight w is unstable, resulting to lower performance in both MOO and TA-MOO.

To overcome the overflow issue, we allocate memory to cache the valid gradient of each task in the previous iteration and replace the infinite value in the current iteration with the valid one in the memory. The storage only requires a tensor with the same shape as the gradient (i.e., as the exact size of the input), therefore, it does not increase the computation resource significantly. As shown in Table 14, this simple technique helps to improve performance of TA-MOO by 5.3% on both the CIFAR10 and CIFAR100 datasets. It also helps to improve performance of MOO by 0.8% and 4.8%, respectively. Finally, after overcoming the gradient issue, the TA-MOO achieves the best performance on the CIFAR100 dataset and the second best performance on the CIFAR10 dataset (0.4% lower in ASR-All but 0.8% higher in ASR-Avg when comparing to MinMax). This result provides additional evidence of the advantage of our method.

D.5 Generating Speed Comparison and Experiments' Stability

Generating Speed Comparison. Table 16 shows the average time to generate one adversarial example in each setting. The results are measured on the CIFAR10 dataset with ResNet18 architecture in the Ensemble of Transformations (EoT) and Universal Perturbation (Uni) settings. We use 1 Titan RTX 24GB for the EoT experiment and 4 Tesla V100 16GB each for the other experiments. It is worth mentioning that our primary focus in this paper is showing the advantage of MOO and the Task-Oriented regularization in generating adversarial examples. Therefore, we did not try to optimize our implementation in terms of generating time.

Experiments' Stability. We conduct an experiment with 5 different random seeds to generate adversarial examples for the ENS setting to evaluate the stability of experimental results on choosing of random seed. The experiment is on the CIFAR10 dataset, with an ensemble of 4 architectures including ResNet18, VGG16, GoogLeNet, and EfficientNet. We report mean and variation values in Table 17. It can be observed that there is a slight variation in attack performances across methods. The variation is small enough compared to

Table 14: Robust adversarial examples against transformations evaluation. The highest/second highest performance is highlighted in **Bold**/Underline. MOO* and TA-MOO* represent version with memory to overcome the infinite gradient issue in the stochastic setting.

		Deterministic		Stochastic	
		A-All	A-Avg	A-All	A-Avg
C10	Uniform	25.98	**55.33**	31.47	**50.55**
	MinMax	30.54	52.20	**33.35**	49.44
	MOO	21.25	49.81	26.97	43.84
	TA-MOO	**31.10**	55.26	28.26	45.67
	MOO*	-	-	27.79	45.91
	TA-MOO*	-	-	32.96	50.27
C100	Uniform	56.19	76.23	59.89	73.73
	MinMax	59.75	75.72	61.30	73.59
	MOO	53.17	74.21	54.96	69.26
	TA-MOO	**60.88**	**76.71**	56.23	69.91
	MOO*	-	-	58.79	72.81
	TA-MOO*	-	-	**61.54**	**74.07**

Table 15: Robust adversarial examples against transformations evaluation. The highest/second highest performance is highlighted in **Bold**/Underline. The most important metric is emphasized in blue color. MOO* and TA-MOO* represent version with memory to overcome the infinite gradient issue in the stochastic setting. I: Identity, H: Horizontal flip, V: Vertical flip, C: Center crop, G: Adjust gamma, B: Adjust brightness, R: Rotation.

		A-All	A-Avg	I	H	V	C	G	B	R
D-C10	Uniform	25.98	**55.33**	**44.85**	41.58	82.90	**72.56**	**45.92**	**49.59**	**49.93**
	MinMax	30.54	52.20	43.31	41.59	78.80	64.83	44.38	46.53	45.97
	MOO	21.25	49.81	36.23	33.93	**87.47**	71.05	37.68	40.21	42.12
	TA-MOO	**31.10**	55.26	44.15	**41.86**	85.19	71.86	45.53	48.70	49.54
D-C100	Uniform	56.19	76.23	**70.43**	69.01	87.66	87.36	71.40	74.25	73.47
	MinMax	59.75	75.72	70.13	69.26	87.45	86.03	71.54	73.30	72.32
	MOO	53.17	74.21	66.96	65.68	**89.16**	87.03	68.49	71.11	71.06
	TA-MOO	**60.88**	**76.71**	**70.43**	**69.37**	89.11	**87.95**	**71.70**	**74.73**	**73.69**
S-C10	Uniform	31.47	**50.55**	**48.58**	44.70	**65.52**	51.14	**47.43**	**48.76**	47.70
	MinMax	**33.35**	49.44	47.35	44.45	62.78	51.75	46.32	47.13	46.34
	MOO	26.97	43.84	40.62	38.45	57.65	48.55	40.41	40.71	40.47
	TA-MOO	28.26	45.67	42.80	39.66	61.98	47.92	41.80	43.01	42.54
	MOO*	27.79	45.91	42.43	39.65	62.11	51.44	41.62	42.21	41.92
	TA-MOO*	32.96	50.27	48.18	**45.26**	62.97	**52.49**	47.03	48.22	**47.76**
S-C100	Uniform	59.89	73.73	**73.19**	**71.15**	79.73	74.81	72.05	73.10	**72.10**
	MinMax	61.30	73.59	72.44	70.55	80.04	75.55	71.99	72.49	**72.10**
	MOO	54.96	69.26	67.62	66.11	75.88	72.72	66.87	68.11	67.49
	TA-MOO	56.23	69.91	68.52	66.92	76.70	72.71	67.57	68.97	67.97
	MOO*	58.79	72.81	71.58	69.08	80.17	75.01	70.78	71.71	71.33
	TA-MOO*	**61.54**	**74.07**	72.95	70.95	**80.94**	**76.22**	**72.22**	**73.21**	72.00

Table 16: Average time per sample for generating adversarial example. All experiments are measured on the CIFAR10 dataset, EoT and Uni are with ResNet18 architecture.

	Ensemble (K=4)	EoT (K=7)	Uni@K=12	Uni@K=20
Uniform	640ms	350ms	1850ms	3030ms
MinMax	1540ms	610ms	1210ms	2080ms
MOO	1770ms	1130ms	5600ms	9280ms
TA-MOO	1960ms	1200ms	5870ms	9500ms

Table 17: Stability of experiments' evaluation on different random seeds. Experiment on the ENS setting, with an ensemble of 4 models: Resnet18, VGG16, GoogleNet and EfficientNet.

	A-All	A-Avg	R	V	G	E
Uniform	28.12 ± 0.09	48.29 ± 0.05	48.81 ± 0.08	49.06 ± 0.08	48.27 ± 0.10	47.06 ± 0.03
MOO	25.61 ± 0.36	45.13 ± 0.30	39.84 ± 0.62	47.29 ± 0.36	37.51 ± 0.36	55.90 ± 0.17
TA-MOO	37.56 ± 0.32	51.15 ± 0.21	49.37 ± 0.15	52.80 ± 0.45	48.98 ± 0.25	53.24 ± 0.13

the gap between methods (i.e., the biggest variation is 0.32% in SAR-All while the smallest gap is 2.51% between MOO and the Uniform approach), therefore, making the comparison still reliable.

D.6 Sensitivity to Hyper-parameters

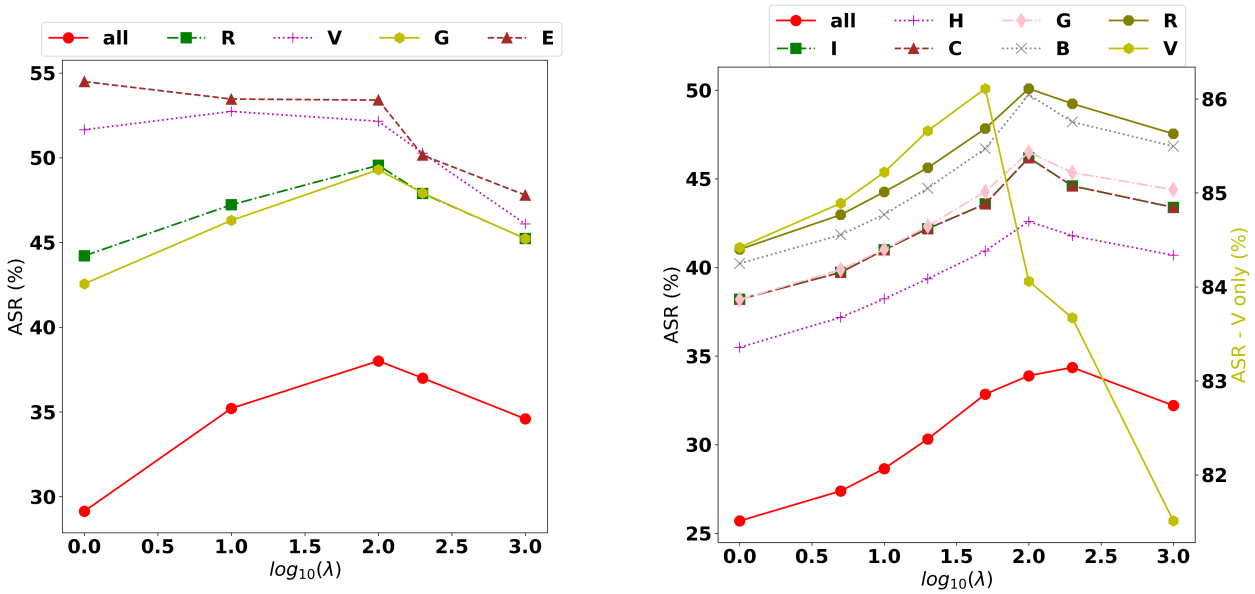
In this section we provide an analytical experiment on the sensitivity of our TA-MOO method to the tradeoff λ. The study has been conducted with the ENS setting with CE loss and the EoT setting with deterministic transformations using ResNet18 architecture. All experiments are on the CIFAR10 dataset. The value of λ is changed from 1 to 1000. It can be observed from Figure 4a (the ENS setting) that (i) increasing λ reduces the performance of dominated task (i.e., ASR on the EfficientNet decreases from 54.49% at $\lambda = 1$ to 53.40% at $\lambda = 100$) while increases performances of other tasks. In overall, it significantly increases the ASR-All performance of the entire ensemble from 29.14% at $\lambda = 1$ to 38.01% at $\lambda = 100$. (ii) However, over-high λ (i.e., $\lambda > 200$) leads to the drop of performance in all tasks, resulting in a lower overall performance.

A similar observation can be seen in the EoT setting in Figure 4b. The attack performance on the dominated task (V-Vertical flipping) decreases from 86.11% at $\lambda = 50$ to 83.67% at $\lambda = 200$. In contract, in the same range of λ the overall performance increases from 32.85% to 34.36%. The performances of all tasks decrease when using too large λ (i.e., $\lambda > 200$). Based on the result of this study, we choose $\lambda = 100$ in all the other experiments.

D.7 Comparison with Standard Attacks

We conducted an additional comparison on the ENS setting to further confirm the effectiveness of our method over standard adversarial attacks (which consider an entire ensemble as a single model). More specifically, we compare with AutoAttack (Croce & Hein, 2020), Brendel-Bethge attack (BB) (Brendel et al., 2019), Carlini-Wagner attack (CW) (Carlini & Wagner, 2017), and PGD attack (Madry et al., 2018). For AutoAttack, we use the standard version which includes 4 different attacks. For BB attack, we initialized with the PGD attack with 20 steps. For CW attack, we set the confidence factor to 1.0. We evaluate these attacks on 2 ensemble settings, a diverse (D) ensemble set with 4 different architectures (ResNet18, VGG16, GoogLeNet, and EfficientNet) and a non-diverse (ND) ensemble set with 4 ResNet18 architectures.

It can be seen from the Table 18 that our TA-MOO attack consistently achieves the best attack performance, with a significant gap compared to the best standard attack. More specifically, our TA-MOO method achieves 38.01% (SAR-All metric) on the diverse ensemble set, while the second best attack is AutoAttack with 30.71% (a gap of 7.3%). On the non-diverse set, the gap between our TA-MOO and AutoAttack is still notably large at 4%. These standard attacks consider an entire ensemble as a single model, i.e., aim to optimize a single objective given a single ensemble output. Therefore, they cannot guarantee a successful attack on each member.



(a) ENS setting

(b) EoT setting

Figure 4: Sensitivity to the parameter λ.

Table 18: Attacking Ensemble model with a diverse set D={R-ResNet18, V-VGG16, G-GoogLeNet, E-EfficientNet} and non-diverse set ND={4 ResNets}. Experiment on the CIFAR10 dataset with cross-entropy objective loss. The most important metric is emphasized in blue.

		A-All	A-Avg	R/R1	V/R2	G/R3	E/R4
D	PGD	28.21	48.34	48.89	49.08	48.38	47.03
	CW	6.10	16.63	13.53	15.76	11.74	25.47
	B&B	6.67	38.03	37.95	38.92	35.58	39.68
	AutoAttack	30.71	45.49	48.32	45.83	47.25	40.56
	MOO	25.16	44.76	39.06	46.83	37.05	**56.11**
	TA-MOO	**38.01**	**51.10**	**49.55**	**52.15**	**49.29**	53.40
ND	PGD	28.17	48.75	51.94	45.55	54.15	43.34
	CW	4.71	13.86	14.92	12.71	17.51	10.31
	B&B	5.29	40.51	49.06	35.19	48.63	29.16
	AutoAttack	37.00	49.32	51.07	48.58	51.08	46.55
	MOO	32.50	52.21	53.25	49.05	56.80	49.76
	TA-MOO	**41.01**	**57.33**	**58.88**	**55.32**	**60.81**	**54.29**

D.8 Attacking the ImageNet dataset

Experimental Setting. We conduct experiments on the ENS setting using the adversarial pre-trained models on the RobustBench (Croce et al., 2021). We use two sets of an ensemble to verify the importance of our task-oriented strategy. The first set is the robust ensemble (RE) set including 3 robust models: ResNet18 (model ID: Salman2020Do_R18 (Salman et al., 2020), robust accuracy 25.32%), ResNet50 (model ID: Salman2020Do_R50 (Salman et al., 2020), robust accuracy 34.96%) and ResNet50 (model ID: Wong2020Fast (Wong et al., 2019), robust accuracy 26.24%). The second set is the less-robust ensemble (LE) which includes 3 models: ResNet18 (model ID: Salman2020Do_R18), ResNet50 (model ID: Salman2020Do_R50) and the

Table 19: Evaluation attacking performance on the ImageNet dataset. RE/LE/TAR/UNTAR represents Robust Ensemble/Less-Robust Ensemble/Targeted Attack/Untargeted Attack, respectively. R18/R50/STD represents robust ResNet18, robust ResNet50 and standard ResNet50 pre-trained model, respectively. The most important metric is emphasized in blue.

		A-All	A-Avg	R18/R18	R50/R50	R50/STD
RE-TAR	Uniform	29.58	39.38	42.50	32.22	43.42
	MOO	29.66	**39.73**	42.86	**32.32**	44.00
	TA-MOO	**29.68**	**39.73**	**42.90**	32.26	**44.02**
LE-TAR	Uniform	30.30	58.14	42.36	32.06	**100.0**
	MOO	30.66	**58.37**	**42.70**	**32.48**	99.94
	TA-MOO	**30.68**	58.25	42.54	32.36	99.86
RE-UNTAR	Uniform	48.58	60.11	64.22	51.72	64.38
	MOO	48.68	**60.20**	**64.30**	51.82	**64.48**
	TA-MOO	**49.80**	59.71	63.80	**52.38**	62.94
LE-UNTAR	Uniform	34.24	61.01	46.98	36.28	99.78
	MOO	44.76	68.29	58.42	46.64	99.80
	TA-MOO	**49.46**	**70.74**	**61.26**	**51.14**	**99.82**

standard training ResNet50 (model ID: Standard_R50, robust accuracy 0%). We use both targeted attack and untargeted attack settings, with $\epsilon = 4/255$, and $\eta = 1/255$ with 20 steps. We use 5000 images of the validation set to evaluate.

Experimental Results. We report experimental results with different settings in Table 19, where RE/LE/-TAR/UNTAR represents Robust Ensemble/Less-Robust Ensemble/Targeted Attack/Untargeted Attack, respectively. It can be seen that, in the robust ensemble setting (RE-TAR and RE-UNTAR), our MOO achieves a similar performance compared to the baseline, while TA-MOO has a further improvement over MOO. The gap of SAR-All between TA-MOO and the uniform weighting strategy is 0.1% in the targeted attack setting (RE-TAR), while that in the untargeted attack setting is 1.2%. In the less-robust ensemble setting (LE-TAR and LE-UNTAR), the improvement of our methods over the baseline is higher than in the robust ensemble setting. With the gap of SAR-All between TA-MOO and the uniform strategy is 0.38% with the targeted attack setting (LE-TAR), while the gap in the untargeted setting (LE-UNTAR) is 15.22% a significantly higher. While it is acknowledged that the targeted attack is a more common protocol in attacking the ImageNet dataset (Athalye et al., 2018), however, we believe that our significant improvement on the untargeted attack is still worth noting.

We conduct an additional experiment on the EoT setting with the ImageNet dataset and report result in Table 20. In this experiment, we use the robust pretrained ResNet18 model (model ID: Salman2020Do_R18) as the victim model. We use the standard attack setting, i.e., targeted attack with $\epsilon = 4/255, \eta = 1/255$ with 20 steps. It can be seen that both MOO and TA-MOO could obtain a better attack performance than the uniform strategy. It is a worth noting that, in the experiment on the CIFAR10/CIFAR100 datasets (i.e., Table 6 in the main paper) the dominating issue of the vertical filliping exists and prevents MOO to obtain a better performance. In the ImageNet dataset, the dominating issue is less serious, therefore, explains the improvement of MOO and corroborates our hypothesis on the issue of dominating task.

Table 20: Evaluation on the EoT setting with the ImageNet dataset. The most important metric is emphasized in blue.

	A-All	A-Avg	I	H	V	C	G	B	R
Uniform	31.52	46.59	41.12	**40.98**	67.42	41.60	43.26	41.82	49.96
MOO	31.92	47.19	41.92	41.78	67.64	**42.10**	43.66	42.74	50.48
TA-MOO	**32.00**	**47.21**	**41.94**	41.80	**67.66**	42.06	**43.70**	**42.80**	**50.52**

Table 21: Evaluation of Attacking Ensemble model on the CIFAR10 (C10) and CIFAR100 (C100) datasets. The highest/second highest performance is highlighted in **Bold**/Underline. The table is copied from Table 1 in the main paper for reading comprehension purpose.

		CW		CE		KL	
		A-All	A-Avg	A-All	A-Avg	A-All	A-Avg
C10	Uniform	26.37	**41.13**	28.21	48.34	17.44	32.85
	MinMax	27.53	41.20	35.75	**51.56**	19.97	**33.13**
	MOO	18.87	34.24	25.16	44.76	15.69	29.54
	TA-MOO	**30.65**	40.41	**38.01**	51.10	**20.56**	31.42
C100	Uniform	52.82	**67.39**	55.86	72.62	38.57	**54.88**
	MinMax	54.96	66.92	63.70	75.44	40.67	53.83
	MOO	51.16	65.87	58.17	73.19	39.18	53.44
	TA-MOO	**55.73**	67.02	**64.89**	**75.85**	**41.97**	53.76

Table 22: Attacking Ensemble model with a diverse set D={R-ResNet18, V-VGG16, G-GoogLeNet, E-EfficientNet} and non-diverse set ND={4 ResNets}. w represents the final w of MOO (mean ± std). $\|\nabla_\delta f_i(\delta)\|$ represents the gradient norm of each model (mean ± std). The table is copied from Table 2 in the main paper for reading comprehension purpose.

		A-All	A-Avg	R/R1	V/R2	G/R3	E/R4
D	$\|\nabla_\delta f_i(\delta)\|$	-	-	7.15 ± 6.87	4.29 ± 4.64	7.35 ± 7.21	0.98 ± 0.72
	w	-	-	0.15 ± 0.14	0.17 ±0.13	0.15 ± 0.14	0.53 ± 0.29
	Uniform	28.21	48.34	48.89	49.08	48.38	47.03
	MOO	25.16	44.76	39.06	46.83	37.05	56.11
	TA-MOO	38.01	51.10	49.55	52.15	49.29	53.40
ND	$\|\nabla_\delta f_i(\delta)\|$	-	-	8.41 ± 8.22	6.68± 6.95	7.36 ± 6.03	5.67 ± 6.09
	w	-	-	0.23 ± 0.21	0.24±0.17	0.23 ± 0.19	0.30 ± 0.21
	Uniform	28.17	48.75	51.94	45.55	54.15	43.34
	MOO	32.50	52.21	53.25	49.05	56.80	49.76
	TA-MOO	41.01	57.33	58.88	55.32	60.81	54.29

E Additional Discussions

E.1 When does MOO Work?

The dominating issue. On one hand, there is the dominating issue that happens in all the three settings. The issue can be recognized by the gap of attack performance among tasks. For example, in Table 22 (i.e., the ENS setting with the diverse ensemble and MOO method), the gap between highest ASR (at EfficientNet) and lowest ASR (at GoogLeNet) is 19%. In the EoT setting, the problem is even worse: The largest gap observed is 53.6% as shown in Table 15 (the highest ASR is 88.19% with Vertical flipping and the lowest ASR is 34.54% with Horizontal flipping in with MOO - D-C10 setting). The dominating issue is also be recognized by the observation that a significant small gradient strength of one task on comparison with other tasks' strength. For example, in Table 22 it can be seen that the gradient strength corresponding to the EfficientNet architecture (mean value is 0.98) is much lower than those of other architectures (mean values are at least 4.29). As the result, the weight corresponding to the EfficientNet architecture is much higher than those of others.

The root of the dominating issue can be the natural of the setting (i.e., as shown in Table 15 with the EoT setting, when the domination of the Vertical flipping task can be observed in all methods) or because of the MOO solver which is discussed in Section E.3

Overcoming the dominating issue. On the other hand, if overcoming this issue, MOO can outperform the Uniform strategy. For example, on attacking the non-diverse ensemble model (i.e., 4 ResNets) MOO surpasses the Uniform strategy by 4.3% and 3.5% in the ASR-All and ASR-Avg metrics, respectively. On generating universal perturbations, MOO outperforms the Uniform strategy in most of the settings. As discussed in Section D.4, a simple memory caching trick can helps to overcome the infinite gradient issue and significantly boosts the performance of MOO or TA-MOO. Therefore, we believe that developing a technique to lessen the dominating issue might be a potential extension to further improve the performance.

