It's a Wrap! Visualisations that Wrap Around Cylindrical, Toroidal, or Spherical Topologies

Kun-Ting Chen

August, 2022

MONASH University

It's a Wrap! Visualisations that Wrap Around Cylindrical, Toroidal, or Spherical Topologies

Kun-Ting Chen
Bachelor and Master of Science

A thesis submitted for the degree of *Doctor of Philosophy* at
Monash University in *2022*
Department of Human-Centred Computing
Faculty of Information Technology

Examiners	Prof. Stephen Kobourov
	Department of Computer Science
	University of Arizona
	Prof. Melanie Tory
	Data Visualization Research at the Roux Institute
	Northeastern University
Main Supervisor	Prof. Tim Dwyer
Secondary Supervisor	Prof. Kim Marriott
Associate Supervisor	Prof. Benjamin Bach

August, 2022

Kun-Ting Chen

It's a Wrap! Visualisations that Wrap Around Cylindrical, Toroidal, or Spherical Topologies

Doctor of Philosophy, August, 2022

Supervisors: Prof. Tim Dwyer, Prof. Kim Marriott and Prof. Benjamin Bach

Monash University

Data Visualisation & Immersive Analytics Lab

Department of Human-Centred Computing

Faculty of Information Technology

Wellington Road, Clayton, Victoria 3800, Australia

To my parents, Hong, and Elizabeth

Copyright notice

Abstract

Visualisation has become increasingly important in helping people understand and gain insights from data; and, ultimately, to make better decisions. Traditional visualisations are designed to be shown on a flat surface (screen or page) but most data is not "flat". For example, the surface of the earth exists on a sphere, however, when that surface is presented on a flat map, key information is hidden, such as geographic paths on the spherical surface being wrapped across the boundaries of the flat map. Similarly, cyclical time-series data has no beginning or end. When such cyclical data is presented on a traditional linear chart, the viewer needs to perceive continuity of the visualisation across the chart's boundaries. Mentally reconnecting the chart across such a boundary may induce additional cognitive load. More complex data such as a network diagram with hundreds or thousands of links between data points leads to a densely connected structure that is even less "flat" and needs to wrap around in multiple dimensions.

To improve the usability of these visualisations, this thesis explores a novel class of interactive wrapped data visualisations, i.e., visualisations that wrap around continuously when interactively panned on a two-dimensional (2D) projection of surfaces of 3D shapes, specifically, cylinder, torus, or sphere. We start with a systematic exploration of the design space of interactive wrapped visualisations, characterising the visualisations that help people understand the relationship within the data and benefit from being understood as 'cylindrical', 'toroidal', or 'spherical'. Subsequently, we investigate the design, development and implementation of a series of wrappable visualisations for cyclical time series, network, and geographic data. We show that these interactive visualisations better preserve the spatial relations in the case of geospatial data, and better reveal the data's underlying structure in the case of abstract data such as networks and cyclical time series. Furthermore, to assist future research and development, we contribute layout algorithms and toolkits to help create pannable wrapped visualisations.

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 information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

August, 2022

Kun-Ting Chen

Publications

Some of the contents and ideas presented in this thesis have appeared previously in the following publications:

Chapter 3: Design Space for Pannable Wrapped Data Visualisations

and

Chapter 4: Cylindrical Wrapping of Cyclic Time Series

[1] Kun-Ting Chen, Tim Dwyer, Benjamin Bach, and Kim Marriott. 2021. Rotate or Wrap? Interactive Visualisations of Cyclical Data on Cylindrical or Toroidal Topologies. IEEE Transactions on Visualisation and Computer Graphics 28, 1 (VIS 2021), 727–736

Chapter 5: Interactive Torus Wrapping for Network Visualisations - Small Networks

[2] Kun-Ting Chen, Tim Dwyer, Kim Marriott, and Benjamin Bach. 2020. Dough-Nets: Visualising Networks Using Torus Wrapping. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI 2020). 1–11.

Chapter 6: Interactive Torus Wrapping for Network Visualisations - Larger Networks

[3] Kun-Ting Chen, Tim Dwyer, Benjamin Bach, and Kim Marriott. 2021. It's a Wrap: Toroidal Wrapping of Network Visualisations Supports Cluster Understanding Tasks. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. (CHI 2021). 1–12.

Chapter 7: Spherical Wrapping for Geographical Data Visualisations

and

Chapter 8: Spherical and Toroidal Wrapping for Network Structured Data

[4] Kun-Ting Chen, Tim Dwyer, Yalong Yang, Benjamin Bach, and Kim Marriott. 2022. GAN'SDA Wrap: Geographic And Network Structured DAta on surfaces thatWrap around. To appear in Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. (CHI 2022).

Acknowledgement

I am grateful to my supervisors: Prof. Tim Dwyer, Prof. Kim Marriott, and Dr. Benjamin Bach who always actively participated in my PhD meeting and offered professional support throughout my doctoral research and publications. I would like to thank my collaborator Dr. Yalong Yang who worked with