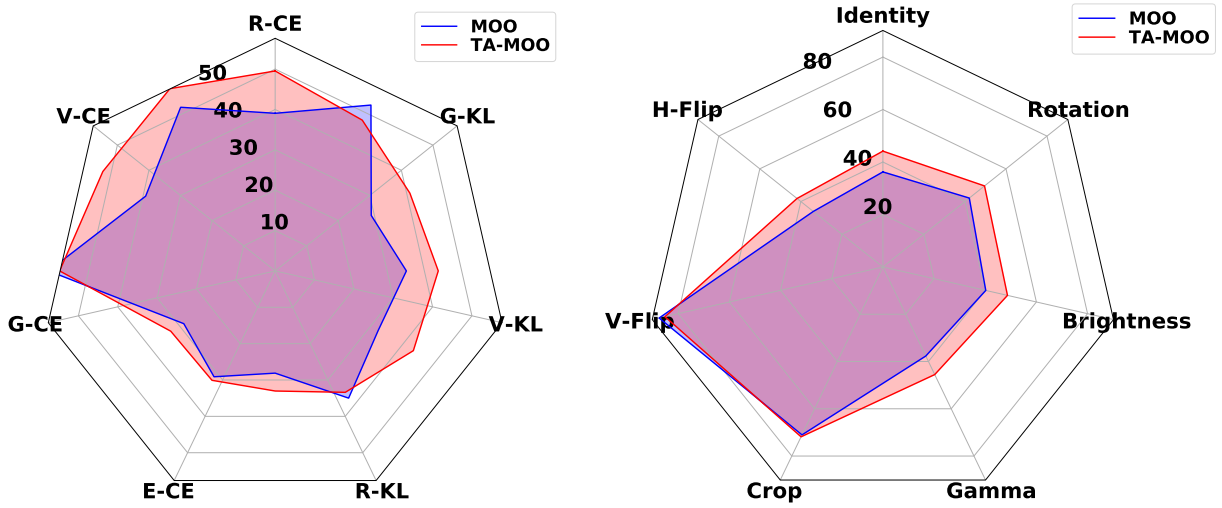
Balancing among goal-unachived tasks. We observed in the EoT setting, the dominating issue is strictly serious when gradients of some tasks are much weaker/stronger than others. It is because of the natural of the transformation operations, therefore, this issue happens regardless status of the tasks. In the set of goal-unachieved tasks' gradients can exist a dominated one, resulting to a much higher weight of the dominated task. Therefore, in order to strike a more balance among goal-unachieved tasks, we apply an additional regularization which minimizes the entropy of goal-unachieved weights $\mathcal{H}(w) = \sum_{i=s+1}^{m} -w_i \log w_i$. If all tasks have been achieved (i.e., $s = m$) then the additional regularization will be ignored. This additional regularization helps to improve further 2% in the EoT setting.

E.2 Importance of the Task-Oriented Regularization.

In this discussion, we would like to provide more experimental results in the ENS and EoT settings to further emphasize the contribution of the Task-Oriented regularization. Figure 5 shows the ASR of each individual task in the ENS setting with three losses and the EoT setting with ResNet18 architecture and deterministic transformations. As shown in Figure 5a, in the ENS setting, the MOO adversary produces a much higher ASR on the EfficientNet architecture than other architectures with any losses. In contrast, the TA-MOO adversary has a lower ASR on the EfficientNet architecture but a much higher ASR on other architectures. Similar observation can be seen in Figure 5b such that the ASR corresponding to the V-flipping of MOO is slightly higher than that of TA-MOO, however, the ASR on other transformations of MOO is much lower than those of TA-MOO.

E.3 More Efficient MOO Solvers

Discussions on the weighted-sum method. One of the most common approaches to solve the MOO problem is the scalarizing method, which formulates a single-objective optimization (SOO) such that the optimal solutions to the SOO problem are Pareto optimal solutions to the MOO problem. While this line of approach (e.g., weighted-sum method) is suitable for end-to-end learning such as deep learning, there are



(a) ENS setting (b) EoT setting

Figure 5: Comparison on the ASR of each individual task. R: ResNet18, V: VGG16, G: GoogLeNet, E: EfficientNet. CE: Cross-entropy loss, KL: Kullback-Leibler divergence, CW: Carnili-Wagner loss

several acknowledged weaknesses: (i) the choice of utility function has a large impact on the computational complexity of the resulted SOO problem (Bjornson et al., 2014; Björnson & Jorswieck, 2013); (ii) a small change in weights may results in big changes in the combined objective (Caballero et al., 1997), and vice versa, a huge different weights may produce nearly similar result (Coello Coello, 1999); (iii) it does not work well in the case of a non-convex objective space (Deb, 2011).

One of the most common replacement for the weighted-sum method is the ϵ constraint method which is applicable to either convex or non-convex problem. Applying a more efficient MOO solver might be one of the potential extensions of this work.

Discussions on the gradient descent solver. Inspired by Sener & Koltun (2018), in this paper we use multi-gradient descent algorithm (Deb, 2011) as an MOO solver which casts the multi-objective problem to a single-objective problem. While Sener & Koltun (2018) used Frank-Wolfe algorithm to project the weight into the desired simplex, we use parameterization with softmax instead. Although this technique is much faster than Frank-Wolfe algorithm, it has some weaknesses that will be addressed in our future work. More specifically, the GD solver with softmax parameterization cannot handle well the edge case which is the root of the dominating issue. The snippet code E.3 provides a minimal example of quadratic optimization problem as similar in MGDA, where the goal is to find $w^* = \underset{w \in \Delta_w}{\operatorname{argmin}} \sum_{i=1}^{5} \|w_i \mathrm{g}_i\|_2^2$. The solver is the Gradient Solver with softmax parameterization. With input_1 where none of elements dominates others, the solver works quite reasonable with the weights corresponding to 4 first elements are equal and less than the last one (corresponding to bigger strength). With input_2 where $g_5 \gg g_1$, the solver still works well where $w_1 = 1$ corresponding to the minimal strength $g_1 = 0.1$. However, with input_3, the solver fails to find a good solution (which should be $w = [1, 0, 0, 0]$ given that input). It is a worth noting that the main goal of this paper is to show the application of Multi-objective Optimization for generating adversarial examples and the impact of the Task-Oriented regularization. Therefore, while the issue of the gradient descent solver is well recognized, we did not take effort to try with a better solver.

```python
import torch
import torch.nn.functional as F
import torch.optim as optim

input_1 = [0.1, 0.1, 0.1, 0.1, 0.2] # normal case
input_2 = [0.01, 0.1, 0.1, 0.1, 2e3] # normal case
```

```python
input_3 = [0.001, 0.002, 0.002, 0.002, 2e3] # dominating issue

init_alpha = [0.2, 0.2, 0.2, 0.2, 0.2]
g = torch.tensor(input_3)
alpha = torch.tensor(init_alpha, requires_grad=True)
opt = optim.SGD([alpha], lr=1.0)

for step in range(20):
    w = F.softmax(alpha, dim=0)
    loss = torch.square(torch.sum(w * g))
    opt.zero_grad()
    loss.backward()
    opt.step()
    print('step={}, w={}'.format(step, w.detach().numpy()))

# Result with input_1
# step=19, w=[0.20344244 0.20344244 0.20344244 0.20344244 0.18623024]
# Result with input_2
# step=19, w=[9.999982e-01 5.582609e-07 5.582609e-07 5.582609e-07 0.]
# Result with input_3
# step=19, w=[0.28042343 0.23985887 0.23985887 0.23985887 0.]
```

Listing 1: Python example of the Gradient Solver with softmax parameterization

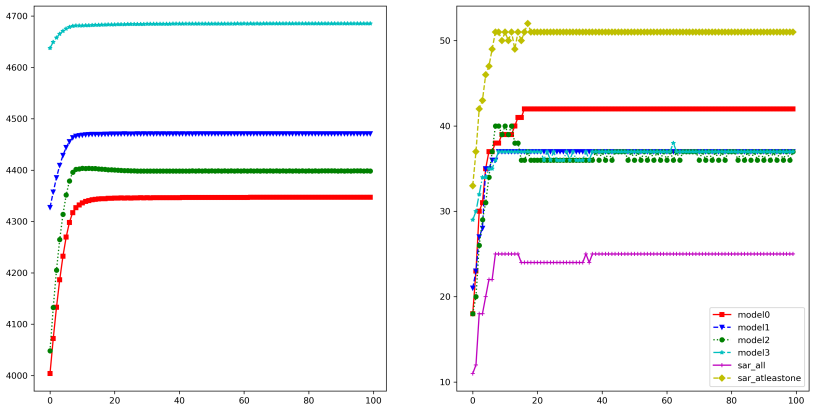
E.4 Correlation between the Objective Loss and Attack Performance.

It is broadly accepted that to fool a model, a feasible approach is maximizing the objective loss (i.e., CE, KL, or CW loss), and the higher the loss, the higher the attack success rate. While it is true with the same architecture, we found that it does not hold when comparing different architectures. Figure 6 shows the adversarial loss and the attack success rate for each model in the ENS setting. With the CW loss as the adversarial objective, it can be observed that there is a positive correlation between the loss value and the ASR, i.e., the higher the loss, the higher the ASR. For example, with the same adversarial examples, the adversarial loss on EfficientNet is the highest and so is ASR. However, there is no clear correlation observed when using CE and KL losses. Therefore, the higher weighted loss does not directly imply a higher success rate for attacking an ensemble of different architectures. The MinMax method (Wang et al., 2021) which solely weighs the tasks' losses, therefore, does not always achieve a good performance in all the tasks.

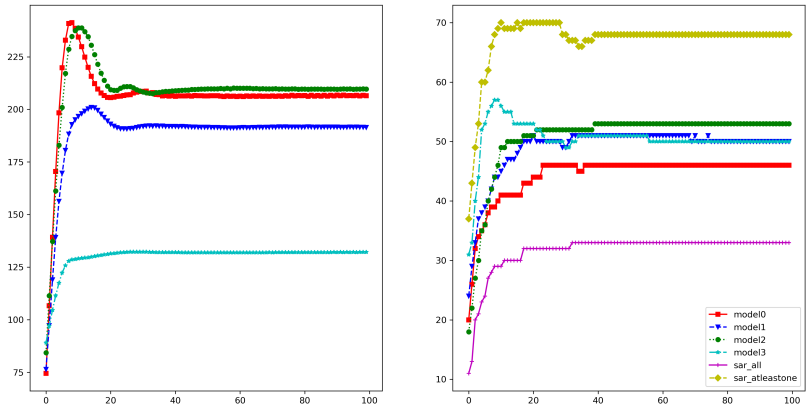
E.5 Conflicting between gradients in the adversarial generation task

In multi-task learning setting, conflicting between gradient is the common issue to tackle with. More specifically, the gradients with respect to the (shared) model parameter of task f_i and task f_j can have a negative correlation (i.e., cosine similarity between $\nabla_\theta f_i(\theta, \delta)$ and $\nabla_\theta f_j(\theta, \delta)$ is negative). However, in the adversarial generation task, we consider the gradient with respect to the input (e.g., $\nabla_\delta f(\theta, \delta)$) to update the adversarial examples. As we explore through empirical experiments, the issue that we to deal with is not the gradient confliction problem but the gradient domination problem. These gradients with respect to the inputs can have a positive correlation but also have a huge difference in their strengths. In this specific challenge, the standard MOO which solely relies on the gradient strengths to calculated the weight for each task is strongly sensitive to the gradient domination problem and in some cases cannot lead to a good solution as discussed in Appendix E.1

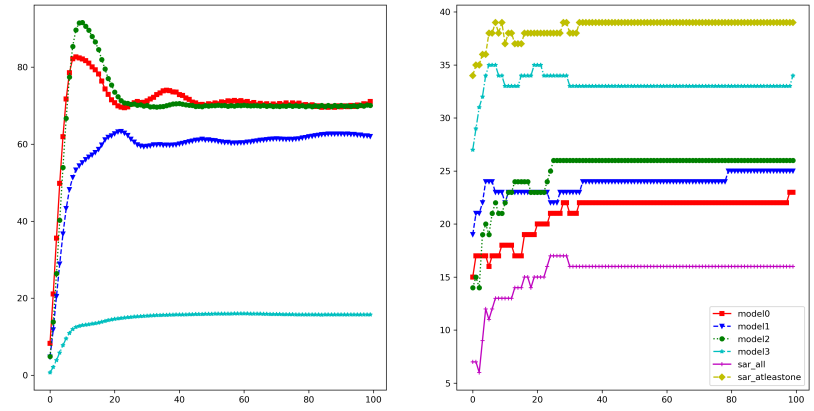
To further support our hypothesis, we would like to provide a measurement on the cosine similarity between gradients on different ensemble members on the ENS setting in Table 23. Each cell (row-ith, column-jth) of the Table reports the cosine similarity between gradient $\nabla_\delta f_i(\delta)$ of model ith and gradient $\nabla_\delta f_j(\delta)$ of model jth (w.r.t. the same input δ). It can be seen that the gradients between different architectures has the positive correlation instead of negative correlation. On the other hand, as shown in the last row, the gradient norm $\|\nabla_\delta f_i(\delta)\|$ varies widely among architectures. While this observation is in line with the widely accepted phenomenon about the transferability of adversarial examples, it also does support our motivation to derive the TA-MOO method to improve the standard MOO.



(a) CW



(b) CE



(c) KL

Figure 6: Loss (left fig) and ASR (right fig) of each task over all attack iterations with the MinMax method. model0/1/2/3 represents R/V/G/E architecture, respectively.

Table 23: Correlation between gradients of ensemble members on ENS setting. Each cell (row-ith, column-jth) reports the cosine similarity (mean ± std) between gradient $\nabla_\delta f_i(\delta)$ of model ith and gradient $\nabla_\delta f_j(\delta)$ of model jth (w.r.t. the same input δ). The last row $\|\nabla_\delta f_i(\delta)\|$ reports the gradient norm of each model. R: ResNet18, V: VGG16, E: EfficientNet, G: G-GoogLeNet.

	R	V	G	E
R	1.00±0.00	0.34±0.15	0.44±0.17	0.35±0.19
V	0.34±0.15	1.00±0.00	0.36±0.19	0.41±0.22
G	0.44±0.17	0.36±0.19	1.00±0.00	0.41±0.18
E	0.35±0.19	0.41±0.22	0.41±0.18	1.00±0.00
$\|\nabla_\delta f_i(\delta)\|$	7.15 ± 6.87	4.29 ± 4.64	7.35 ± 7.21	0.98 ± 0.72

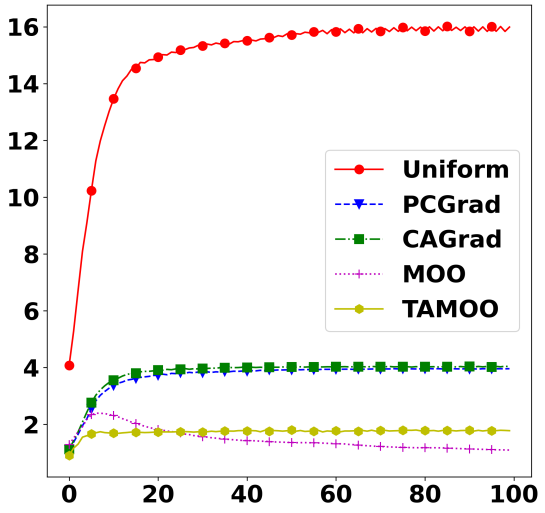
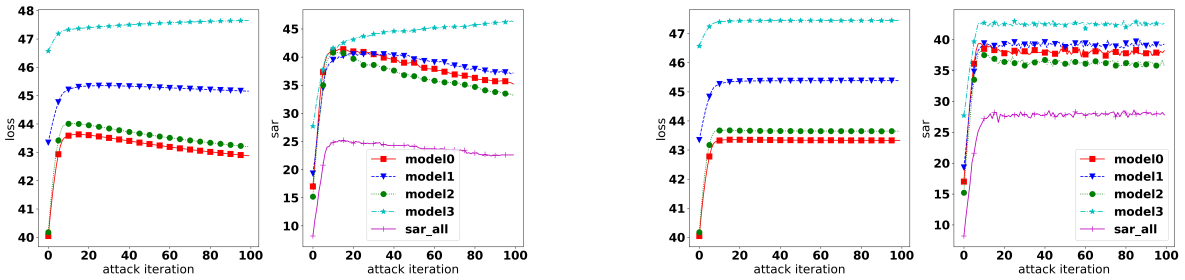


Figure 7: Norm of the gradient $\nabla_\delta f(\delta)$ over all attack iterations. Measure on the diverse set of the ENS setting, with CE loss.

E.6 Discussion on the Convergence of our methods

In multi-task learning, the gradient of each task is calculated with respect to the (shared) model parameter (e.g., $\nabla_\theta f(\theta, \delta)$). Therefore, to quantify the convergence of a multi-task learning method, we can measure the gradient norm of the comment gradient direction to quantify the convergence of the model. The gradient norm is expected to be a very small value when the model reaches to the Pareto optimality points. However, in adversarial generation problem, the gradient of each task is calculated with respect to the input (e.g., $\nabla_\delta f(\theta, \delta)$). Therefore, unlike in the multi-task learning, there is a different behavior of gradient in the adversarial generation task. To verify our hypothesis, we measure the gradient norm of the gradient over all attack iterations and visualize in Figure 7. It can be seen that the gradient norm of all attacks tends to converge to a large value. It is a worth noting that we use projected gradient descent with l_∞ in all attacks. Therefore, in each attack iteration, the amount to update is not the gradient $\nabla_\delta f(\theta, \delta)$ but the sign of it scaling with a step size η_δ. However, there is still an interesting observation such that MOO and TA-MOO attack have a much lower gradient norm than other attacks.

We would like to propose a simple alternative approach to quantify the convergence of our method in the adversarial generation setting. More specifically, we leverage the advantage of the adversarial generation task



(a) MOO (b) TA-MOO

Figure 8: Loss (left fig) and SAR (right fig) of each task over all attack iterations. model0/1/2/3 represents R/V/G/E architecture, respectively. The CW loss is used as the adversaries's objective function.

such that we can access to the label to audit whether the task is successful or not. Therefore, we simply measure the loss and the success attack rate over all attack iterations as shown in Figure 8.

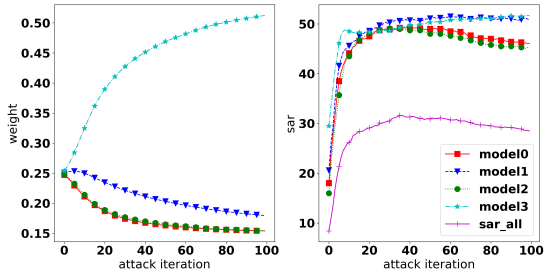
First, we would like to recall the definition of the Pareto optimality. Given m objective function $f(\delta) \triangleq [f_1(\delta), ..., f_m(\delta)]$, the Pareto optimality δ^* of the multi-objective optimization $\delta^* = \underset{\delta}{\text{argmax}} f(\delta)$ if there is no feasible solution δ' such that is strictly better than δ^* in some tasks (i.e., $f_i(\delta') > f_i(\delta^*)$ for some i) while equally good as δ^* in all other tasks (i.e., $f_j(\delta') = f_j(\delta^*), j \neq i$). Bear this definition in mind, it can be seen from the loss progress of MOO attack in Figure 8a that (i) from iteration 1st to around iteration 10th all the losses are increased quickly showing that the method optimize efficiently; (ii) after iteration 10th, the loss w.r.t. the EfficientNet model (i.e., model3 in the legend) continually increases while other losses continually decrease. Therefore, any solution after iteration 10th do not dominate each other indicating that the method reaches the Pareto front.

On the other hand, it can be seen from Figure 8b that the loss progress of our TA-MOO is more stable. TA-MOO also can optimize to the optimal point efficiently as MOO does, however, after reaching the peak, the losses in all tasks are more stable than those in MOO. This observation indicates that the solutions after the peak point are also in the Pareto front but are more concentrated than those in MOO. It can explain the stability of the success attack rate in TA-MOO in Figure 8b. Comparing across both MOO and TA-MOO at their last iteration shows that while the loss w.r.t. the EfficientNet model (model3) in MOO is a bit higher than that in TA-MOO, these other losses w.r.t. V/G/E models in MOO is lower than those in TA-MOO. This observation indicates that in term of losses, the solutions of MOO and TA-MOO do not dominate each other. However, the solution of TA-MOO is more stable and leads better final attacking performance.

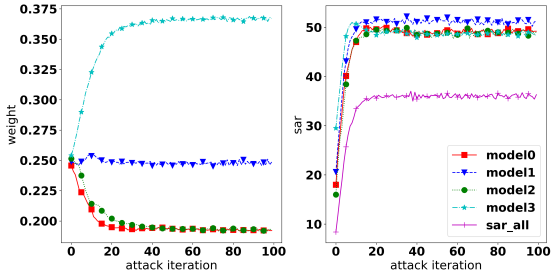
E.7 Additional Experiments with Different initializations for MOO

In our method, the default initialization for the weight w is $1/m$ equally for all tasks. Therefore, one raising valid concern is that *Might better initialization can help to boost the performance?*. To answer this question, we first find the optimal initital weight by using the weight at the last iteration when running MOO and TA-MOO attacks with the default initialization. For example, as shown in Figure 9a for the ENS setting with diverse architectures, the average weight that MOO assigns for model R/V/G/E converging to 0.15/0.17/0.15/0.53 (*set A*), respectively. The average weights' distribution learned by TA-MOO is 0.19/0.25/0.19/0.37 (*set B*), respectively. It is a worth noting that, we consider each set of weights for each data sample separately, and the above weights are just the average over entire testing set (e.g., 10K sample), while the full statistic (mean $\pm$ std) of weights can be seen in Table 2. In order to make the experiment to be more comprehensive with diverse initializations, we use two additional sets including set C=[0.22, 0.23, 0.22, 0.33] and set D=[0.24, 0.25, 0.24, 0.27].

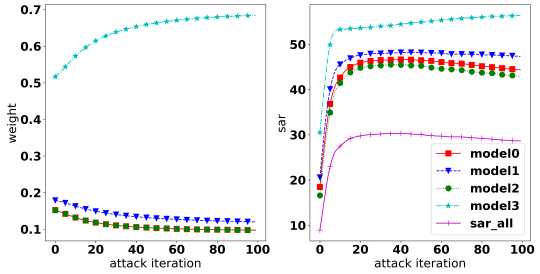
Given these above four weights sets A/B/C/D, we then init the standard MOO with one of these above sets and adjust the learning rate η_w with three options 5e-3, 5e-5, 1e-8 and report results in Table 24. The



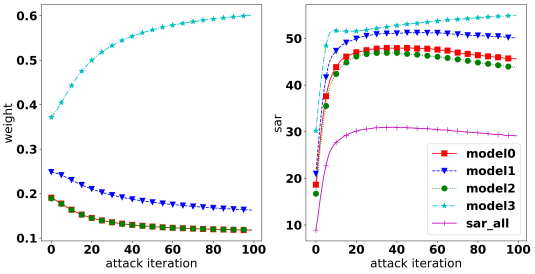
(a) MOO



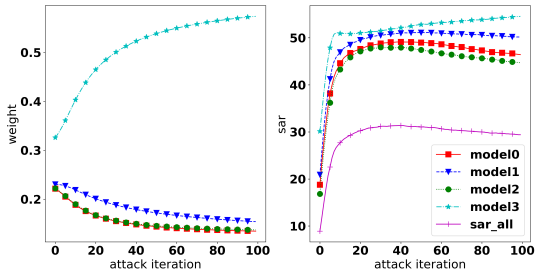
(b) TA-MOO



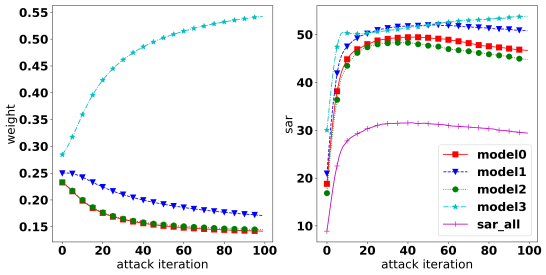
(c) MOO^A with η_w = 5e-3



(d) MOO^B with η_w = 5e-3



(e) MOO^C with η_w = 5e-3



(f) MOO^D with η_w = 5e-3

Figure 9: Weight (left fig) and SAR (right fig) of each task over all attack iterations. model0/1/2/3 represents R/V/G/E architecture, respectively.

complete attacking progress can be seen in Figure 9. It can be seen from Table 24 that better initialization does help to improve the performance of the standard MOO. The best setting is the initialization with set D and η_w = 5e-3 achieves 29.53% in A-All metric, a 4.37% improvement over the default MOO initialization. It can be seen from the evolution of the weights in Figure 9c that even initializing with the converged weights (i.e., set A) from the pre-running attack, the weight of each task does not stand still but converges to a different value. It is another different behavior in adversarial generation task compared to the multi-task learning problem. On the other hand, despite of the extensive tuning, the performance of MOO is still far below the TA-MOO approach, with the gap of 8.48% in A-All metric.

Table 24: Attacking Ensemble model with a diverse set D={R-ResNet18, V-VGG16, G-GoogLeNet, E-EfficientNet}. $\text{MOO}^{A/B/C/D}$ is MOO with initial weights from set A/B/C/D, respectively. η_w denotes the learning rate to update for the weight w.

	η_w = 5e-3	η_w = 5e-5	η_w = 1e-8
MOO^A	28.64	29.18	29.12
MOO^B	29.13	28.75	28.65
MOO^C	29.38	28.46	28.33
MOO^D	29.53	28.37	28.18
MOO	25.16	-	-
TA-MOO	38.01	-	-

4.5 Concluding Remarks

In this chapter, we have presented our contributions towards improving adversarial robustness through the lens of ensemble learning, as outlined in Bui et al. (2021b) and Bui et al. (2023). To this end, we've uncovered two key principles that play pivotal roles in fortifying the robustness of ensemble models.

Firstly, we have emphasized the critical significance of transferability among adversarial examples across ensemble members. By increasing the diversity among ensemble members, we can reduce the transferability of adversarial examples between them, thereby improving the overall robustness of the ensemble model. The main idea behind our approach was to prioritize the correct predictions of one model on a given adversarial example while discouraging other models from making unanimous predictions.

Secondly, we observed that adversarial examples that fool all ensemble members are crucial for improving the robustness of ensemble models. Building upon this observation, we extended our investigations in Bui et al. (2023) by presenting a novel method for generating transferable adversarial examples that lie in the joint insecure region of all ensemble members. These adversarial examples are particularly beneficial for improving the robustness of ensemble models, as they are capable of fooling all ensemble members simultaneously.

In summary, our chapter has not only elucidated fundamental principles for strengthening ensemble model robustness but has also contributed to the field by introducing a novel approach for generating transferable adversarial examples. These findings collectively advance our understanding of the intricate dynamics of ensemble learning in the context of adversarial challenges.

Chapter 5

Distributional Approaches to Adversarial Robustness

5.1 Introduction

In the previous chapter, we demonstrated that incorporating the global and local information of the data manifold in learning robust representations can greatly enhance a model's adversarial robustness. However, these methods, along with other AT-based methods, often seek a pointwise adversary by independently perturbing each data sample.

Considering adversarial effects at a distributional level, on the other hand, may offer unexplored benefits. Unlike AT, distributional robustness seeks a worst-case distribution that generates adversarial examples from a known uncertainty set of distributions located in the ball centered around the data distribution. This approach is expected to have better generalization performance on unseen data.

In this chapter, we present our contributions towards improving adversarial robustness through the lens of distributional robustness, as introduced in Bui et al. (2022). In particular, we proposed a unified framework that connects Wasserstein distributional robustness with current state-of-the-art AT methods. We introduced a new cost function of the Wasserstein distance and propose a unified formulation of the risk function in WDR, with which, we could generalize and encompass the existing AT methods including

SOTA ones in the distribution robustness setting. Through extensive experiments, we demonstrated that with a better generalization capacity of distributional robustness, the resulting AT methods in our framework could achieve better adversarial robustness than their standard AT counterparts.

The major content of this chapter is in the following attached paper:

- **Anh Bui**, Trung Le, Quan Tran, He Zhao, Dinh Phung, *"A Unified Wasserstein Distributional Robustness Framework for Adversarial Training".* In Proceedings of the International Conference on Learning Representation (ICLR) 2022.

  The code of this paper is released at https://github.com/tuananhbui89/Unified-Distributional-Robustness.

5.2 Related Work

Beyond Point-wise Adversarial Training. Since its proposal in Szegedy et al. (2014), adversarial training has undergone extensive study and achieved remarkable success in enhancing adversarial robustness. However, most existing adversarial training methods employ a point-wise adversary, independently perturbing each data sample. This approach disregards the global structure of the data manifold and the classifier's decision boundary concerning other data samples. Consequently, it may result in overfitting issues, where a model becomes robust against one specific adversary but remains vulnerable to others (Rice et al., 2020, Yu et al., 2022).

Several recent works have aimed to go beyond point-wise AT and can be categorized into two primary directions: (1) modeling the global/local structure of the data manifold as a distribution and (2) heuristically incorporating the global/local information of the data manifold into the AT process. It is worth noting that the former direction closely relates to distributional robustness, while the latter is more closely associated with manifold regularization (Mao et al., 2019, Bui et al., 2020, Jin and Rinard, 2020) or geometry-aware regularization (Zhang et al., 2021, Zhu et al., 2022).

Moreover, local information of the data manifold refers to the information concerning the data samples in the vicinity of the current sample, which can be represented as the local distribution of the data manifold. On the other hand, global information of

the data manifold refers to information beyond the local neighborhood of the current sample, such as the distribution of the entire data manifold or the classifier's decision boundary.

Modeling the Local Distribution. Dong et al. (2020) proposed to construct a distribution over each data sample to model the adversarial examples around it. The optimization problem is formulated as a min-max problem over the distribution of the adversarial examples.

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{p(\delta)\in\mathcal{A}(x)} \mathbb{E}_{\delta\sim p(\delta)} \left[\ell(\theta, x+\delta, y) \right] \right]$$

Here, $\mathcal{A}(x)$ represents a set of distributions over the perturbation δ applied to the data sample x. The objective of the inner maximization is to learn an adversarial distribution that enables a point-wise adversary to generate effective adversarial examples. On the other hand, the goal of the outer minimization is to learn a model that remains robust against the adversarial distribution, even in the presence of the worst-case point-wise adversary. However, it is worth noting that the optimal solution of the inner maximization tends to degenerate into the point-wise adversary, which is not the desired solution.

$$\max_{p(\delta)\in\mathcal{A}(x)} \mathbb{E}_{\delta\sim p(\delta)} \left[\ell(\theta, x+\delta, y) \right] \leq \max_{\delta\in\mathcal{S}} \ell(\theta, x+\delta, y)$$

To mitigate this problem, the authors proposed an additional entropic regularization term to increase the support of the adversarial distribution.

Thanh et al. (2022) on the other hand, proposed a particle-based AT algorithm, which models the local distribution of each data sample as a set of particles. To diversify the particle set, the authors utilized the Stein Variational Gradient Descent (SVGD) (Liu and Wang, 2016) to update the particles which encourages the particles move away from each other.

Modeling the Global Distribution. Early works such as Sinha et al. (2017) and Staib and Jegelka (2017) were pioneers in establishing a connection between distribution

robustness and adversarial training. They utilized the dual form of Wasserstein distributional robustness (Blanchet and Murthy, 2019) to identify the worst-case perturbations for adversarial training. Unlike previous approaches, Sinha et al. (2017) aimed to find the worst adversarial distribution over the entire data distribution. They formulated the optimization problem as a min-max problem over the distribution of the data samples, which can be expressed as follows:

$$\min_{\tilde{\mathbb{Q}}:\mathcal{W}_c\left(\tilde{\mathbb{Q}},\mathbb{Q}\right)<\epsilon} \mathbb{E}_{\tilde{\mathbb{Q}}}\left[f\left(z\right)\right],$$

where $\epsilon > 0$ and $\mathcal{W}_c$ denotes the optimal transport (OT) cost, or a Wasserstein distance if c is a metric, defined as:

$$\mathcal{W}_c\left(\tilde{\mathbb{Q}},\mathbb{Q}\right) := \inf_{\gamma\in\Gamma\left(\tilde{\mathbb{Q}},\mathbb{Q}\right)} \int c d\gamma,$$

where $\Gamma\left(\tilde{\mathbb{Q}},\mathbb{Q}\right)$ is the set of couplings whose marginals are $\tilde{\mathbb{Q}}$ and $\mathbb{Q}$.

To solve the above problem, Sinha et al. (2017) proposed a dual form of Wasserstein DR (Blanchet and Murthy, 2019), which has been discussed in Section 2.7.

In contrast, Phan et al. (2023) proposed an alternative approach that utilizes a specific Wasserstein (WS) distance, allowing for a closed-form solution to the primal problem without relying on the dual form used in Sinha et al. (2017). The proposed WS distance takes into account the transportation cost between a batch of data samples and their corresponding adversaries. However, it should be noted that their method operates at the batch level and does not fully capture the global distribution of the data manifold.

Another noteworthy contribution is the approach presented by Le et al. (2022), which employs a Generative Adversarial Network (GAN) framework. In this approach, the feature extractor serves as the generator, and an additional discriminator is introduced to differentiate between the real feature distribution and the adversarial feature distribution. By leveraging the discriminator's knowledge, stronger adversaries are generated that not only deceive the classifier but also outwit the discriminator. Due to the fact that the discriminator is trained on the entire distribution of the data manifold, this method can be considered a global distributional robustness technique.

5.3 Unified Wasserstein Distributional Robustness

A UNIFIED WASSERSTEIN DISTRIBUTIONAL ROBUSTNESS FRAMEWORK FOR ADVERSARIAL TRAINING

Tuan Anh Bui[1], Trung Le[1], Quan Hung Tran[2], He Zhao[1], and Dinh Phung[1, 3]

[1]Monash University [2]Adobe Research [3]VinAI Research

ABSTRACT

It is well-known that deep neural networks (DNNs) are susceptible to adversarial attacks, exposing a severe fragility of deep learning systems. As the result, adversarial training (AT) method, by incorporating adversarial examples during training, represents a natural and effective approach to strengthen the robustness of a DNN-based classifier. However, most AT-based methods, notably PGD-AT and TRADES, typically seek a pointwise adversary that generates the worst-case adversarial example by independently perturbing each data sample, as a way to "probe" the vulnerability of the classifier. Arguably, there are unexplored benefits in considering such adversarial effects from an entire distribution. To this end, this paper presents a unified framework that connects Wasserstein distributional robustness with current state-of-the-art AT methods. We introduce a new Wasserstein cost function and a new series of risk functions, with which we show that standard AT methods are special cases of their counterparts in our framework. This connection leads to an intuitive relaxation and generalization of existing AT methods and facilitates the development of a new family of distributional robustness AT-based algorithms. Extensive experiments show that our distributional robustness AT algorithms robustify further their standard AT counterparts in various settings.[1]

1 INTRODUCTION

Despite remarkable performances of DNN-based deep learning methods, even the state-of-the-art (SOTA) models are reported to be vulnerable to adversarial attacks (Biggio et al., 2013; Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018; Athalye et al., 2018; Zhao et al., 2019b; 2021a), which is of significant concern given the large number of applications of deep learning in real-world scenarios. Usually, adversarial attacks are generated by adding small perturbations to benign data but to change the predictions of the target model. To enhance the robustness of DNNs, various adversarial defense methods have been developed, recently Pang et al. (2019); Dong et al. (2020); Zhang et al. (2020b); Bai et al. (2020). Among a number of adversarial defenses, Adversarial Training (AT) is one of the most effective and widely-used approaches (Goodfellow et al., 2015; Madry et al., 2018; Shafahi et al., 2019; Tramèr & Boneh, 2019; Zhang & Wang, 2019; Xie et al., 2020). In general, given a classifier, AT can be viewed as a robust optimization process (Ben-Tal et al., 2009) of seeking a pointwise adversary (Staib & Jegelka, 2017) that generates the worst-case adversarial example by independently perturbing each data sample.

Different from AT, Distributional Robustness (DR) (Delage & Ye, 2010; Duchi et al., 2021; Gao et al., 2017; Gao & Kleywegt, 2016; Rahimian & Mehrotra, 2019) looks for a worst-case distribution that generates adversarial examples from a known uncertainty set of distributions located in the ball centered around the data distribution. To measure the distance between distributions, different kinds of metrics have been considered in DR, such as f-divergence (Ben-Tal et al., 2013; Miyato et al., 2015; Namkoong & Duchi, 2016) and Wasserstein distance (Shafieezadeh-Abadeh et al., 2015; Blanchet et al., 2019; Kuhn et al., 2019), where the latter has shown advantages over others on efficiency and simplicity (Staib & Jegelka, 2017; Sinha et al., 2018). Therefore, adversary in DR does not look for the perturbation of a specific data sample, but moves the entire distribution around the data distribution, thus, is expected to have better generalization than AT on unseen data samples

[1]Our code is available at https://github.com/tuananhbui89/Unified-Distributional-Robustness

(Staib & Jegelka, 2017; Sinha et al., 2018). Conceptually and theoretically, DR can be viewed as a generalization and better alternative to AT and several attempts (Staib & Jegelka, 2017; Sinha et al., 2018) have shed light on connecting AT with DR. However, to the best of our knowledge, practical DR approaches that achieve comparable peformance with SOTA AT methods on adversarial robustness have not been developed yet.

To bridge this gap, we propose a unified framework that connects distributional robustness with various SOTA AT methods. Built on top of Wasserstein Distributional Robustness (WDR), we introduce a new cost function of the Wasserstein distances and propose a unified formulation of the risk function in WDR, with which, we can generalize and encompass SOTA AT methods in the DR setting, including PGD-AT (Madry et al., 2018), TRADES (Zhang et al., 2019), MART (Wang et al., 2019) and AWP (Wu et al., 2020). With better generalization capacity of distributional robustness, the resulted AT methods in our DR framework are shown to be able to achieve better adversarial robustness than their standard AT counterparts.

The contributions of this paper are in both theoretical and practical aspects, summarized as follows: **1)** Theoretically, we propose a general framework that bridges distributional robustness and standard robustness achieved by AT. The proposed framework encompasses the DR versions of the SOTA AT methods and we prove that these AT methods are special cases of their DR counterparts. **2)** Practically, motivated by our theoretical study, we develop a novel family of algorithms that generalize the AT methods in the standard robustness setting, which have better generalization capacity. **3)** Empirically, we conduct extensive experiments on benchmark datasets, which show that the proposed AT methods in the distributional robustness setting achieve better performance than standard AT methods.

2 PRELIMINARIES

2.1 DISTRIBUTIONAL ROBUSTNESS

Distributional Robustness (DR) is an emerging framework for learning and decision-making under uncertainty, which seeks the worst-case expected loss among a ball of distributions, containing all distributions that are close to the empirical distribution (Gao et al., 2017). As the Wasserstein distance is a powerful and convenient tool of measuring closeness between distributions, Wasserstein DR has been one of the most widely-used variant of DR, which has rich applications in (semi)-supervised learning (Blanchet & Kang, 2020; Chen & Paschalidis, 2018; Yang, 2020), generative modeling (Huynh et al., 2021; Dam et al., 2019), transfer learning and domain adaptation (Lee & Raginsky, 2018; Duchi et al., 2019; Zhao et al., 2019a; Nguyen et al., 2021a;b; Le et al., 2021b;a), topic modeling (Zhao et al., 2021b), and reinforcement learning (Abdullah et al., 2019; Smirnova et al., 2019; Derman & Mannor, 2020). For more comprehensive review, please refer to the surveys of Kuhn et al. (2019); Rahimian & Mehrotra (2019). Here we consider a generic Polish space S endowed with a distribution $\mathbb{P}$. Let $f : S \to \mathbb{R}$ be a real-valued (risk) function and $c : S \times S \to \mathbb{R}_+$ be a cost function. Distributional robustness setting aims to find the distribution $\mathbb{Q}$ in the vicinity of $\mathbb{P}$ and maximizes the risk in the $\mathbb{E}$ form (Sinha et al., 2018; Blanchet & Murthy, 2019):

$$\sup_{\mathbb{Q}:\mathcal{W}_c(\mathbb{P},\mathbb{Q})<\epsilon} \mathbb{E}_{\mathbb{Q}}\left[f\left(z\right)\right], \tag{1}$$

where $\epsilon > 0$ and $\mathcal{W}_c$ denotes the optimal transport (OT) cost, or a Wasserstein distance if c is a metric, defined as:

$$\mathcal{W}_c\left(\mathbb{P},\mathbb{Q}\right) := \inf_{\gamma\in\Gamma(\mathbb{P},\mathbb{Q})} \int c d\gamma, \tag{2}$$

where $\Gamma\left(\mathbb{P},\mathbb{Q}\right)$ is the set of couplings whose marginals are $\mathbb{P}$ and $\mathbb{Q}$. With the assumption that $f \in L^1\left(\mathbb{P}\right)$ is upper semi-continuous and the cost c is a non-negative lower semi-continuous satisfying $c(z, z') = 0$ iff $z = z'$, Sinha et al. (2018); Blanchet & Murthy (2019) show that the *dual* form for Eq. (1) is:

$$\inf_{\lambda\geq 0}\left\{\lambda\epsilon + \mathop{\mathbb{E}}_{z\sim\mathbb{P}}\left[\sup_{z'}\left\{f\left(z'\right) - \lambda c\left(z', z\right)\right\}\right]\right\}. \tag{3}$$

Sinha et al. (2018) further employs a Lagrangian for Wasserstein-based uncertainty sets to arrive at a relaxed version with $\lambda \geq 0$:

$$\sup_{\mathbb{Q}}\left\{\mathbb{E}_{\mathbb{Q}}\left[f\left(z\right)\right] - \lambda\mathcal{W}_c\left(\mathbb{P},\mathbb{Q}\right)\right\} = \mathop{\mathbb{E}}_{z\sim\mathbb{P}}\left[\sup_{z'}\left\{f\left(z'\right) - \lambda c\left(z', z\right)\right\}\right]. \tag{4}$$

2.2 ADVERSARIAL ROBUSTNESS WITH ADVERSARIAL TRAINING

In this paper, we are interested in image classification tasks and focus on the adversaries that add small perturbations to the pixels of an image to generate attacks based on gradients, which are the most popular and effective. FGSM (Goodfellow et al., 2015) and PGD (Madry et al., 2018) are the most representative gradient-based attacks and PGD is the most widely-used one, due to its effectiveness and simplicity. Now we consider a classification problem on the space $S = \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ is the data space, $\mathcal{Y}$ is the label space. We would like to learn a classifier that predicts the label of a datum well $h_\theta : \mathcal{X} \to \mathcal{Y}$. Learning of the classifier can be done by minimising its loss: $\ell\left(h_\theta\left(x\right), y\right)$, which can typically be the the cross-entropy loss. In addition to predicting well on benign data, an adversarial defense aims to make the classifier robust against adversarial examples. As the most successful approach, adversarial training is a straightforward method that creates and then incorporates adversarial examples into the training process. With this general idea, different AT methods vary in the way of picking which adversarial examples one should train on. Here we list three widely-used AT methods.

PGD-AT (Madry et al., 2018) seeks the *most violating* examples to improve model robustness:

$$\inf_\theta \mathbb{E}_\mathbb{P}\left[\beta \sup_{x' \in B_\epsilon(x)} CE\left(h_\theta\left(x'\right), y\right) + CE\left(h_\theta\left(x\right), y\right)\right], \tag{5}$$

where $B_\epsilon\left(x\right) = \{x' : c_\mathcal{X}\left(x, x'\right) \leq \epsilon\}$, $\beta > 0$ is the trade-off parameter and cross-entropy loss CE.

TRADES (Zhang et al., 2019) seeks the *most divergent* examples to improve model robustness:

$$\inf_\theta \mathbb{E}_\mathbb{P}\left[\beta \sup_{x'} D_{KL}\left(h_\theta\left(x'\right), h_\theta\left(x\right)\right) + CE\left(h_\theta\left(x\right), y\right)\right], \tag{6}$$

where $x' \in B_\epsilon\left(x\right)$ and D_{KL} is the usual Kullback-Leibler (KL) divergence.

MART (Wang et al., 2019) takes into account prediction confidence:

$$\inf_\theta \mathbb{E}_\mathbb{P}\Big[\beta\left(1 - \left[h_\theta\left(x\right)\right]_y\right) \sup_{x' \in B_\epsilon(x)} D_{KL}\left(h_\theta\left(x'\right), h_\theta\left(x\right)\right) + BCE\left(h_\theta\left(x\right), y\right)\Big], \tag{7}$$

where $BCE\left(h_\theta\left(x\right), y\right)$ is defined as: $-\log\left(\left[h_\theta\left(x\right)\right]_y\right) - \log\left(1 - \max_{k \neq y}\left[h_\theta\left(x\right)\right]_k\right)$.

2.3 CONNECTING DISTRIBUTIONAL ROBUSTNESS TO ADVERSARIAL TRAINING

To bridge distributional and adversarial robustness, Sinha et al. (2018) proposes an AT method, named Wasserstein Risk Minimization (WRM), which generalizes PGD-AT through the principled lens of distributionally robust optimization. For smooth loss functions, WRM enjoys convergence guarantees similar to non-robust approaches while certifying performance even for the worst-case population loss. Specifically, assume that $\mathbb{P}$ is a joint distribution that generates a pair $z = (x, y)$ where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The cost function is defined as: $c\left(z, z'\right) = c_\mathcal{X}\left(x, x'\right) + \infty \times \mathbf{1}\left\{y \neq y'\right\}$ where $z' = (x', y')$, $c_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is a cost function on $\mathcal{X}$, and $\mathbf{1}\left\{\cdot\right\}$ is the indicator function. One can define the risk function f as the loss of the classifier, i.e., $f\left(z\right) := \ell\left(h_\theta\left(x\right), y\right)$. Together with Eq. (1), attaining a robust classifier is to solve the following min-max problem:

$$\inf_\theta \sup_{\mathbb{Q}:\mathcal{W}_c(\mathbb{P},\mathbb{Q})<\epsilon} \mathbb{E}_\mathbb{Q}\left[\ell\left(h_\theta\left(x\right), y\right)\right]. \tag{8}$$

The above equation shows the generalisation of WRM to PGD-AT. With Eq. (3) and Eq. (4), one can arrive at Eq. (9) as below where $\lambda \geq 0$ is a trade-off parameter:

$$\inf_\theta \mathbb{E}_\mathbb{P}\left[\sup_{x'}\left\{\ell\left(h_\theta\left(x'\right), y\right) - \lambda c_\mathcal{X}\left(x', x\right)\right\}\right]. \tag{9}$$

3 PROPOSED UNIFIED DISTRIBUTION ROBUSTNESS FRAMEWORK

Although WRM (Sinha et al., 2018) sheds light on connecting distributional robustness with adversarial training, its framework and formulation is limited to PGD-AT, which cannot encompass more

advanced AT methods including TRADES and MART. In this paper, we propose a unified formulation for distributional robustness, which is a more general framework connecting state-of-the-art AT and existing distributional robustness approaches where they become special cases.

Let $\mathbb{P}^d$ be the data distribution that generates instance $x \sim \mathbb{P}^d$ and $\mathbb{P}^l_{\cdot|x}$ the conditional to generate label $y \sim \mathbb{P}^l_{\cdot|x}$ given x where $x \in \mathcal{X}, y \in \mathcal{Y}$. For our purpose, we consider the space $S = \mathcal{X} \times \mathcal{X} \times \mathcal{Y}$ and a joint distribution $\mathbb{P}_\triangle$ on S consisting of samples (x, x, y) where $x \sim \mathbb{P}^d$ and $y \sim \mathbb{P}^l_{\cdot|x}$. Now consider a distribution $\mathbb{Q}$ on S such that $\mathcal{W}_c\left(\mathbb{Q}, \mathbb{P}_\triangle\right) < \epsilon$. A draw $z \sim \mathbb{P}_\triangle$ will take the form $z = (x, x, y)$ whereas $z' \sim \mathbb{Q}$ will be $z' = (x', x'', y')$. We propose cost function $c(z, z')$ defined as:

$$c(z, z') = c_{\mathcal{X}}\left(x, x'\right) + \infty \times c_{\mathcal{X}}\left(x, x''\right) + \infty \times \mathbf{1}\left\{y \neq y'\right\}, \tag{10}$$

where we note that this cost function is non negative, satisfies $c(z, z) = 0$ and lower semi-continuous, i.e., $\lim_{z' \to z_0} \inf c(z, z') \geq c(z, z_0)$.

With our new setting, it is useful to understand the "vicinity"of $\mathbb{P}_\triangle$ via the distribution OT-ball condition $\mathcal{W}_c\left(\mathbb{Q}, \mathbb{P}_\triangle\right) < \epsilon$. Since there exists a transport plan $\gamma \in \Gamma\left(\mathbb{P}_\triangle, \mathbb{Q}\right)$ s.t. $\int c d\gamma < \epsilon$ and $c(z, z')$ is finite a.s. γ, this implies that if $(z, z') \sim \gamma$, then first, it is easy to see that $x'' = x$ and $y' = y$, and second, x' tends to be close to x. To see why the later is the case, since $\mathbb{P}^d$ is a marginal of $\mathbb{P}_\triangle$ on the first x in (x, x, y), therefore if $\mathbb{Q}^d$ is the marginal of $\mathbb{Q}$ on x' in (x', x'', y') then $\mathcal{W}_d\left(\mathbb{Q}^d, \mathbb{P}^d\right) \leq \mathcal{W}_d\left(\mathbb{Q}, \mathbb{P}_\triangle\right) < \epsilon$, which explains the closeness between of x and x'.

Given $z' = (x', x'', y') \sim \mathbb{Q}$ where $\mathcal{W}_c\left(\mathbb{Q}, \mathbb{P}_\triangle\right) < \epsilon$, we define a unified risk function $g_\theta\left(z'\right)$ w.r.t a classifier h_θ that encompasses the unified distributional robustness (UDR) version for PGD-AT, TRADES, and MART (cf Section 2.2):

- *UDR-PGD*: $g_\theta\left(z'\right) := CE\left(h_\theta\left(x''\right), y'\right) + \beta CE\left(h_\theta\left(x'\right), y'\right)$.
- *UDR-TRADES*: $g_\theta\left(z'\right) := CE\left(h_\theta\left(x''\right), y'\right) + \beta D_{KL}\left(h_\theta\left(x'\right), h_\theta\left(x''\right)\right)$.
- *UDR-MART*: $g_\theta\left(z'\right) := BCE\left(h_\theta\left(x''\right), y'\right) + \beta(1 - [h_\theta\left(x''\right)]_y) D_{KL}\left(h_\theta\left(x'\right), h_\theta\left(x''\right)\right)$.[2]

Now we derive the primal and dual objectives for the proposed UDR framework. With the UDR risk function $g_\theta(z')$ defined previously, following Eq. (1) and Eq. (3), the primal (left) and dual (right) forms of our UDR objective are:

$$\inf_\theta \sup_{\mathbb{Q}: \mathcal{W}_c(\mathbb{Q}, \mathbb{P}_\triangle) < \epsilon} \mathbb{E}_{\mathbb{Q}}\left[g_\theta\left(z'\right)\right] = \inf_\theta \inf_{\lambda \geq 0} \left(\lambda\epsilon + \mathbb{E}_{\mathbb{P}_\triangle}\left[\sup_{z'}\left\{g_\theta\left(z'\right) - \lambda c\left(z', z\right)\right\}\right]\right). \tag{11}$$

With the cost function c defined in Eq. (10), the dual form in (11) can be rewritten as:

$$\begin{aligned} &\inf_{\theta, \lambda \geq 0} \left(\lambda\epsilon + \mathbb{E}_{\mathbb{P}_\triangle}\left[\sup_{x', x'' = x, y' = y}\left\{g_\theta\left(z'\right) - \lambda c_{\mathcal{X}}\left(x', x\right)\right\}\right]\right) = \\ &\inf_{\theta, \lambda \geq 0} \left(\lambda\epsilon + \mathbb{E}_{\mathbb{P}}\left[\sup_{x'}\left\{g_\theta\left(x', x, y\right) - \lambda c_{\mathcal{X}}\left(x', x\right)\right\}\right]\right) \end{aligned} \tag{12}$$

where we note that $\mathbb{P}$ is a distribution over pairs (x, y) for which $x \sim \mathbb{P}^d$ and $y \sim \mathbb{P}^l_{\cdot|x}$. The min-max problem in Eq. (12) encompasses the PGD-AT, TRADES, and MART distributional robustness counterparts on the choice of the function $g_\theta\left(x', x, y\right)$ by simply choosing an appropriate $g_\theta\left(x', x, y\right)$ as shown in Section 2.3.

In what follows, we prove that standard PGD-AT, TRADES, and MART presented in Section 2 are specific cases of their UDR counterparts by specifying corresponding cost functions. Given a cost function $c_{\mathcal{X}}$ (e.g., L_1, L_2, and L_∞), we define a new cost function $\tilde{c}_{\mathcal{X}}$ as:

$$\tilde{c}_{\mathcal{X}}\left(x, x'\right) = \begin{cases} c_{\mathcal{X}}\left(x, x'\right) & \text{if } c_{\mathcal{X}}(x, x') \leq \epsilon \\ \infty & \text{otherwise.} \end{cases} \tag{13}$$

The cost function $\tilde{c}_{\mathcal{X}}$ is lower semi-continuous. By defining the ball $B_\epsilon\left(x\right) := \left\{x' : c_{\mathcal{X}}\left(x, x'\right) \leq \epsilon\right\} = \left\{x' : \tilde{c}_{\mathcal{X}}\left(x, x'\right) \leq \epsilon\right\}$, we achieve the following theorem on the relation between distributional and standard robustness.

[2]To encompass MART with our framework, we assume a classifier is adversarially trained by Eq. (7) with adversarial examples generated by $\sup_{x' \in B_\epsilon(x)} D_{KL}\left(h_\theta\left(x'\right), h_\theta\left(x\right)\right) + BCE\left(h_\theta\left(x\right), y\right)$. This is slightly different from the original MART, where the adversarial examples are generated by $\sup_{x' \in B_\epsilon(x)} CE\left(h_\theta\left(x'\right), y\right)$.

Theorem 1. *With the cost function $\tilde{c}_x$ defined as above, the optimization problem:*

$$\inf_{\theta,\lambda\geq 0}\left(\lambda\epsilon+\mathbb{E}_{\mathbb{P}}\left[\sup_{x'}\left\{g_\theta\left(x',x,y\right)-\lambda\tilde{c}_{\mathcal{X}}\left(x',x\right)\right\}\right]\right) \tag{14}$$

is equivalent to the optimization problem:

$$\inf_{\theta}\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\epsilon(x)} g_\theta\left(x',x,y\right)\right]. \tag{15}$$

Proof. See Appendix A for the proof. □

Theoretical contribution and comparison to previous work. Theorem 1 says that the standard PGD-AT, TRADES, and MART are special cases of their UDR counterparts, which indicates that our UDR versions of AT have a richer expressiveness capacity than the standard ones. Different from WRM (Sinha et al., 2018) , our proposed framework is developed based on theoretical foundation of (Blanchet & Murthy, 2019). It is worth noting that the theoretical development is *not trivial* because theory developed in Blanchet & Murthy (2019) is only valid for a bounded cost function, while the cost function $\tilde{c}$ is unbounded. More specifically, the transformation from primal to dual forms in Eq. (11) requires the cost function c to be bounded. In Theorem 2 in Appendix A, we prove this primal-dual form transformation for the unbounded cost function $\tilde{c}_{\mathcal{X}}$, which is certainly not trivial.

Moreover, our UDR is *fundamentally distinctive* from WRM in its ability to adapt and learn λ, while this is a hyper-parameter in WRM. As a result of a fixed λ, WRM is fundamentally same as PGD in the sense that these methods can only utilize local information of relevant benign examples when crafting adversarial examples. In contrast, our UDR can leverage both local and global information of multiple benign examples when crafting adversarial examples due to the fact that λ is adaptable and captures the global information when solving the outer minimization in (14). Further explanation can be found in Appendix B.

4 Learning Robust Models with UDR

In this section we introduce the details of how to learn robust models with UDR. To do this, we first discuss the induced cost function $\tilde{c}_{\mathcal{X}}$ defined as in Eq (13), which assists us in understanding the connection between distributional and standard robustness approaches. We note that $\tilde{c}_{\mathcal{X}}$ is non-differential outside the perturbation ball (i.e., $c_{\mathcal{X}}(x',x)\geq\epsilon$). To circumvent this, we introduce a smoothed version $\hat{c}_{\mathcal{X}}$ to approximate $\tilde{c}_{\mathcal{X}}$ as follows:

$$\hat{c}_{\mathcal{X}}\left(x,x'\right):=\mathbf{1}\left\{c_{\mathcal{X}}(x,x')<\epsilon\right\}c_{\mathcal{X}}(x,x')+\mathbf{1}\left\{c_{\mathcal{X}}(x,x')\geq\epsilon\right\}\left(\epsilon+\frac{c_{\mathcal{X}}(x,x')-\epsilon}{\tau}\right), \tag{16}$$

where $\tau>0$ is the temperature to control the growing rate of the cost function when x' goes out of the perturbation ball. It is obvious that $\hat{c}_{\mathcal{X}}\left(x,x'\right)$ is continuous and approaches $\tilde{c}_{\mathcal{X}}\left(x,x'\right)$ when $\tau\to 0$. Using the smoothed function $\hat{c}_{\mathcal{X}}\left(x,x'\right)$ from Eq. (16), the final object of our UDR becomes:

$$\inf_{\theta,\lambda\geq 0}\left(\lambda\epsilon+\mathbb{E}_{\mathbb{P}}\left[\sup_{x'}\left\{g_\theta\left(x',x,y\right)-\lambda\hat{c}_{\mathcal{X}}\left(x',x\right)\right\}\right]\right). \tag{17}$$

With this final objective, our training strategy involves three iterative steps at each iteration w.r.t. a batch of data examples, which are shown in Algorithm 1.

1. Craft adversarial examples w.r.t. the current model and the parameter λ. Given the current model θ and the parameter λ, we find the adversarial examples by solving:

$$x^a=\text{argmax}_{x'}\left\{g_\theta(x',x,y)-\lambda\hat{c}_{\mathcal{X}}\left(x',x\right)\right\}, \tag{18}$$

where different methods (i.e., UDR-PGD, UDR-TRADES, etc.) specifies $g_\theta(x',x,y)$ differently.

Similar to other AT methods like PGD-AT, we employ iterative gradient ascent update steps to optimise to find x^a. Specifically, we start from a random example inside the ball B_ϵ and update in k steps with the step size $\eta>0$. Since the magnitude of the gradient $\nabla_{x'}g_\theta(x',x,y)$ is significantly smaller than that of $\nabla_{x'}\hat{c}_{\mathcal{X}}(x',x)$, we use sign $\left(\nabla_{x'}\hat{c}_{\mathcal{X}}(x',x)\right)$ in the update formula rather than $\nabla_{x'}\hat{c}_{\mathcal{X}}(x',x)$. These steps are shown in 2(a) to 2(c) of Algorithm 1.

An important difference from ours to other AT methods is that at each update step, we do not apply any explicit projecting operations onto the ball B_ϵ. Indeed, the parameter λ controls how distant x^a to its benign counterpart x. Thus, this can be viewed as implicitly projecting onto a soft ball governed by the magnitude of the parameter λ and the temperature τ. Specifically, when λ becomes higher, the crafted adversarial examples x^a stay closer to their benign counterparts x and vice versa. When τ is set closer to 0, the smoothed cost function $\hat{c}_\mathcal{X}$ approximates the cost function $\tilde{c}_\mathcal{X}$ more tightly. Thus, our soft-ball projection is more identical to the hard ball projection as in projected gradient ascent.

Algorithm 1 The pseudocode of our proposed method.

Input: training set $\mathcal{D}$, number of iterations T, batch size N, adversary parameters $\{k, \epsilon, \eta\}$
for $t = 1$ to T **do**

1. **Sample** mini-batch $\{x_i, y_i\}_{i=1}^N \sim \mathcal{D}$
2. **Find** adversarial examples $\{x_i^a\}_{i=1}^N$ using Eq. (18)
   (a) **Initialize** randomly: $x_i^0 = x_i + noise$ where $noise \sim \mathcal{U}(-\epsilon, \epsilon)$
   (b) **for** $n = 1$ to k **do**
      i. $x_i^{inter} = x_i^n + \eta \text{sign}\left(\nabla_x g_\theta(x_i^n, x_i, y_i)\right)$
      ii. $x_i^{n+1} = x_i^{inter} - \eta\lambda\nabla_x \hat{c}(x_i^{inter}, x_i)$
   (c) **Clip** to valid range: $x_i^a = clip(x_i^k, 0, 1)$
3. **Update** parameter λ using Eq. (19)
4. **Update** model parameter θ using Eq. (20)

Output: model parameter θ

2. Update the parameter λ. Given current model θ, we craft a batch of adversarial examples $\{x_i^a\}_{i=1}^N$ corresponding to the benign examples $\{x_i\}_{i=1}^N$ crafted as above. Inspired by the Danskin's theorem , we update λ as follows:

$$\lambda_n = \lambda - \eta_\lambda \left(\epsilon - \frac{1}{N} \sum_{i=1}^N \hat{c}_\mathcal{X}(x_i^a, x_i) \right), \tag{19}$$

where $\eta_\lambda > 0$ is a learning rate and λ_n represents the new value of λ.

The proposed update of λ is intuitive: *if the adversarial examples stay close to their benign examples, i.e., $\sum_{i=1}^N \hat{c}_\mathcal{X}(x_i^a, x_i) < \epsilon$, λ decreases to make them more distant to the benign examples and vice versa*. Therefore the adversarial examples are crafted more diversely, which can further strengthen the robustness of the model.

3. Update the model parameter θ. Given the set of adversarial examples $\{x_i^a\}_{i=1}^N$ crafted as above and their benign examples $\{x_i\}_{i=1}^N$ with the labels $\{y_i\}_{i=1}^N$, we update the model parameter θ to minimize $\mathbb{E}_\mathbb{P}\left[\nabla g_\theta(x^a, x, y)\right]$ using the current batches of adversarial and benign examples:

$$\theta_n = \theta - \frac{\eta_\theta}{N} \sum_{i=1}^N \nabla_\theta g_\theta(x_i^a, x_i, y_i), \tag{20}$$

where $\eta_\theta > 0$ is a learning rate and θ_n specifies the new model parameter.

5 EXPERIMENTS

We use MNIST (LeCun et al., 1998), CIFAR10 and CIFAR100 (Krizhevsky et al., 2009) as the benchmark datasets in our experiment. The inputs were normalized to $[0, 1]$. We apply padding 4 pixels at all borders before random cropping and random horizontal flips as used in Zhang et al. (2019). We use both standard CNN architecture (Carlini & Wagner, 2017) and ResNet architecture (He et al., 2016) in our experiment. The architecture and training setting are provided in Appendix D.

We compare our UDR with the SOTA AT methods, i.e., **PGD-AT** (Madry et al., 2018), **TRADES** (Zhang et al., 2019) and **MART** (Wang et al., 2019). Because TRADES and MART performances are strongly dependent on the trade-off ratio (i.e., β in Eq. (6) and (7)) between natural loss and robust loss, we use the original setting in their papers (CIFAR10/CIFAR100: $\beta = 6$ for TRADES/UDR-TRADES, $\beta = 5$ for MART/UDR-MART; MNIST: $\beta = 1$ for all the methods). We also tried with the distributional robustness method WRM (Sinha et al., 2018). However, WRM did not seem to obtain reasonable performance in our experiments. Its results can be found in Appendix F. For all the AT methods, we use $\{k = 40, \epsilon = 0.3, \eta = 0.01\}$ for the MNIST dataset,

Table 1: Comparisons of natural classification accuracy (Nat) and adversarial accuracies against different attacks. Best scores are highlighted in boldface.

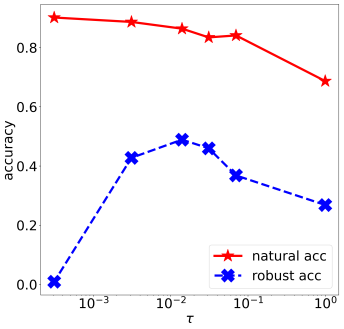
	MNIST				CIFAR10				CIFAR100			
	Nat	PGD	AA	B&B	Nat	PGD	AA	B&B	Nat	PGD	AA	B&B
PGD-AT	99.4	94.0	88.9	91.3	**86.4**	46.0	42.5	44.2	72.4	41.7	39.3	39.6
UDR-PGD	**99.5**	**94.3**	**90.0**	**91.4**	**86.4**	**48.9**	**44.8**	**46.0**	**73.5**	**45.1**	**41.9**	**42.3**
TRADES	**99.4**	95.1	90.9	92.2	80.8	51.9	49.1	50.2	68.1	49.7	46.7	47.2
UDR-TRADES	**99.4**	**96.9**	**92.2**	**95.2**	**84.4**	**53.6**	**49.9**	**51.0**	**69.6**	**49.9**	**47.8**	**48.7**
MART	**99.3**	94.7	90.6	92.9	**81.9**	53.3	48.2	49.3	**68.1**	49.8	44.8	45.4
UDR-MART	**99.3**	**96.0**	**92.3**	**94.4**	80.1	**54.1**	**49.1**	**50.4**	67.5	**52.0**	**48.5**	**48.6**

Table 2: Robustness evaluation under different PGD attack strengths ϵ. *Avg* represents for the average improvement of our DR methods over their counterparts.

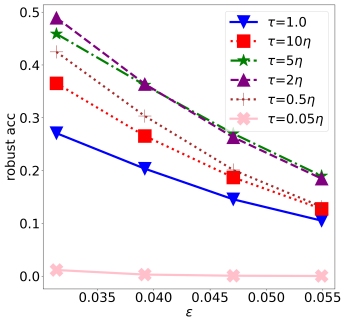
MNIST							
ϵ	0.3	0.325	0.35	0.375	0.4	0.425	Avg
PGD-AT	94.0	67.8	21.1	6.8	2.3	1.2	-
UDR-PGD	**94.3**	**92.9**	**90.1**	**79.2**	**22.3**	**3.8**	31.57
TRADES	95.5	85.2	34.4	5.8	0.6	0.1	-
UDR-TRADES	**96.9**	**96.9**	**95.8**	**95.1**	**94.5**	**88.5**	57.68
MART	94.7	66.1	9.4	0.9	0.2	0.1	-
UDR-MART	**96.0**	**95.0**	**94.1**	**92.8**	**88.8**	**37.7**	55.5

CIFAR10							
ϵ	$\frac{8}{255}$	$\frac{10}{255}$	$\frac{12}{255}$	$\frac{14}{255}$	$\frac{16}{255}$	$\frac{20}{255}$	Avg
PGD-AT	46.0	33.7	23.7	15.2	9.5	3.6	-
UDR-PGD	**48.9**	**36.4**	**26.3**	**18.5**	**13.0**	**7.1**	3.08
TRADES	51.9	42.5	33.7	25.7	18.9	9.1	-
UDR-TRADES	**53.6**	**43.6**	**35.2**	**27.5**	**20.7**	**10.9**	1.62
MART	53.3	43.2	34.1	25.5	18.4	9.0	-
UDR-MART	**54.1**	**46.0**	**37.3**	**29.7**	**22.9**	**12.2**	3.12

CIFAR100							
ϵ	$\frac{10}{1000}$	$\frac{12.5}{1000}$	$\frac{15}{1000}$	$\frac{17.5}{1000}$	$\frac{20}{1000}$	$\frac{25}{1000}$	Avg
PGD-AT	41.7	34.5	27.8	22.6	18.2	11.7	-
UDR-PGD	**45.1**	**38.3**	**31.9**	**26.2**	**21.4**	**14.2**	3.43
TRADES	49.7	44.3	39.9	35.2	31.2	23.5	-
UDR-TRADES	**49.9**	**44.8**	**40.3**	**35.7**	**31.7**	**24.2**	0.47
MART	49.8	45.3	41.0	36.6	32.4	25.1	-
UDR-MART	**52.0**	**47.8**	**44.1**	**40.2**	**36.2**	**29.4**	3.25



(a) Natural/robust accuracy trade-off



(b) Robustness in correlation with τ

Figure 1: Further analysis on parameter sensitivity.

$\{k = 10, \epsilon = 8/255, \eta = 2/255\}$ for the CIFAR10 dataset and $\{k = 10, \epsilon = 0.01, \eta = 0.001\}$ for the CIFAR100 dataset, where k is number of iteration, ϵ is the distortion bound and η is the step size of the adversaries.

We use different SOTA attacks to evaluate the defense methods including: **1) PGD attack** (Madry et al., 2018) which is one of the most widely-used gradient based attacks. For PGD, we set $k = 200$ and $\epsilon = 0.3, \eta = 0.01$ for MNIST, $\epsilon = 8/255, \eta = 2/255$ for CIFAR10, and $\epsilon = 0.01, \eta = 0.001$ for CIFAR100, which are the standard settings. **2) B&B attack** (Brendel et al., 2019) which is a decision based attack. Following Tramer et al. (2020), we initialized with the PGD attack with $k = 20$ and corresponding $\{\epsilon, \eta\}$ then apply B&B attack with 200 steps. **3) Auto-Attack (AA)** (Croce & Hein, 2020b) which is an ensemble methods of four different attacks. We use $\epsilon = 0.3, 8/255, 0.01$, for MNIST, CIFAR10, and CIFAR100, respectively. The distortion metric we use in our experiments is l_∞ for all measures. We use the full test set for PGD and 1000 test samples for the other attacks.

5.1 Main Results

Whitebox Attacks with fixed ϵ. First, we compare the natural and robust accuracy of the AT methods and their counterparts under our UDR framework, against several SOTA attacks. Note

Table 3: Adversarial accuracy in the blackbox settings. *Avg* represents for the average improvement of our DR methods over their counterparts.

Target \ Source	PGD-AT	UDR-P	TRADES	UDR-T	MART	UDR-M	Avg
PGD-AT	-	-	61.6	61.6	61.7	62.4	-
UDR-PGD	-	-	**63.6**	**63.4**	**64.0**	**64.1**	2.0
TRADES	61.2	61.3	-	-	58.9	59.8	-
UDR-TRADES	**62.7**	**62.8**	-	-	**61.1**	**61.6**	1.8
MART	61.4	61.4	58.9	59.5	-	-	-
UDR-MART	**62.3**	**62.1**	**60.1**	**60.5**	-	-	1.0

that in this experiment, the attacks are with their standard settings. The result of this experiment is shown in Table 1. It can be observed that for all the AT methods, our UDR versions are able to boost the model robustness significantly against all the strong attack methods in comparison on all the three datasets. These improvements clearly show that our UDR empowered AT methods achieve the SOTA adversarial robustness performance. Specifically, our UDR-PGD's improvement over PGD on both CIFAR10 and CIFAR100 is over 3% against all the attacks. Similarly, our UDR-MART also improves over MART with a 3% gap on CIFAR100.

Whitebox Attacks with varied ϵ. Recall that UDR is designed to have better generalization capacity than standard adversarial robustness. In this experiment, we exam the generalization capacity by attacking the AT methods (including our UDR variants) with PGD with varied attack strength ϵ while keeping other parameters of PGD attack the same. This is a highly practical scenario where attackers may use various attack strengths that are different from that the model is trained with. The results of this experiment are shown in Table 2. We have the following remarks of the results: **1)** All AT methods perform reasonably well (our UDR variants are better than their counterparts) when PGD attacks with the same ϵ that these methods are trained on. This is shown in the first column on all the datasets, whose results are in line with these in Table 1. **2)** With increased ϵ, the performance of all the AT methods deteriorates, which is natural. However, the advantage of our UDR methods over their counterparts becomes more and more significant. For example, when $\epsilon = 0.375$, all of our UDR methods can achieve at least 80% robust accuracy on MNIST, while others can barely defend. This clearly demonstrates the benefit of our UDR framework on generalization capacity.

Blackbox Attacks. To further exam the generalization of the UDR framework, we conduct the experiment with the blackbox setting via transferred attacks. Specifically, we use PGD to generate adversarial examples according to the model trained with a specific AT method, i.e., the *source* method. Next, we use the generated adversarial examples to attack another AT method, i.e., the *target* method. This is to see whether an AT method can defend against attacks generated from other models. We report the results in Table 3. It can be seen that with better generalization capacity, our UDR methods also outperform their standard counterparts with a margin of 2% in the blackbox setting.

Table 4: Robustness evaluation against Auto-Attack and PGD ($k = 100$) with WRN-34-10 on the full test set of CIFAR10 dataset. (*) Omit the cross-entropy loss of natural images. Detail can be found in Appendix D.

	Nat	PGD	AA	C&W
PGD-AT*	84.93	55.04	52.12	40.85
UDR-PGD*	84.60	55.71	52.98	47.31
TRADES	85.70	56.97	53.82	47.65
UDR-TRADES	84.93	57.35	54.45	49.14
AWP-AT	85.57	57.78	53.91	49.91
UDR-AWP-AT	85.51	58.65	54.40	54.44
Zhang et al. (2020a)	84.52	-	53.51	-
Huang et al. (2020)	83.48	-	53.34	-
Zhang et al. (2019)	84.92	-	53.08	-
Cui et al. (2021)	88.22	-	52.86	-

Results with WideResNet architecture. We would like to provide further experimental results on the CIFAR10 dataset with WideResNet (WRN-34-10) as shown in Table 4. It can be seen that our distributional frameworks consistently outperform their standard AT counterparts in both metrics. More specifically, our improvement over PGD-AT against Auto-Attack is around 0.8%, while that for TRADES is 0.5%. To make a more concrete conclusion, we deploy our framework on a recent SOTA standard AT which is AWP-AT Wu et al. (2020). The result shows that our distributional robustness version (UDR-AWP-AT) also improves its counterpart by 0.5%. With the same setting (i.e., same architecture and without additional data), our UDR-TRADES and UDR-AWP-AT achieve

Table 5: Average norm L_1 and L_∞ of the perturbation $\delta = |x^a - x|_p$

	L_1	L_∞	$p(\delta \leq 0.9\epsilon)$	$p(\delta \leq \epsilon)$	$p(\delta \leq 1.1\epsilon)$
PGD	0.0270	0.031	19.7%	100%	100%
UDR-PGD at epoch 0th	0.0278	0.031	18.9%	100%	100%
UDR-PGD at epoch 200th	0.0301	0.034	19.5%	22.1%	100%

Table 6: Comparison to PGD-AT with different perturbation limitations.

	$\frac{8}{255}$	$\frac{10}{255}$	$\frac{12}{255}$	$\frac{14}{255}$	$\frac{16}{255}$	$\frac{20}{255}$	Avg
PGD-AT at $\epsilon = 0.031$	46.0	33.7	23.7	15.2	9.5	3.6	-
PGD-AT at $\epsilon = 0.034$	46.7	34.8	24.7	16.2	10.1	3.7	0.75
PGD-AT at $\epsilon = 0.037$	44.9	33.3	23.7	15.6	10.0	3.8	-0.07
UDR-PGD at $\epsilon = 0.031$	48.9	36.4	26.3	18.5	13.0	7.1	3.08

better robustness than recently listed methods on RobustBench (Croce et al., 2020).[3] Remarkably, the additional experiment with C&W (L2) attack shows a significant improvement of our distributional methods over standard AT by around 5%. More discussion can be found in Appendix F.

5.2 Analytical Results

Benefit of the soft-ball projection. Here we would like to analytically study why our UDR methods are better than standard AT methods, by taking UDR-PGD and PGD-AT as examples. The visualization on the synthetic dataset can be found in Appendix E. Recall that one of the key differences between UDR-PGD and PGD-AT is that the former uses the soft-ball projection and the later use the hard-ball one, discussed in the second paragraph under Eq. (18). More specifically, Table 5 reports the average norm (L_1 and L_∞) of the perturbation $\delta = |x^a - x|_p$ in PGD and our UDR-PGD. It can be seen that: (i) At the beginning of the training process, there is no difference between the norms of the perturbations generated by PGD and our UDR-PGD. More specifically, most of the pixels lie on the edge of the hard-ball projection (i.e., $p(0.9\epsilon \leq \delta \leq \epsilon) = p(\delta \leq \epsilon) - p(\delta \leq 0.9\epsilon) > 80\%$). (ii) When our model converges, there are 77.9% pixels lying slightly beyond the hard-ball projection (i.e., $p(\delta > \epsilon)$). It is because our soft-ball projection can be adaptive based on the value of . This flexibility helps the adversarial examples reach a better local optimum of the prediction loss, therefore, benefits the adversarial training.

Next, we show that doing PGD adversarial training with larger ϵ cannot achieve the same defence performance as our methods with the soft-ball projection. We conduct more experiments with PGD-AT with $\epsilon = 0.034$ (the final when our model converages) and $\epsilon = 0.037$ to show that simply extending the hard-ball projection doesn't benefit adversarial training. More specifically, the average robustness improvement with $\epsilon = 0.034$ is 0.75%, while there is no improvement with $\epsilon = 0.037$.

Parameter sensitivity of τ. Figure 1a and 1b show the our framework's sensitivity to τ on CIFAR10 under the PGD attack. It can be observed that overly small values of τ can hardly improve adversarial robustness while overly big values of τ may hurt the natural performance ($acc_{nat} = 68.7\%$ with $\tau = 1.0$). Empirically, we find that $\tau = 2\eta$ performs well in our experiments.

6 Conclusions

In this paper, we have presented a new unified distributional robustness framework for adversarial training, which unifies and generalizes standard AT approaches with improved adversarial robustness. By defining a new family of risk functions, our framework facilitates the development of the distributional robustness counterparts of the SOTA AT methods including PGD-AT, TRADES, MART and AWP. Moreover, we introduce a new cost function, which enables us to bridge the connections between standard AT methods and their distributional robustness counterparts and to show that the former ones can be viewed as the special cases of the later ones. Extensive experiments on the benchmark datasets including MNIST, CIFAR10, CIFAR100 show that our proposed algorithms are able to boost the model robustness against strong attacks with better generalization capacity.

[3] RobustBench reported a robust accuracy of 56.17% for AWP-TRADES version from Wu et al. (2020) which is higher than ours but might not be used as a reference.

ACKNOWLEDGEMENT

This work was partially supported by the Australian Defence Science and Technology (DST) Group under the Next Generation Technology Fund (NGTF) scheme. The authors are grateful to the anonymous (meta) reviewers for their helpful comments.

REFERENCES

Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. Wasserstein robust reinforcement learning. *arXiv preprint arXiv:1907.13196*, 2019. 2.1

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pp. 274–283, 2018. 1

Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing. In *International Conference on Learning Representations*, 2020. 1

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton university press, 2009. 1

Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013. 1

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013. 1

Jose Blanchet and Yang Kang. Semi-supervised learning based on distributionally robust optimization. *Data Analysis and Applications 3: Computational, Classification, Financial, Statistical and Stochastic Methods*, 5:1–33, 2020. 2.1

Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019. 2.1, 2.1, 3, A, A, C

Jose Blanchet, Yang Kang, and Karthyek Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019. 1

Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. In *Advances in Neural Information Processing Systems*, pp. 12861–12871, 2019. 5, C

Anh Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving adversarial robustness by enforcing local and global compactness. In *European Conference on Computer Vision*, pp. 209–223. Springer, 2020. C

Anh Bui, Trung Le, He Zhao, Paul Montague, Seyit Camtepe, and Dinh Phung. Understanding and achieving efficient robustness with adversarial supervised contrastive learning. *arXiv preprint arXiv:2101.10027*, 2021a. C

Anh Tuan Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving ensemble robustness by collaboratively promoting and demoting adversarial robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6831–6839, 2021b. C

N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017. 5, D, F

Ruidi Chen and Ioannis C Paschalidis. A robust learning approach for regression models based on distributionally robust optimization. *Journal of Machine Learning Research*, 19(13), 2018. 2.1

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pp. 2196–2205. PMLR, 2020a. C

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020b. 5

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020. 5.1, D

Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. *International Conference on Computer Vision*, 2021. 4

Nhan Dam, Quan Hoang, Trung Le, Tu Dinh Nguyen, Hung Bui, and Dinh Phung. Three-player wasserstein gan via amortised duality. In *International Joint Conference on Artificial Intelligence 2019*, pp. 2202–2208. Association for the Advancement of Artificial Intelligence (AAAI), 2019. 2.1

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010. 1

Esther Derman and Shie Mannor. Distributional robustness and regularization in reinforcement learning. *arXiv preprint arXiv:2003.02894*, 2020. 2.1

Yinpeng Dong, Zhijie Deng, Tianyu Pang, Jun Zhu, and Hang Su. Adversarial distributional training for robust deep learning. *Advances in Neural Information Processing Systems*, 33:8270–8283, 2020. 1, C

John C Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses against mixture covariate shifts. *Under review*, 2019. 2.1

John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021. 1

Rui Gao and Anton J Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016. 1

Rui Gao, Xi Chen, and Anton J Kleywegt. Wasserstein distributionally robust optimization and variation regularization. *arXiv preprint arXiv:1712.06050*, 2017. 1, 2.1

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6572`. 1, 2.2, C

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 5

Quan Hoang, Trung Le, and Dinh Phung. Parameterized rate-distortion stochastic encoder. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4293–4303. PMLR, 13–18 Jul 2020. C

Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. *Advances in Neural Information Processing Systems*, 33, 2020. 4

Viet Huynh, Dinh Phung, and He Zhao. Optimal transport for deep generative models: State of the art and research challenges. In *The 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4450–4457, 2021. 2.1

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 5

Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations Research & Management Science in the Age of Analytics*, pp. 130–166. INFORMS, 2019. 1, 2.1

Trung Le, Dat Do, Tuan Nguyen, Huy Nguyen, Hung Bui, Nhat Ho, and Dinh Phung. On label shift in domain adaptation via wasserstein distance. *arXiv preprint arXiv:2110.15520*, 2021a. 2.1

Trung Le, Tuan Nguyen, Nhat Ho, Hung Bui, and Dinh Phung. Lamda: Label matching deep domain adaptation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6043–6054. PMLR, 18–24 Jul 2021b. 2.1

Trung Le, Anh Bui, Tue Le, He Zhao, Paul Montague, Quan Tran, and Phung Dinh. On global-view based defense via adversarial attack and defense risk guaranteed bounds. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022. C

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. In *NeurIPS*, pp. 2692–2701, 2018. 2.1

Alexander Levine and Soheil Feizi. Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pp. 3938–3947. PMLR, 2020. C

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2.2, 5, C

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015. 1

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. C

Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. *Advances in Neural Information Processing Systems*, 32:5541–5551, 2019. C

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *NIPS*, volume 29, pp. 2208–2216, 2016. 1

Tuan Nguyen, Trung Le, Nhan Dam, Quan Hung Tran, Truyen Nguyen, and Dinh Phung. Tidot: A teacher imitation learning approach for domain adaptation with optimal transport. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 2862–2868. International Joint Conferences on Artificial Intelligence Organization, 8 2021a. URL https://doi.org/10.24963/ijcai.2021/394. Main Track. 2.1

Tuan Nguyen, Trung Le, He Zhao, Quan Hung Tran, Truyen Nguyen, and Dinh Phung. Most: Multi-source domain adaptation via optimal transport for student-teacher learning. In *Uncertainty in Artificial Intelligence*, pp. 225–235. PMLR, 2021b. 2.1

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pp. 4970–4979, 2019. 1, C

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations*, 2020. D

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019. 1, 2.1

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020. D

A. Shafahi, M. Najibi, M A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L S. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pp. 3353–3364, 2019. 1

Soroosh Shafieezadeh-Abadeh, Peyman Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pp. 1576–1584, 2015. 1

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018. 1, 2.1, 2.1, 2.1, 2.3, 3, 3, 5, C, F

Elena Smirnova, Elvis Dohmatob, and Jérémie Mary. Distributionally robust reinforcement learning. *arXiv preprint arXiv:1902.08708*, 2019. 2.1

Matthew Staib and Stefanie Jegelka. Distributionally robust deep learning as a generalization of adversarial training. *NIPS workshop on Machine Learning and Computer Security*, 2017. 1, C

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6199`. 1

Nguyen-Duc Thanh, Le Trung, Zhao He, Cai Jianfei, and Phung Dinh. Particle-based adversarial local distribution regularization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022. C

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems*, pp. 5858–5868, 2019. 1

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33, 2020. 5

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019. 1, 2.2, 5, D

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 5.1, 3, D

Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020. 1, C

Insoon Yang. Wasserstein distributionally robust stochastic control: A data-driven approach. *IEEE Transactions on Automatic Control*, 2020. 2.1

Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In *Advances in Neural Information Processing Systems*, pp. 1829–1839, 2019. 1, C

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482, 2019. 1, 2.2, 5, 4, C

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning*, pp. 11278–11287. PMLR, 2020a. 4

Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*, 2020b. 1

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pp. 7523–7532. PMLR, 2019a. 2.1

He Zhao, Trung Le, Paul Montague, Olivier De Vel, Tamas Abraham, and Dinh Phung. Perturbations are not enough: Generating adversarial examples with spatial distortions. *arXiv preprint arXiv:1910.01329*, 2019b. 1

He Zhao, Thanh Nguyen, Trung Le, Paul Montague, Olivier De Vel, Tamas Abraham, and Dinh Phung. Learning to attack with fewer pixels: A probabilistic post-hoc framework for refining arbitrary dense adversarial attacks. *arXiv preprint arXiv:2010.06131*, 2021a. 1

He Zhao, Dinh Phung, Viet Huynh, Trung Le, and Wray Buntine. Neural topic model via optimal transport. In *International Conference on Learning Representations (ICLR)*, 2021b. `https://openreview.net/forum?id=Oos98K9Lv-k`. 2.1

A THEORETICAL DEVELOPMENT

Theorem 1. *With the cost function $\tilde{c}_{\mathcal{X}}$ defined as above , the optimization problem:*

$$\inf_{\theta,\lambda\geq 0}\left\{\lambda\epsilon+\mathbb{E}_{\mathbb{P}}\left[\sup_{x'}\left\{g_\theta\left(x',x,y\right)-\lambda\tilde{c}_{\mathcal{X}}\left(x',x\right)\right\}\right]\right\} \tag{21}$$

is equivalent to the optimization problem:

$$\inf_{\theta}\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\epsilon(x)} g_\theta\left(x',x,y\right)\right]. \tag{22}$$

Proof. We need to prove that

$$\inf_{\lambda\geq 0}\left\{\lambda\epsilon+\mathbb{E}_{\mathbb{P}}\left[\sup_{x'}\left\{g_\theta\left(x',x,y\right)-\lambda\tilde{c}_{\mathcal{X}}\left(x',x\right)\right\}\right]\right\}=\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\epsilon(x)} g_\theta\left(x',x,y\right)\right]. \tag{23}$$

By the definition of the cost function $\tilde{c}_{\mathcal{X}}$, the LHS of (23) can be rewritten as:

$$\min\left\{\inf_{\lambda>0}\left\{\lambda\epsilon+\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\epsilon(x)}\left\{g_\theta\left(x',x,y\right)-\lambda c_{\mathcal{X}}\left(x',x\right)\right\}\right]\right\},\mathbb{E}_{\mathbb{P}}\left[\sup_{x'} g_\theta\left(x',x,y\right)\right]\right\}. \tag{24}$$

Given any $\lambda>0$ and $x'\in B_\epsilon\left(x\right)$, we have

$$\lambda\epsilon+g_\theta\left(x',x,y\right)-\lambda c_{\mathcal{X}}\left(x',x\right)=g_\theta\left(x',x,y\right)+\lambda\left(\epsilon-c_{\mathcal{X}}\left(x',x\right)\right)\geq\mathbb{E}_{\mathbb{P}}\left[g_\theta\left(x',x,y\right)\right].$$

Hence, we arrive at

$$\lambda\epsilon+\sup_{x'\in B_\epsilon(x)}\left\{g_\theta\left(x',x,y\right)-\lambda c_{\mathcal{X}}\left(x',x\right)\right\}\geq\sup_{x'\in B_e(x)} g_\theta\left(x',x,y\right).$$

$$\lambda\epsilon+\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\epsilon(x)}\left\{g_\theta\left(x',x,y\right)-\lambda c_{\mathcal{X}}\left(x',x\right)\right\}\right]\geq\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_e(x)} g_\theta\left(x',x,y\right)\right].$$

which follows that

$$\begin{aligned}&\inf_{\lambda>0}\left\{\lambda\epsilon+\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\epsilon(x)}\left\{g_\theta\left(x',x,y\right)-\lambda c_{\mathcal{X}}\left(x',x\right)\right\}\right]\right\}\\&\qquad\geq\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_e(x)}\mathbb{E}_{\mathbb{P}}\left[g_\theta\left(x',x,y\right)\right]\right].\end{aligned} \tag{25}$$

We now prove the inequality

$$\lim_{\lambda \to 0^+} \left\{ \lambda \epsilon + \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_\epsilon(x)} \left\{ g_\theta \left(x', x, y \right) - \lambda c_{\mathcal{X}} \left(x', x \right) \right\} \right] \right\} = \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_e(x)} \mathbb{E}_{\mathbb{P}} \left[g_\theta \left(x', x, y \right) \right] \right].$$

Take a sequence $\{\lambda_n\}_{n \geq 1} \to 0^+$. Given a feasible pair (x, y), we define

$$f_n \left(x'; x, y \right) := g_\theta \left(x', x, y \right) + \lambda_n \left[\epsilon - c_{\mathcal{X}} \left(x', x \right) \right], \forall x' \in B_\epsilon \left(x \right).$$

It is evident that $f_n \left(x'; x, y \right)$ converges pointwise to $g_\theta \left(x', x, y \right)$ over the compact set $B_\epsilon \left(x \right)$. Therefore, $f_n \left(x'; x, y \right)$ converges uniformly to $g_\theta \left(x', x, y \right)$ on this set. This follows that

$$\forall \alpha > 0, \exists n_0 = n \left(\alpha \right) : \left| f_n \left(x'; x, y \right) - g_\theta \left(x', x, y \right) \right| < \alpha, \forall x' \in B_\epsilon \left(x \right), n \geq n_0.$$

Hence, we obtain for all $x' \in B_\epsilon \left(x \right)$ and $n \geq n_0$:

$$g_\theta \left(x', x, y \right) - \alpha < f_n \left(x'; x, y \right) < g_\theta \left(x', x, y \right) + \alpha.$$

This leads to the following for all $n \geq n_0$:

$$\sup_{x' \in B_\epsilon(x)} g_\theta \left(x', x, y \right) - \alpha \leq \sup_{x' \in B_\epsilon(x)} f_n \left(x'; x, y \right) \leq \sup_{x' \in B_\epsilon(x)} g_\theta \left(x', x, y \right) + \alpha.$$

Therefore, we obtain:

$$\lim_{n \to \infty} \sup_{x' \in B_\epsilon(x)} f_n \left(x'; x, y \right) = \sup_{x' \in B_\epsilon(x)} g_\theta \left(x', x, y \right)$$

for all feasible pairs (x, y), which further means that

$$\lim_{n \to \infty} \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_\epsilon(x)} f_n \left(x'; x, y \right) \right] = \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_\epsilon(x)} g_\theta \left(x', x, y \right) \right],$$

or equivalently

$$\lim_{n \to \infty} \mathbb{E}_{\mathbb{P}} \left[\lambda_n \epsilon + \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_\epsilon(x)} \left\{ g_\theta \left(x', x, y \right) - \lambda_n c_{\mathcal{X}} \left(x', x \right) \right\} \right] \right] = \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_\epsilon(x)} g_\theta \left(x', x, y \right) \right]. \tag{26}$$

Because Eq. (26) holds for every sequence $\{\lambda_n\}_{n \geq 1} \to 0^+$, we reach

$$\lim_{\lambda \to 0^+} \left\{ \lambda \epsilon + \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_\epsilon(x)} \left\{ g_\theta \left(x', x, y \right) - \lambda c_{\mathcal{X}} \left(x', x \right) \right\} \right] \right\} = \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_e(x)} \mathbb{E}_{\mathbb{P}} \left[g_\theta \left(x', x, y \right) \right] \right]. \tag{27}$$

By combining (25) and (27), we reach

$$\inf_{\lambda > 0} \left\{ \lambda \epsilon + \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_\epsilon(x)} \left\{ g_\theta \left(x', x, y \right) - \lambda c_{\mathcal{X}} \left(x', x \right) \right\} \right] \right\} = \mathbb{E}_{\mathbb{P}} \left[\sup_{x' \in B_e(x)} \mathbb{E}_{\mathbb{P}} \left[g_\theta \left(x', x, y \right) \right] \right]. \tag{28}$$

Finally, we have

$$\begin{aligned}
&\inf_{\lambda\geq 0}\left\{\lambda\epsilon+\mathbb{E}_{\mathbb{P}}\left[\sup_{x'}\left\{g_\theta\left(x',x,y\right)-\lambda\tilde{c}_{\mathcal{X}}\left(x',x\right)\right\}\right]\right\}\\
=&\min\left\{\inf_{\lambda>0}\left\{\lambda\epsilon+\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\epsilon(x)}\left\{g_\theta\left(x',x,y\right)-\lambda c_{\mathcal{X}}\left(x',x\right)\right\}\right]\right\},\mathbb{E}_{\mathbb{P}}\left[\sup_{x'}g_\theta\left(x',x,y\right)\right]\right\}\\
=&\min\left\{\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_e(x)}\mathbb{E}_{\mathbb{P}}\left[g_\theta\left(x',x,y\right)\right]\right],\mathbb{E}_{\mathbb{P}}\left[\sup_{x'}g_\theta\left(x',x,y\right)\right]\right\}\\
=&\mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_e(x)}\mathbb{E}_{\mathbb{P}}\left[g_\theta\left(x',x,y\right)\right]\right].
\end{aligned}$$

That concludes our proof. □

One of most technical challenge we need to bypass in our work is that in theory developed in Blanchet & Murthy (2019), to equivalently transform the primal form to the dual form, it requires the cost function to be finite. In the following theorem, we reprove the equivalence of the primal and dual forms in our context.

Theorem 2. *Assume that the function g is upper-bounded by a number L. We have the following equality between the primal form and dual form*

$$\sup_{\mathbb{Q}:\mathcal{W}_c(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon}\mathbb{E}_{\mathbb{Q}}\left[g\left(z'\right)\right]=\inf_{\lambda\geq 0}\left\{\lambda\epsilon+\mathbb{E}_{\mathbb{P}_\triangle}\left[\sup_{z'}\left\{g\left(z'\right)-\lambda c\left(z',z\right)\right\}\right]\right\},$$

where $z=(x,x,y)$, $z'=(x',x'',y')$, and we have defined

$$c\left(z,z'\right)=\tilde{c}_{\mathcal{X}}\left(x,x'\right)+\infty\times\tilde{c}_{\mathcal{X}}\left(x,x''\right)+\infty\times\mathbf{1}\left\{y\neq y'\right\},$$

for which we have defined

$$\tilde{c}_{\mathcal{X}}\left(x,x'\right)=\begin{cases}c_{\mathcal{X}}\left(x,x'\right) & \textit{if } c_{\mathcal{X}}(x,x')\leq\epsilon\\ \infty & \textit{otherwise}.\end{cases}$$

Proof. Given a positive integer number $n>0$, we define the following metrics:

$$c^n\left(z,z'\right)=\tilde{c}^n_{\mathcal{X}}\left(x,x'\right)+\infty\times\tilde{c}^n_{\mathcal{X}}\left(x,x''\right)+\infty\times\mathbf{1}\left\{y\neq y'\right\},$$

$$\tilde{c}^n_{\mathcal{X}}\left(x,x'\right)=\begin{cases}c_{\mathcal{X}}\left(x,x'\right) & \text{if } c_{\mathcal{X}}\left(x,x'\right)<\epsilon.\\ n & \text{otherwise.}\end{cases}$$

We have $\tilde{c}^n_{\mathcal{X}}\nearrow\tilde{c}_{\mathcal{X}}$ and $c^n\nearrow c$. We now prove that

$$\sup_{\mathbb{Q}:\mathcal{W}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon}\mathbb{E}_{\mathbb{Q}}\left[g\left(z'\right)\right]=\inf_n\sup_{\mathbb{Q}:\mathcal{W}_{c^n}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon}\mathbb{E}_{\mathbb{Q}}\left[g\left(z'\right)\right].$$

In fact, for each n, we have $c^n\leq c$. Therefore, $\mathcal{W}_{c^n}\left(\mathbb{Q},\mathbb{P}_\triangle\right)\leq\mathcal{W}_c\left(\mathbb{Q},\mathbb{P}_\triangle\right)$, hence $\left\{\mathbb{Q}:\mathcal{W}_c\left(\mathbb{Q},\mathbb{P}_\triangle\right)<\epsilon\right\}\subset\left\{\mathbb{Q}:\mathcal{W}_{c^n}\left(\mathbb{Q},\mathbb{P}_\triangle\right)<\epsilon\right\}$, implying that

$$\sup_{\mathbb{Q}:\mathcal{W}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon}\mathbb{E}_{\mathbb{Q}}\left[g\left(z'\right)\right]\leq\sup_{\mathbb{Q}:\mathcal{W}_{c^n}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon}\mathbb{E}_{\mathbb{Q}}\left[g\left(z'\right)\right].$$

$$\sup_{\mathbb{Q}:\mathcal{W}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon}\mathbb{E}_{\mathbb{Q}}\left[g\left(z'\right)\right]\leq\inf_n\sup_{\mathbb{Q}:\mathcal{W}_{c^n}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon}\mathbb{E}_{\mathbb{Q}}\left[g\left(z'\right)\right].$$

Let us define

$$A=\cup_{(x,y)\in\mathcal{D}}\left\{\left(z,z'\right):z=\left(x,x,y\right),z'=\left(x',x'',y'\right),c_{\mathcal{X}}\left(x,x'\right)<\epsilon,x''=x,y'=y\right\},$$

$$B=\cup_{(x,y)\in\mathcal{D}}\left\{\left(z,z'\right):z=\left(x,x,y\right),z'=\left(x',x'',y'\right),c_{\mathcal{X}}\left(x,x'\right)\geq\epsilon,x''=x,y'=y\right\}.$$

To simplify our proof, without generalization ability, for each n, we denote $\mathbb{Q}_n$ as the distribution in $\{\mathbb{Q} : \mathcal{W}_{c^n}(\mathbb{Q}, \mathbb{P}_\triangle) < \epsilon\}$ that peaks $\mathbb{E}_{\mathbb{Q}}[g_\theta(z')]$ and γ_n as the optimal transport plan of $\mathcal{W}_{c^n}(\mathbb{Q}_n, \mathbb{P}_\triangle)$ which admits $\mathbb{P}_\triangle$ and $\mathbb{Q}_n$ as its marginals. Note that because $\mathcal{W}_{c^n}(\mathbb{Q}_n, \mathbb{P}_\triangle) < \epsilon$, the support of γ_n almost surely determines on $A \cup B$.We then have

$$\begin{aligned}
\mathcal{W}_{c^n}(\mathbb{Q}_n, \mathbb{P}_\triangle) &= \int c^n(z, z')\, d\gamma_n(z, z') \\
&= \int_A c^n(z, z')\, d\gamma_n(z, z') + \int_B c^n(z, z')\, d\gamma_n(z, z') \\
&= \int_A c_{\mathcal{X}}(x, x')\, d\gamma_n(z, z') + \int_B n d\gamma_n(z, z') \\
&= \int_A c_{\mathcal{X}}(x, x')\, d\gamma_n(z, z') + n\gamma_n(B) < \epsilon.
\end{aligned}$$

Therefore, we obtain: $\gamma_n(B) < \frac{\epsilon}{n}$. We now define $\bar{\gamma}_n$ as a restricted measure of γ_n on A, meaning that $\bar{\gamma}_n(C) = \frac{\gamma_n(A)+\gamma_n(B)}{\gamma_n(A)}\gamma_n(C) = \left(1 + o\left(n^{-1}\right)\right)\gamma_n(C)$ for any measure set $C \subset A$, where $\lim_{n\to\infty} o\left(n^{-1}\right) = 0$. Let $\mathbb{P}_n$ as marginal distribution of $\mathbb{Q}_n$ corresponding to the dimensions of z'. It appears that

$$\begin{aligned}
\mathcal{W}_c(\mathbb{P}_n, \mathbb{P}_\triangle) &\leq \int_A c(z, z')\, d\bar{\gamma}_n(z, z') + \int_B c(z, z')\, d\bar{\gamma}_n(z, z') \\
&\overset{(1)}{=} \int_A c_{\mathcal{X}}(x, x')\, d\bar{\gamma}_n(z, z') < \int_A \epsilon d\bar{\gamma}_n(z, z') = \epsilon.
\end{aligned}$$

Note that we have $\overset{(1)}{=}$ because $\bar{\gamma}_n(B) = 0$.

This implies that $\mathbb{P}_n \in \{\mathbb{Q} : \mathcal{W}_c(\mathbb{Q}, \mathbb{P}_\triangle) < \epsilon\}$, which follows that

$$\begin{aligned}
\sup_{\mathbb{Q}:\mathcal{W}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon} \mathbb{E}_{\mathbb{Q}}[g(z')] &\geq \mathbb{E}_{\mathbb{P}_n}[g_\theta(z')] = \mathbb{E}_{\bar{\gamma}_n}[g(z')] \\
&= \int_A g(z')\, d\bar{\gamma}_n(z, z') + \int_B g(z')\, d\bar{\gamma}_n(z, z') \\
&\overset{(1)}{=} \int_A g(z')\, d\bar{\gamma}_n(z, z') = \frac{\gamma_n(A) + \gamma_n(B)}{\gamma_n(A)} \int_A g(z')\, d\gamma_n(z, z') \\
&= \left(1 + o\left(n^{-1}\right)\right)\left[\int_{A\cup B} g(z')\, d\gamma_n(z, z') - \int_B g(z')\, d\gamma_n(z, z')\right] \\
&= \left(1 + o\left(n^{-1}\right)\right)\left[\int_{A\cup B} g(z')\, d\mathbb{Q}_n(z') - \int_B g(z')\, d\gamma_n(z, z')\right] \\
&\geq \left(1 + o\left(n^{-1}\right)\right)\left[\sup_{\mathbb{Q}:\mathcal{W}_{c^n}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon} \mathbb{E}_{\mathbb{Q}}[g_\theta(z')] - \int_B L d\gamma_n(z, z')\right]
\end{aligned}$$

$$\begin{aligned}
\sup_{\mathbb{Q}:\mathcal{W}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon} \mathbb{E}_{\mathbb{Q}}[g(z')] &\geq \left(1 + o\left(n^{-1}\right)\right)\left[\sup_{\mathbb{Q}:\mathcal{W}_{c^n}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon} \mathbb{E}_{\mathbb{Q}}[g_\theta(z')] - L\gamma_n(B)\right] \\
&\overset{(2)}{\geq} \left(1 + o\left(n^{-1}\right)\right)\left[\sup_{\mathbb{Q}:\mathcal{W}_{c^n}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon} \mathbb{E}_{\mathbb{Q}}[g_\theta(z')] - \frac{L\epsilon}{n}\right].
\end{aligned}$$

Note that we have $\overset{(1)}{=}$ due to $\bar{\gamma}_n(B) = 0$ and $\overset{(2)}{\geq}$ due to $\gamma_n(B) < \frac{\epsilon}{n}$. Therefore, we reach the conclusion

$$\sup_{\mathbb{Q}:\mathcal{W}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon} \mathbb{E}_{\mathbb{Q}}[g_\theta(z')] = \inf_n \sup_{\mathbb{Q}:\mathcal{W}_{c^n}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon} \mathbb{E}_{\mathbb{Q}}[g_\theta(z')].$$

Next, we apply primal-dual form in Blanchet & Murthy (2019) for the finite metric $\tilde{c}^n_{\mathcal{X}}$ to reach

$$\sup_{\mathcal{W}_{c^n}(\mathbb{Q},\mathbb{P}_\triangle)<\epsilon} \mathbb{E}_{\mathbb{Q}}\left[g_\theta\left(z'\right)\right] = \inf_{\lambda\geq 0}\left\{\lambda\epsilon + \mathbb{E}_{\mathbb{P}_\triangle}\left[\sup_{z'}\left\{g_\theta\left(z'\right) - \lambda c^n\left(z', z\right)\right\}\right]\right\}.$$

Finally, taking $n \to \infty$ and noting that $c^n \nearrow c$, we reach the conclusion. □

B Further Explanation Why Our UDR can utilize Global Information and the Advantage of Soft-ball

Algorithm 1 The pseudocode of our proposed method.

Input: training set $\mathcal{D}$, number of iterations T, batch size N, adversary parameters $\{k, \epsilon, \eta\}$
for $t = 1$ to T **do**

1. **Sample** mini-batch $\{x_i, y_i\}_{i=1}^N \sim \mathcal{D}$
2. **Find** adversarial examples $\{x_i^a\}_{i=1}^N$ using Eq. (18)
   (a) **Initialize** randomly: $x_i^0 = x_i + noise$ where $noise \sim \mathcal{U}(-\epsilon, \epsilon)$
   (b) **for** $n = 1$ to k **do**
      i. $x_i^{inter} = x_i^n + \eta \text{sign}\left(\nabla_x g_\theta(x_i^n, x_i, y_i)\right)$
      ii. $x_i^{n+1} = x_i^{inter} - \eta\lambda\nabla_x \hat{c}(x_i^{inter}, x_i)$
   (c) **Clip** to valid range: $x_i^a = clip(x_i^k, 0, 1)$
3. **Update** parameter λ using Eq. (19)
4. **Update** model parameter θ using Eq. (20)

Output: model parameter θ

The advantage of our soft ball comes from the adaptive capability of λ, which is controlled by a global effect regarding how far adversarial examples x_i^a from benign examples x_i. Let us revisit Algorithm 1. In the step 2.(b).i, we update

$$x_i^{inter} = x_i^n + \eta \text{sign}\left(\nabla_x g_\theta(x_i^n, x_i, y_i)\right)$$

with the aim to find x_i^{inter} that can maximize $g_\theta(\cdot, x_i, y_i)$ as in the standard versions.

Furthermore, in the step 2.(b).ii, we update

$$\begin{aligned} x_i^{n+1} &= x_i^{inter} - \eta\lambda\nabla_x \hat{c}(x_i^{inter}, x_i) = x_i^{inter} - \eta\lambda\left(x_i^{inter} - x_i\right) \\ &= \left(1 - \eta\lambda\right) x_i^{inter} + \eta\lambda x_i, \end{aligned} \tag{29}$$

where we assume L2 cost $c(x, x') = \frac{1}{2}\|x - x'\|^2$ is used. It is evident that x_i^{n+1} is an interpolation point of x_i^{inter} and x_i, hence x_i^{n+1} is drawn back to x_i wherein the drawn-back amount is proportional to $\eta\lambda$.

We now revisit the formula to update λ as Eq. (19)

$$\lambda_n = \lambda - \eta_\lambda\left(\epsilon - \frac{1}{N}\sum_{i=1}^N \hat{c}_{\mathcal{X}}(x_i^a, x_i)\right),$$

which indicates that λ is globally controlled. More specifically, if average distance from x_i^a to x_i (i.e., $\frac{1}{N}\sum_{i=1}^N \hat{c}_{\mathcal{X}}(x_i^a, x_i)$) is less than ϵ (i.e., adversarial examples are globally close to benign examples), λ is adapted decreasingly. Linking with the formula in Eq. (29), in this case, x_i^{n+1} gets back to x_i less aggressively to maintain the distance between x_i^a and x_i. Otherwise, adversarial examples are globally far from benign examples, λ is adapted increasingly. In this case, x_i^{n+1} gets back to x_i more aggressively to reduce more the distance between x_i^a and x_i.

C Related Work

Adversarial Attacks. In this paper, we are interested in image classification tasks and focus on the adversaries that add small perturbations to the pixels of an image to generate attacks based on gradients, which are the most popular and effective. FGSM (Goodfellow et al., 2015) and PGD (Madry et al., 2018) are the most representative gradient-based attacks and PGD is the most widely-used one, due to its effectiveness and simplicity. Recently, there are several variants of PGD that achieve improved performance, for example, Auto-Attack by ensembling PGD with other attacks (Croce & Hein, 2020a) and the B&B method (Brendel et al., 2019) by attacking with decision-based boundary initialized with PGD. Along with PGD, these attacks have been considered as benchmark attacks for adversarial robustness.

Adversarial defenses. Among various kinds of defense approaches, Adversarial Training (AT), originating in Goodfellow et al. (2015), has drawn the most research attention. Given its effectiveness and efficiency, many variants of AT have been proposed with (1) different types of adversarial examples (e.g., the worst-case examples as in Goodfellow et al. (2015) or most divergent examples as in Zhang et al. (2019)), (2) different searching strategies (e.g., non-iterative FGSM and Rand FGSM (Madry et al., 2018)), (3) additional regularizations (e.g., adding constraints in the latent space (Zhang & Wang, 2019; Bui et al., 2020; 2021a; Hoang et al., 2020)), and (4) different model architectures (e.g., activation function (Xie et al., 2020) or ensemble models (Pang et al., 2019; Bui et al., 2021b)).

Distributional robustness. There have been a few works attempting to connect DR with adversarial machine learning or improve adversarial robustness based on the ideas of DR (Sinha et al., 2018; Staib & Jegelka, 2017; Miyato et al., 2018; Zhang & Wang, 2019; Najafi et al., 2019; Levine & Feizi, 2020; Le et al., 2022; Thanh et al., 2022). A recent work of Dong et al. (2020) proposes a new AT algorithm by constructing a distribution over each data sample to model the adversarial examples around it, which is still in the category of pointwise adversary (Sinha et al., 2018) and has no relations to DR. Although its aim of enhancing adversarial robustness is visually related ours, its mythology is different from ours. Therefore, we consider Sinha et al. (2018); Staib & Jegelka (2017) as the most relevant ones to ours. Specifically, both works leverage the dual form of Wasserstein DR (Blanchet & Murthy, 2019) for searching worst-case perturbations for AT, where Sinha et al. (2018) (WRM) focuses on certified robustness with comprehensive study on the tradeoffs between complexity, generality, guarantees, and speed, while Staib & Jegelka (2017) (FDRO) points out that Wasserstein robust optimization can be viewed as the generalization to standard AT.

Although our study is inspired by the two works, there are significant differences and new results of ours: **1)** We introduce a new Wasserstein cost function and a new series of risk functions in WDR, which facilitate our framework to generalize and encompass many SOTA AT methods. While WRM can be viewed as the generalization to PGD-AT only. **2)** Most importantly, although WDR has been demonstrated to have superior properties over standard AT in the two papers, unfortunately, WRM and FDRO have not been observed to outperform standard AT methods. For example, the experiments of FDRO show that adversarial robustness on MNIST of WRM and FDRO is worse than that of AT with PGD and iterative-FGSM (Staib & Jegelka, 2017). Moreover, WRM and FDRO's effectiveness either on more complex colored images (e.g., CIFAR10) or against more advanced attacks (e.g., Auto-Attack) has not been carefully studied yet. On the contrary, we conduct extensive experiments to show the SOTA performance of our proposed algorithms.

D Experimental Settings

For MNIST dataset. We use a standard CNN architecture for the MNIST dataset which is identical with that in Carlini & Wagner (2017). We use the SGD optimizer with momentum 0.9, starting learning rate 1e-2 and reduce the learning rate ($\times 0.1$) at epoch {55, 75, 90}. We train with 100 epochs.

For CIFAR10 and CIFAR100 dataset with ResNet18 architecture. We use the ResNet18 for the CIFAR10 and CIFAR100 dataset. We use the SGD optimizer with momentum 0.9, weight decay 3.5e-3 as in the official implementation from Wang et al. (2019).[4] The starting learning rate 1e-2 and reduce the learning rate ($\times 0.1$) at epoch {75, 90, 100}. We train with 200 epochs.

[4]https://github.com/YisenWang/MART

For hard/soft-ball projection experiments. For PGD-AT, we use the following three ad-hoc strategies for ϵ: 1) Fixing $\epsilon = 8/255$; 2) Fixing $\epsilon = 16/255$; 3) Gradually increasing/decreasing $\epsilon$from $8/255$ to $16/255$, from epoch 20 to epoch 70, with the changing rate $\delta = 8/255/50$ per epoch. For example, the perturbation bound of the increasing strategy at epoch i is: $\epsilon_i = \min(\frac{16}{255}, \max(\frac{8}{255}, \frac{8}{255} + (i-20)\delta))$; the perturbation bound for decreasing strategy is: $\epsilon_i = \max(\frac{8}{255}, \min(\frac{16}{255}, \frac{16}{255} - (i-20)\delta))$.

For CIFAR10 with WideResNet architecture. We follow the setting in Pang et al. (2020) for the additional experiments on CIFAR10 with WideResNet-34-10 architecture. More specifically, we train with 200 epochs with SGD optimizer with momentum 0.9, weight decay 5e-4. The learning rate is 0.1 and reduce at epoch 100th and 150th with rate 0.1 (Rice et al., 2020; Wu et al., 2020). More importantly, to match the performance as reported in Croce et al. (2020), we omit the cross-entropy loss of the natural images in PGD-AT and UDR-PGD. More specifically, the objective function of PGD-AT in Eq. (5) has been replaced by: $\inf_\theta \mathbb{E}_\mathbb{P}\left[\beta \sup_{x' \in B_\epsilon(x)} CE\left(h_\theta\left(x'\right), y\right)\right]$ while the unified risk function for UDR-PGD to be: $g_\theta\left(z'\right) := \beta CE\left(h_\theta\left(x'\right), y'\right)$. We also switch Batch Normalization layer to evaluation stage when crafting adversarial examples as adviced in Pang et al. (2020).

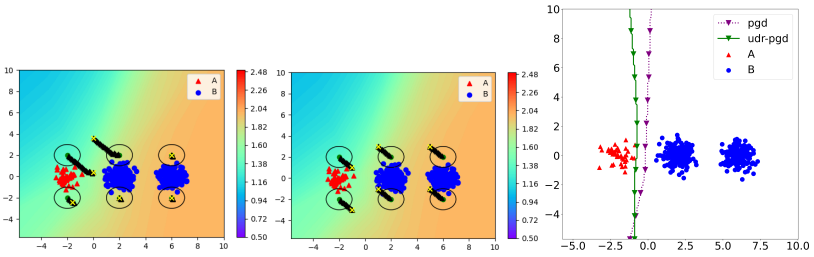
E Visualizing the Benefit of Distributional Robustness

Synthetic dataset setting. We conduct an experiment on a synthetic dataset with a simple MLP model to visualize the benefit of our UDR framework over the standard AT methods, by taking UDR-PGD and PGD-AT as examples. The synthetic dataset consists of three clusters A, B1, B2 where A, B are two classes as shown in Figure 2c. The data points are sampled from normal distributions, i.e., $A \sim \mathcal{N}\left((-2, 0), \Sigma\right), B1 \sim \mathcal{N}\left((2, 0), \Sigma\right)$ and $B2 \sim \mathcal{N}\left((6, 0), \Sigma\right)$ where $\Sigma = 0.5 * I$ with I is the identity matrix. There are total 10k training samples and 2k testing samples with densities of three clusters are 10%, 50% and 40%, respectively. We use a simple model of 4 Fully-Connected (FC) layers as follows: Input –> ReLU(FC(10)) –> ReLU(FC(10)) –> ReLU(FC(10)) –> Softmax(FC(2)), where FC(k) represents for FC with k hidden units. We use Adam optimizer with learning rate 1e-3 and train with 30 epochs. We use $\{k = 20, \epsilon = 1.0, \eta = 0.1\}$ for adversarial training (either PGD-AT or UDR-PGD) and PGD attack with $\{k = 200, \epsilon = 2.0, \eta = 0.1\}$ for evaluation.

It is a worth noting that while the distance between clusters is 2, we limit the perturbation $\epsilon = 1$ for the adversarial training to show the advantage on the flexibility of the soft-ball projection on the same/limited perturbation budget. Intuitively, cluster A has the lowest density (10%), therefore, the ideal decision boundary should be surrounded cluster A which sacrifices the robustness of the cluster A but increases the overall robustness eventually.

Comparison between UDR-PGD and PGD-AT. First, we visualize the trajectory of adversarial example from PGD and our UDR-PGD as in Figures 2b,2a to compare behaviors of two adversaries on the same pre-trained model. It can be seen that: (i) the PGD's adversarial examples and ours are pushed toward the lower confident region to maximize the prediction loss $g_\theta(x', x, y)$; (ii) however, while the adversarial examples of PGD are limited on the hard-projection ball, our adversarial examples have more flexibility. Specifically, those are close to the decision boundary (cluster A, B1) can go further, while those are distant to the decision boundary (cluster B2) stay close to the original input. This flexibility helps the adversarial examples reach better local optimum of the prediction loss, hence, benefits the adversarial training. Consequently, as shown in Figure 2c the final decision boundary of our UDR-PGD is closer to the ideal decision boundary than that of PGD-AT, hence, achieving a better robustness. Quantitative result shows that the robust accuracy of our UDR-PGD is 82.6%, while that of PGD-AT is 74.5% with the same PGD attack $\{k = 200, \epsilon = 2.0, \eta = 0.1\}$.

Comparison among UDR-PGD settings. Here we would like to provide more understanding about our framework through the experiment with PGD-AT as shown in Figure 3. First, we compare the trajectories of the adversarial examples of UDR-PGD with different λ as shown in Figures 3a,3b. It can be seen that the crafted adversarial examples stay closer to their benign counterparts when λ becomes higher (i.e., $\lambda = 0.1$ in Figure 3a). In contrast, the soft-projection ball is extended when λ becomes smaller (i.e., $\lambda = 0.01$ in Figure 3b). On the other hand, with the same λ but smaller τas

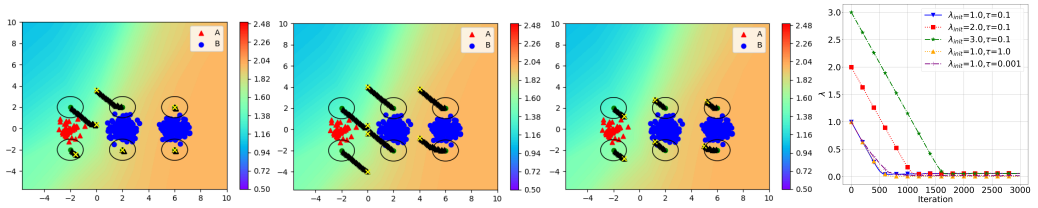


(a) UDR-PGD (b) PGD (c) Decision Boundary

Figure 2: (a)/(b): Trajectory of PGD and UDR-PGD adversarial examples. Each trajectory includes 20 intermediate steps. For better visualization, we do not use random initialization. The model is the natural training at epoch 1. (c) The final decision boundary comparison.

shown in Figure 3c, the soft-ball projection is more identical to the hard ball projection as shown in Figure 2b. These behaviors concur with the theoretical expectation as discussed in Section 4.1 in the main paper.

Figure 3d shows the learning progress of parameter λ. It can be observed that (i) the λ converges to 0 regardless of its initialization value and (ii) the convergence rate of λ depends on the parameter τ (i.e., smaller τ slower convergence). We choose $\tau = 2\eta$ for the experiments on real-world image datasets.



(a) $\lambda = 0.1, \tau = 1.0$ (b) $\lambda = 0.01, \tau = 1.0$ (c) $\lambda = 0.01, \tau = 0.01$ (d) Changing of λ

Figure 3: (a)/(b)/(c): Trajectory of UDR-PGD adversarial examples with different settings. Each trajectory includes 20 intermediate steps. For better visualization, we do not use random initialization. The model is the natural training at epoch 1. (d) The changing of parameter λ.

Further results of soft-ball projection. In Figure 4, we compare our UDR-PGD with the soft-ball projection to PGD-AT with the hard-ball projection with different settings against the PGD attack on CIFAR10. For PGD-AT, we use the following three ad-hoc strategies for ϵ: 1) Fixing $\epsilon = 8/255$; 2) Fixing $\epsilon = 16/255$; 3) Gradually increasing/decreasing ϵ from 8/255 to 16/255 (Refer to Appendix D for details). It can be seen that it is hard to find an effective strategy of the perturbation boundary of the hard-ball projection for PGD-AT, which can outperform ours. This demonstrates the benefit of our soft-project operation.

F More results and analysis

Further results with C&W (L2) attack. We enrich the comprehensiveness of the experiments by further evaluating the defense methods with C&W (L2) attack (Carlini & Wagner, 2017) which is a very strong optimization based attack. The experiment has been conducted on the CIFAR10 dataset with WideResNet architecture. The hyper-parameters are $c \in \{0.5, 0.7, 1.0\}, kappa = 0, steps = 1000, lr = 0.01$ where $kappa$ is the confidence coefficient and c is box-constraint coefficient.[5] As shown in Table 7, our distributional robustness version significantly outperform the standard ones in term of robust accuracy. For example, against C&W (c=0.5) attack, the robust accuracy gap between UDR-PGD and PGD-AT is 6% while that for UDR-AWP-AT and AWP-AT is around 5%. The average improvement of robust accuracies against different levels of attack strengths is around

[5]We use the implementation from https://github.com/Harry24k/adversarial-attacks-pytorch

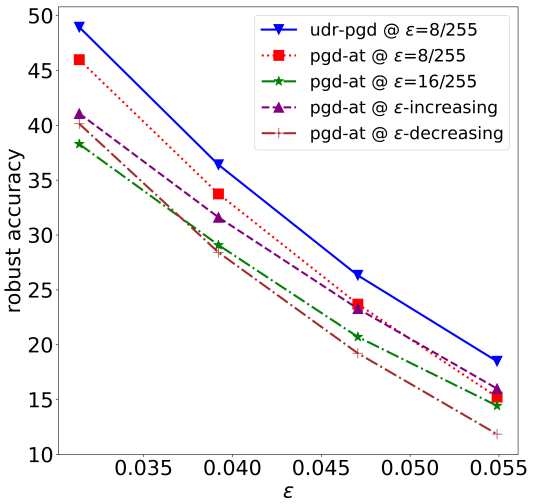


Figure 4: Hard/soft-ball projections

Table 7: Robustness evaluation against C&W attack with WRN-34-10 on the full test set of the CIFAR10 dataset (10K test images). c is box-constraint coefficient. (*) Omit the cross-entropy loss of natural images.

	Nat	$c = 0.5$	$c = 0.7$	$c = 1.0$	Avg-Gap
PGD-AT*	84.93	40.85	25.90	12.95	-
UDR-PGD*	84.60	**47.31**	**31.58**	**16.57**	5.25
TRADES	85.70	47.65	34.30	21.03	-
UDR-TRADES	84.93	**49.14**	**36.33**	**23.28**	1.92
AWP-AT	85.57	49.91	34.31	18.97	-
UDR-AWP-AT	85.51	**54.44**	**39.86**	**23.61**	4.91

5%. This result strongly emphasizes the contribution of our distributional robustness and the soft-ball projection over the standard adversarial training.

Experimental results of WRM (Sinha et al., 2018). The performance of WRM highly depends on the Lagrange dual parameter γ (or $\epsilon = 0.5/\gamma$ in their implementation[6]), which controls the robustness level. As mentioned in their paper, with large γ, the method is less robust but more tractable. Generally, decreasing γ will reduce the natural accuracy but increase the robustness of the model as shown in Table 8. We obtained the best performance on MNIST with $\gamma = 0.05$ (CNN), while on CIFAR10 and CIFAR100 with $\gamma = 0.5$ (ResNet18). The best results with three benchmark datasets have been reported as in Table 9 (recall results from Table 1). It is a worth mentioning that while we could obtain a similar performance as reported Sinha et al. (2017) on the MNIST dataset with their architecture (3 Convolution layers + 1 FC layer), however, WRM seems much less effective with larger architectures.

Table 8: Result of WRM with different $\epsilon = 0.5/\gamma$ on the CIFAR10 dataset.

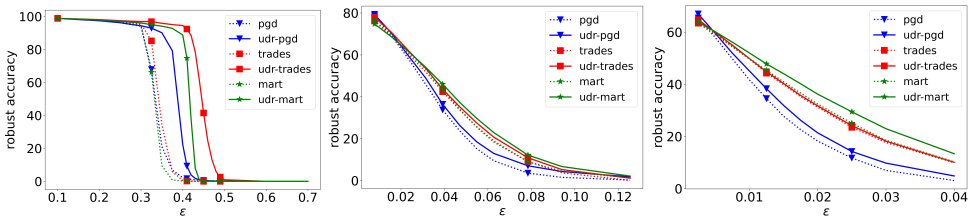
	Nat	PGD	AA	B&B
$\epsilon = 0.1$	90.9	15.3	13.7	15.8
$\epsilon = 0.5$	86.7	33.9	32.6	35.4
$\epsilon = 1.0$	83.7	40.9	39.8	41.4
$\epsilon = 2.0$	79.4	45.4	43.6	45.5
$\epsilon = 5.0$	71.6	47.5	45.2	46.2
$\epsilon = 10.0$	65.0	46.6	43.4	44.4

[6]https://github.com/duchi-lab/certifiable-distributional-robustness/blob/master/attacks_tf.py

Table 9: Comparisons of natural classification accuracy (Nat) and adversarial accuracies against different attacks. Recall results from Table 1 with additional results of WRM. Best scores are highlighted in boldface.

	MNIST				CIFAR10				CIFAR100			
	Nat	PGD	AA	B&B	Nat	PGD	AA	B&B	Nat	PGD	AA	B&B
WRM	91.8	27.1	4.5	8.2	83.7	40.9	39.8	41.4	56.6	24.7	21.3	22.9
PGD-AT	99.4	94.0	88.9	91.3	**86.4**	46.0	42.5	44.2	72.4	41.7	39.3	39.6
UDR-PGD	**99.5**	**94.3**	**90.0**	**91.4**	**86.4**	**48.9**	**44.8**	**46.0**	**73.5**	**45.1**	**41.9**	**42.3**

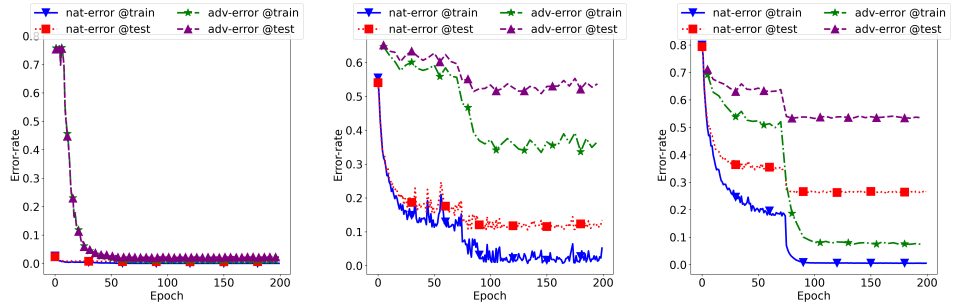
Further results of whitebox attacks with varied ϵ. Here we would like to provide more results on defending against whitebox attacks with a bigger range of ϵ as shown in Figure 5. It can be seen that in a wide range of attack strengths our DR methods consistently outperform their AT counterparts.



(a) MNIST (b) CIFAR10 (c) CIFAR100

Figure 5: Robustness evaluation against multiple attack strengths.

The convergence of the algorithm. During the training, we observed that while adversarial examples distribute inside/outside the hard ball ϵ differently (i.e., as shown in Figure 2a), but generally the average distance to original input is less than ϵ. Therefore, according to the update formulation in Eq. (19), λ tends to decrease to 0 and eventually is stable at 0 because of very small learning rate as shown in Figure 3d. In addition, we visualize the training progress as shown in Figure 6 to show the convergence of our method. It can be seen that, the error-rate reduces over training progress and converges at the end of the training progress.



(a) MNIST (b) CIFAR10 (c) CIFAR100

Figure 6: Training progress of our UDR-PGD on different datasets, evaluating on the full training set (e.g., 50k images) and the full testing set (e.g., 10k images). Robust accuracy is against PGD attack with $k = 20$.

Further experiment result on CIFAR100. We would like to provide additional experiment result on CIFAR100 dataset such that all defenses are adversarially trained with $\epsilon = \frac{8}{255}$. Our UDR-PGD outperforms PGD 3.7% at $\epsilon = \frac{8}{255}$ and 2.3% on average, while our UDR-TRADES and UDR-MART outperform their counterparts by around 0.5% and 0.7%, respectively. It is worth noting that, in our experiment, MART is quite sensitive with changes of (MART's natural accuracy drops to a lower

Table 10: Robustness evaluation on CIFAR100 dataset. The last column "Avg" represents the average gap of robust accuracy between our methods and their standard AT counterparts.

	Nat	$\frac{8}{255}$	$\frac{10}{255}$	$\frac{12}{255}$	$\frac{14}{255}$	$\frac{16}{255}$	$\frac{20}{255}$	Avg
PGD-AT	63.7	22.8	16.1	11.4	7.8	5.1	2.4	-
UDR-PGD	64.5	26.5	18.9	13.7	9.8	7.0	3.5	2.30
TRADES	60.2	30.3	24.5	18.8	14.8	11.5	6.7	-
UDR-TRADES	60.1	30.8	25.1	19.3	15.5	12.2	7.5	0.52
MART	54.1	32.0	26.8	21.9	17.4	13.8	7.6	-
UDR-MART	54.4	32.3	27.4	22.5	18.4	14.4	8.5	0.67

Table 11: Distance function and its gradient

	$c_{\mathcal{X}}(x, x')$	$\nabla_{x'} c(x, x')$
L_1	$\sum_{i=1}^{d} \left\|x_i - x_i^{'}\right\|$	$1, \forall i \in [1, d]$
L_2	$\frac{1}{2}\sum_{i=1}^{d} (x_i - x_i')^2$	$\sum_{i=1}^{d}(x_i' - x_i)$
L_∞	$\max_i \left\|x_i - x_i^{'}\right\|$	$\begin{cases} 1, i = \text{argmax}_i \left\|x_i - x_i^{'}\right\| \\ 0, \text{otherwise} \end{cases}$

performance than that of TRADES); that might explain the lower gap between UDR-MART and MART with the new ϵ.

G Choosing The Cost Function

In this section, we provide the technical details of our learning algorithm in Section 4 in the main paper, especially, the important of choosing cost function $\hat{c}(x, x')$. Given the current model θ and the parameter λ, we find the adversarial examples by solving:

$$x^a = \text{argmax}_{x'} \left\{ g_\theta(x', x, y) - \lambda \hat{c}_{\mathcal{X}}\left(x', x\right) \right\}$$

We employ multiple gradient ascent update steps without projecting onto the hard ball B_ϵ. Specifically, the updated adversarial at step $t+1$ as follows:

$$x^{t+1} = x^t + \eta \left(\nabla_{x'} g_\theta(x', x, y) - \lambda \nabla_{x'} \hat{c}_{\mathcal{X}}\left(x', x\right) \right)$$

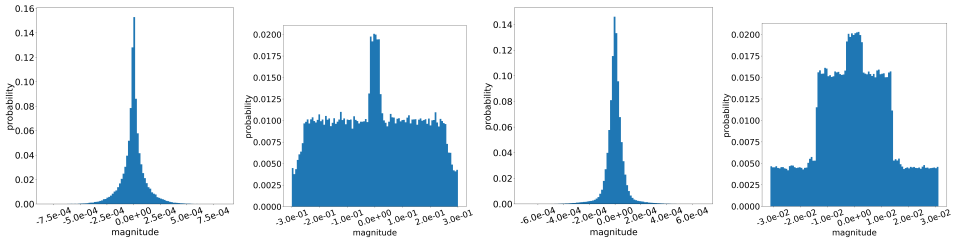
Given the smoothed cost function as in Equation (19), the updating step is as follows:

$$x^{t+1} = \begin{cases} x^t + \eta \left(\nabla_{x'} g_\theta(x', x, y) - \lambda \nabla_{x'} c_{\mathcal{X}}\left(x', x\right) \right), & \text{if } c_{\mathcal{X}}\left(x', x\right) < \epsilon \\ x^t + \eta \left(\nabla_{x'} g_\theta(x', x, y) - \frac{\lambda}{\tau} \nabla_{x'} c_{\mathcal{X}}\left(x', x\right) \right), & \text{otherwise.} \end{cases}$$

It shows that, the pixels that are out-of-perturbation ball B_ϵ will be traced back with a longer step, depending on the parameter τ. We consider three popular distance functions of $c_{\mathcal{X}}\left(x', x\right)$ with their gradient as Table 11. It is worth noting that, while the norm L_1, L_2 have gradient in all pixels, the L_∞ has gradient in only one pixel per image. It means that, when using L_∞ norm as the cost function $c_{\mathcal{X}}(x, x')$, only single pixel has been traced back at each iteration. In contrast, using L_2 will project all pixels toward the original input x with the step size of each. As in the discussion in Section F, only small part of an MNIST image contributes to the prediction, while in contrast, most of pixels of a CIFAR10 image affect to the prediction. Based on this observation, we use the L_∞ for the MNIST dataset and L_2 for the CIFAR10 dataset in the updating step. However, the perturbation strength ϵ has been measured in L_∞, therefore, we still use L_∞ in the Equation (22) to update λ.

We also visualize the histogram of gradient $\nabla_{x'} g_\theta(x', x, y)$ and $\nabla_{x'} \hat{c}_{\mathcal{X}}\left(x', x\right)$ as shown in Figure 7. It can be seen that the strength of gradient $grad1 = \nabla_{x'} g_\theta(x', x, y)$ is much smaller than $grad2 = \nabla_{x'} \hat{c}_{\mathcal{X}}\left(x', x\right)$, for example, on the MNIST dataset, $grad1 \in [-5 \times 10^{-4}, 5 \times 10^{-4}]$ while $grad2 \in [-0.3, 0.3]$ which is 600 times larger. Therefore, if using single update step, the gradient $\nabla_{x'} \hat{c}_{\mathcal{X}}\left(x', x\right)$ dominates the other and pulls the adversarial examples close to the natural input. These adversarial examples are weaker and do not helps to improve the robustness. Alternatively,

we break single update step for solving Equation (21) to two sub-steps as shown in Algorithm 1 to balance between push/pull steps. It also can be seen that the $grad2$ corresponds with the perturbation boundary ϵ and the step size η. For example, on the MNIST dataset, $grad2$ has the range from $[-0.3, 0.3]$ and has the highest density around $[-0.01, 0.01]$ where $\{0.3, 0.01\}$ are the perturbation boundary and step size in the experiment.



(a) CIFAR10, $grad1$ (b) MNIST, $grad2$ (c) CIFAR10, $grad1$ (d) CIFAR10, $grad2$

Figure 7: Histogram of gradient strength of $grad1 = \nabla_{x'} g_\theta(x', x, y)$ and $grad2 = \nabla_{x'} \hat{c}_{\mathcal{X}}(x', x)$ on MNIST and CIFAR10 dataset. We use L_2 norm for the cost function $c_{\mathcal{X}}(x', x)$, $\tau = \eta$ and $\lambda = 1$

5.4 Concluding Remarks

In this chapter, we have presented our contributions towards improving adversarial robustness through the lens of distributional robustness as introduced in Bui et al. (2022). In particular, in term of theory, we found that the existing AT methods can be unified under the framework of Wasserstein distributional robustness. We proposed a unified formulation of the risk function in WDR, with which, we could generalize and encompass the existing AT methods including SOTA ones in the distribution robustness setting. In term of practice, we proposed a soft ball projection method which adaptively adjusts the radius of the ball to the data distribution. Our method is simple yet effective, and it can be easily incorporated into existing AT methods to improve their robustness.

Chapter 6

Conclusion

Throughout this thesis, three defense strategies have been presented to enhance adversarial robustness of deep neural networks against adversarial attacks. In this final chapter, we summarize the key contributions made in this thesis and acknowledge certain limitations of our work. Additionally, we provide recommendations for exciting avenues that can be explored in future research.

6.1 Contributions

This thesis has made the following contributions to the field of adversarial machine learning.

In Chapter 3, we proposed novel adversarial training frameworks from the perspective of representation learning. We first introduced ADR (Bui et al., 2020) which imposes local and global compactness in the latent space. By enforcing local compactness, the representations of benign and adversarial examples are encouraged to be close to each other. Moreover, when combined with global compactness, the obtained latent space is more discriminative, where representations from the same class close and those from different classes distant. We demonstrated that ADR can be used as a general framework that improves the robustness of existing adversarial training methods.

In the subsequent work, we proposed ASCL (Bui et al., 2021a) which employs the contrastive learning principle to refine the robust representation. ASCL aims to minimize

the relative divergence between the latent representations of benign and adversarial examples instead of the absolute divergence as in ADR. We also developed strategies for selecting positive and negative samples that further enhance adversarial robustness. We demonstrated that ASCL outperforms several AT methods, including ADR, and achieves comparable performance to state-of-the-art methods.

In Chapter 4, we explored ensemble learning as a defense strategy against adversarial attacks. We first introduced CCE (Bui et al., 2021b) which is a novel collaboration strategy to enhance ensemble diversity, reducing the transferability of adversarial examples among the ensemble members. We used adversarial examples generated from one ensemble member to strengthen the robustness of this member, and simultaneously weaken the robustness of other members. We demonstrated that CCE effectively reduces the joint adversarial region of the ensemble, mitigates adversarial transferability, and improves the robustness of the ensemble.

Based on the insight that adversarial examples sampled from the joint adversarial region are crucial for improving the robustness of the ensemble, we proposed TAMOO (Bui et al., 2023) which is a multi-objective optimization framework for generating adversarial examples. We also introduced novel geometry-based regularization favoring unsuccessful tasks while maintaining successful ones. We conducted extensive experiments on different tasks to validate the effectiveness of TAMOO in generating stronger and more robust adversarial examples. More importantly, when combined with adversarial training, TAMOO significantly improves the robustness of the ensemble.

In Chapter 5, we proposed UDR (Bui et al., 2022) which is a powerful unified distributional robustness framework. We introduced a novel cost function based on the Wasserstein distance between the data distribution and its worst-case distribution, which allows us to generalize and encompass the existing AT methods including SOTA ones in our unified distribution robustness framework. Through extensive experiments, we demonstrated that with a better generalization capacity of distributional robustness, the resulting AT methods in our framework can achieve better adversarial robustness than their standard AT counterparts.

6.2 Limitations

General limitations. Firstly, we acknowledge that evaluating adversarial robustness is a challenging task, as discussed in Carlini et al. (2019). While we made efforts to ensure a fair and rigorous evaluation, there are still limitations that can be addressed. For instance, our evaluation focused on L_∞ adversarial examples, which is the most commonly studied setting in the literature since proposed in Goodfellow et al. (2015). However, research has shown that adversarial training with one norm does not generalize well to other norms, as noted by (Tramèr and Boneh, 2019). Therefore, it would be interesting to access the robustness of our methods against other L_p norms adversarial examples (Carlini and Wagner, 2017, Chen et al., 2018), multiple norms (Tran et al., 2022) or even Wasserstein ones (Wong et al., 2019b, Wu et al., 2020c).

Secondly, because of the limitation of our computational resources, as well as the common practice in the field, we evaluated our methods on small datasets including MNIST (LeCun et al., 1998), CIFAR-10/100 (Krizhevsky et al., 2009). While we attempted experiments on large scale datasets such as ImageNet (Deng et al., 2009), it was limited to adversarial attack experiments only. It would be valuable to investigate the performance of our methods on large-scale datasets in adversarial training experiments.

Although our methods are not limited to any specific architecture, we only utilized CNN architectures such as ResNet (He et al., 2016) and WideResNet (Zagoruyko and Komodakis, 2016) in our experiments. It would be interesting to explore how our methods perform on more recent architectures, such as Transformers (Vaswani et al., 2017, Dosovitskiy et al., 2021). Especially, recent studies have shown that adversarial examples crafted on CNNs and Transformers are less transferable to each other (Mahmood et al., 2021), which suggests an ensemble of principally different architectures might be more robust than an ensemble of the similar ones.

Limitations of the proposed methods. In addition to the general limitations mentioned above, each of our proposed methods has specific limitations.

ADR (Bui et al., 2020) and ASCL (Bui et al., 2021a) rely on representation learning and have demonstrated effectiveness in improving adversarial robustness. However, they are currently limited to the supervised learning setting where labeled training data is

available. Recent studies have shown that incorporating large amounts of unlabeled data can significantly enhance adversarial robustness (Rebuffi et al., 2021, Wang et al., 2023). Additionally, while we trained the feature extractor and classifier jointly as a regularization term in adversarial training, it has been suggested that training them separately can yield better results, as indicated in the Supervised Contrastive Learning framework (Khosla et al., 2020).

Moreover, our methods have been limited to the contrastive learning principle, which is one of the most popular representation learning approaches. However, recent alternative contrastive learning approaches such as DINO (Caron et al., 2021) have shown promising results in representation learning. Especially, the latent space obtained from DINO has been demonstrated to be more discriminative than the one obtained from contrastive learning (Caron et al., 2021), which might be suitable for adversarial robustness. Therefore, it would be interesting to explore the effectiveness of our methods on the latent space obtained from DINO.

CCE (Bui et al., 2021b) and TAMOO (Bui et al., 2023) are based on the idea of ensemble learning. However, we only evaluated on one type of ensemble mechanism, which is averaging the predictions of the ensemble members. It would be interesting to see how our methods can perform on other types of ensemble mechanisms such as voting (Zhou, 2012) or stacking (Wolpert, 1992). Furthermore, our methods primarily focused on the transferability of adversarial examples among the ensemble members to improve the robustness of the ensemble. However, it is important to consider the problem of non-transferable adversarial examples which can still successfully attack the ensemble.

UDR (Bui et al., 2022) is based on the idea of distributional robustness. While we have demonstrated its effectiveness in improving empirical adversarial robustness, we have not yet evaluated its effectiveness in enhancing certified robustness as in Sinha et al. (2017).

6.3 Future work

In addition to the aforementioned limitations, there are several compelling and promising directions for future research in adversarial machine learning that we would like to highlight.

Robust Architectures. The Forward-Forward algorithm proposed by Hinton (2022) presents a revolutionary learning procedure that deviates from the standard backpropagation algorithm. Exploring the robustness of the Forward-Forward algorithm from both attack and defense perspectives could be an intriguing area of research. Most existing adversarial attacks are designed for neural networks trained with backpropagation, where gradient-based methods are used to generate adversarial examples. If the Forward-Forward algorithm can successfully replace backpropagation, it could be a transformative development in the field of adversarial machine learning.

Bayesian neural networks (BNNs) (MacKay, 1992, Neal, 2012) is another promising direction for achieving robustness. While BNNs have been studied for decades, recent advances in variational inference (Blundell et al., 2015, Gal and Ghahramani, 2016) have made BNNs more practical. Recent studies have shown that BNNs can be more robust to adversarial attacks than standard neural networks (Carbone et al., 2020). In principle, BNNs can not only provide a prediction but also a measure of uncertainty for its prediction. While adaptive white-box attackers can adjust their strategy based on the uncertainty output, existing adversarial attacks have not been yet successfully adapted to BNNs. We believe that our findings on the effectiveness of ensemble learning and multi-objective optimization can be applied to BNNs to improve their robustness.

Recently, Ma et al. (2023) introduced a novel architecture, which does not rely on the traditional convolution or attention layers. This unique approach treats images as sets of unorganized points and employs a clustering algorithm for feature extraction and prediction. While the robustness of this architecture to adversarial attacks remains untested, its unconventional design suggests the potential for increased resilience against existing attack methods.

More Effective Adversarial Examples. While adversarial examples pose significant risks to AI systems, the most effective attacks are still limited to the white-box setting. The black-box attacks which still require a large number of queries to be effective, are less concerned with real-world applications. While acknowledging well aware of adversarial attacks, many AI practitioners have not implemented proper defense mechanisms or regulations to prevent them. Controversially, one approach to raising awareness and encouraging defense against adversarial attacks is to develop more effective attack methods.

Robust Continual Learning. Continual learning (Parisi et al., 2019) is a machine learning setting where a model learns from a sequence of tasks without forgetting the previously learned tasks. While continual learning has received extensive attention, its robustness aspect has not been thoroughly explored. For example, it remains unclear whether adversarial examples from previous tasks can be transferred to the current task, or in other words, whether the adversarial vulnerability of the model is accumulated or changed over time. Preserving model robustness while learning new tasks is an open question in this area of research.

Leveraging Pretrained Models. Pretrained models such as BERT (Kenton and Toutanova, 2019) and GPT-3 (Brown et al., 2020) have demonstrated impressive capabilities in natural language processing. It has been shown that incorporating embedding of label information extracted by pretrained language models can improve the performance of semantic semantic segmentation or classification tasks (Ding et al., 2022, Dao et al., 2023). We believe that these label information with high semantic meaning can be leveraged as global features to improve the robustness of the model.

Robust Multimodal Models. Multimodal machine learning (Baltrusaitis et al., 2018) is an emerging field that aims to learn from multiple modalities such as text, image, video, audio, etc. In recent years, pretrained multimodal models such as CLIP (Radford et al., 2021) have demonstrated impressive capabilities in understanding multimodal data, benefiting many downstream tasks such as image classification, object detection but most notably, text-to-image generation such as ImageGen (Saharia et al., 2022), Stable Diffusion (Rombach et al., 2022), Dall-E (Ramesh et al., 2021). Recently, Schlarmann and Hein (2023), Dong et al. (2023), Bailey et al. (2023) have demonstrated that with simple gradient based attacks such as PGD (Madry et al., 2018), these multimodal models can be easily fooled by these adversarial examples, opening up a new avenue for adversarial machine learning research.

Bibliography

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. Deep learning techniques for medical image segmentation: Achievements and challenges. *Journal of digital imaging*, 32(4):582–596, 2019.

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *arXiv preprint arXiv:2001.05566*, 2020.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In

Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6199`.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6572`.

Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1378, 2017.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. *arXiv preprint arXiv:1806.02371*, 2018.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018.

Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 1–17. Springer, 2020a.

Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*, 2018.

Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pages 26693–26712. PMLR, 2022.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *International Conference on Learning Representations*, 2019a.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016a.

Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access*, 9:155161–155196, 2021.

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick Drew McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016b.

Shupeng Gui, Haotao Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. *Advances in Neural Information Processing Systems*, 32, 2019.

G S. Dhillon, K. Azizzadenesheli, Z C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. *arXiv preprint arXiv:1907.02044*, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *arXiv preprint arXiv:2003.01690*, 2020.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283, 2018.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33, 2020b.

Van-Anh Nguyen, Trung Le, Anh Tuan Bui, Thanh-Toan Do, and Dinh Phung. Optimal transport model distributional robustness. *arXiv preprint arXiv:2306.04178*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. *arXiv preprint arXiv:2302.04638*, 2023.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

Ajil Jalal, Andrew Ilyas, Constantinos Daskalakis, and Alexandros G Dimakis. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*, 2017.

David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019.

Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 480–491, 2019.

Anh Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving adversarial robustness by enforcing local and global compactness. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII*, pages 209–223. Springer, 2020.

Anh Bui, Trung Le, He Zhao, Paul Montague, Seyit Camtepe, and Dinh Phung. Understanding and achieving efficient robustness with adversarial supervised contrastive learning. *arXiv preprint arXiv:2101.10027*, 2021a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020a.

Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 372–387. IEEE, 2016c. doi: 10.1109/EuroSP.2016.36. URL `https://doi.org/10.1109/EuroSP.2016.36`.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *Advances in Neural Information Processing Systems*, pages 13824–13833, 2019.

A. Shafahi, M. Najibi, M A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L S. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3353–3364, 2019.

Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017.

Thilo Strauss, Markus Hanselmann, Andrej Junginger, and Holger Ulmer. Ensemble methods as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1709.03423*, 2017.

Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018a.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pages 4970–4979, 2019.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55 (1):119–139, 1997.

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

Sanjay Kariyappa and Moinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019.

Anh Tuan Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving ensemble robustness by collaboratively promoting and demoting adversarial robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6831–6839, 2021b.

Anh Tuan Bui, Trung Le, He Zhao, Quan Hung Tran, Paul Montague, and Dinh Phung. Generating adversarial examples with task oriented multi-objective optimization. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=2f81Q622ww`.

Matthew Staib and Stefanie Jegelka. Distributionally robust deep learning as a generalization of adversarial training. *NIPS workshop on Machine Learning and Computer Security*, 2017.

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Anh Tuan Bui, Trung Le, Quan Hung Tran, He Zhao, and Dinh Phung. A unified wasserstein distributional robustness framework for adversarial training. In *International Conference on Learning Representations*, 2022.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pages 5286–5295. PMLR, 2018.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018.

Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. *arXiv preprint arXiv:2302.06588*, 2023.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: tracing through training. In *International Conference on Machine Learning*, pages 8326–8335. PMLR, 2020.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Sai Vemprala, Aleksander Madry, and Ashish Kapoor. Unadversarial examples: Designing objects for robust vision. *Advances in Neural Information Processing Systems*, 34:15270–15284, 2021.

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1615–1631, 2018.

Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 159–172, 2018.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, pages 269–277, 2017.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004.

Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, 2005.

Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25, 2006.

Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. *Machine Learning*, 81:121–148, 2010.

Battista Biggio, Giorgio Fumera, and Fabio Roli. Multiple classifier systems for robust classifier design in adversarial environments. *International Journal of Machine Learning and Cybernetics*, 1:27–41, 2010.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndi'c, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at

test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

Pin-Yu Chen. Adversarial machine learning for good. Technical report, IBM Research, 2022.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1467–1474, 2012.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30, 2017.

Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*, pages 480–501. Springer, 2020.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018b.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Tuan Anh Nguyen and Anh Tuan Tran. Wanet-imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021.

Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11966–11976, 2021.

Jacob Dumford and Walter Scheirer. Backdooring convolutional neural networks via targeted weight perturbations. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2020.

Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13198–13207, 2020.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Jean-Baptiste Truong, Pratyush Maini, Robert J Walls, and Nicolas Papernot. Data-free model extraction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4771–4780, 2021.

Hengrui Jia, Christopher A Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. Entangled watermarks as a defense against model extraction. In *USENIX Security Symposium*, pages 1937–1954, 2021.

Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*, 2017a.

Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpretability via model extraction. *arXiv preprint arXiv:1706.09773*, 2017b.

Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021.

Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.

Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4: 61–84, 2017.

Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *arXiv preprint arXiv:2007.07646*, 2020.

Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *International Conference on Machine Learning*, pages 6808–6817. PMLR, 2019b.

Kaiwen Wu, Allen Wang, and Yaoliang Yu. Stronger and faster wasserstein adversarial attacks. In *International Conference on Machine Learning*, pages 10377–10387. PMLR, 2020c.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pages 1277–1294. IEEE, 2020b.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad

Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3478–3488, 2021.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zidek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. *Advances in neural information processing systems*, 16, 2003.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, pages 84–92. Springer, 2015.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Lilian Weng. Contrastive representation learning. *lilianweng.github.io*, May 2021. URL `https://lilianweng.github.io/posts/2021-05-31-contrastive/`.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020.

Jean-Antoine D'esid'eri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

Leonid V Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 2018.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Alfred Galichon. *Optimal transport methods in economics*. Princeton University Press, 2016.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Pablo D Fajgelbaum and Edouard Schaal. Optimal transport networks in spatial equilibrium. *Econometrica*, 88(4):1411–1452, 2020.

Steven Haker, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent. Optimal mass transport for registration and warping. *International Journal of computer vision*, 60:225–240, 2004.

Zhengyu Su, Yalin Wang, Rui Shi, Wei Zeng, Jian Sun, Feng Luo, and Xianfeng Gu. Optimal mass transport for shape matching and comparison. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2246–2259, 2015.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*, pages 435–446. Springer, 2012.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10285–10295, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A C. Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, pages 1542–1553. PMLR, 2020c.

Rui Gao, Xi Chen, and Anton J Kleywegt. Wasserstein distributionally robust optimization and variation regularization. *arXiv preprint arXiv:1712.06050*, 2017.

Jose Blanchet and Yang Kang. Semi-supervised learning based on distributionally robust optimization. *arXiv preprint arXiv:1702.08848*, 2017.

Ruidi Chen and Ioannis C Paschalidis. A robust learning approach for regression models based on distributionally robust optimization. *Journal of Machine Learning Research*, 19(13), 2018.

Insoon Yang. Wasserstein distributionally robust stochastic control: A data-driven approach. *IEEE Transactions on Automatic Control*, 2020.

Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. *arXiv preprint arXiv:1705.07815*, 2017.

John C Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses against mixture covariate shifts. *Under review*, 2019.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019.

Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. Wasserstein robust reinforcement learning. *arXiv preprint arXiv:1907.13196*, 2019.

Elena Smirnova, Elvis Dohmatob, and Jérémie Mary. Distributionally robust reinforcement learning. *arXiv preprint arXiv:1902.08708*, 2019.

Esther Derman and Shie Mannor. Distributional robustness and regularization in reinforcement learning. *arXiv preprint arXiv:2003.02894*, 2020.

Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.

Trung Le, Anh Tuan Bui, He Zhao, Paul Montague, Quan Tran, Dinh Phung, et al. On global-view based defense via adversarial attack and defense risk guaranteed bounds. In *International Conference on Artificial Intelligence and Statistics*, pages 11438–11460. PMLR, 2022.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In *Advances in Neural Information Processing Systems*, pages 1829–1839, 2019.

C. Xie, Y. Wu, L v d. Maaten, A L Yuille, and K. He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *arXiv preprint arXiv:2010.13337*, 2020.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *arXiv preprint arXiv:2006.07589*, 2020.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural networks*, 12 (10):1399–1404, 1999.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems*, 33:1633–1645, 2020.

Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. In *Advances in Neural Information Processing Systems*, pages 12861–12871, 2019.

Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: diversifying

vulnerabilities for enhanced robust generation of ensembles. *Advances in Neural Information Processing Systems*, 33:5505–5515, 2020.

Zhuolin Yang, Linyi Li, Xiaojun Xu, Shiliang Zuo, Qian Chen, Pan Zhou, Benjamin Rubinstein, Ce Zhang, and Bo Li. Trs: Transferability reduced ensemble via promoting gradient diversity and model smoothness. *Advances in Neural Information Processing Systems*, 34:17642–17655, 2021.

Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu. Understanding robust overfitting of adversarial training and beyond. In *International Conference on Machine Learning*, pages 25595–25610. PMLR, 2022.

Charles Jin and Martin Rinard. Manifold regularization for adversarial robustness. *stat*, 1050:9, 2020.

Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*, 2021.

Bin Zhu, Zhaoquan Gu, Le Wang, Jinyin Chen, and Qi Xuan. Improving robustness of language models from a geometry-aware perspective. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3115–3125, 2022.

Yinpeng Dong, Zhijie Deng, Tianyu Pang, Jun Zhu, and Hang Su. Adversarial distributional training for robust deep learning. *Advances in Neural Information Processing Systems*, 33:8270–8283, 2020.

Nguyen-Duc Thanh, Le Trung, Zhao He, Cai Jianfei, and Phung Dinh. Particle-based adversarial local distribution regularization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

Hoang Phan, Trung Le, Trung Phung, Anh Tuan Bui, Nhat Ho, and Dinh Phung. Global-local regularization via distributional robustness. In *International Conference on Artificial Intelligence and Statistics*, pages 7644–7664. PMLR, 2023.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems*, pages 5858–5868, 2019.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Ngoc N Tran, Anh Tuan Bui, Dinh Phung, and Trung Le. Multiple perturbation attack: Attack pixelwise under different l_p norms for better adversarial performance. *arXiv preprint arXiv:2212.03069*, 2022.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7838–7847, 2021.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Ginevra Carbone, Matthew Wicker, Luca Laurenti, Andrea Patane, Luca Bortolussi, and Guido Sanguinetti. Robustness of bayesian neural networks to gradient-based attacks. *Advances in Neural Information Processing Systems*, 33:15602–15613, 2020.

Xu Ma, Yuqian Zhou, Huan Wang, Can Qin, Bin Sun, Chang Liu, and Yun Fu. Image as set of points, 2023.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022.

Son D Dao, Dat Huynh, He Zhao, Dinh Phung, and Jianfei Cai. Open-vocabulary multi-label image classification with pretrained vision-language model. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2135–2140. IEEE, 2023.

Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

Christian Schlarmann and Matthias Hein. On the adversarial robustness of multi-modal foundation models. *arXiv preprint arXiv:2308.10741*, 2023.

Yinpeng Dong, Huanran Chen, Jiawei Chen, Zhengwei Fang, Xiao Yang, Yichi Zhang, Yu Tian, Hang Su, and Jun Zhu. How robust is google's bard to adversarial image attacks? *arXiv preprint arXiv:2309.11751*, 2023.

Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons. Image hijacking: Adversarial images can control generative models at runtime. *arXiv preprint arXiv:2309.00236*, 2023.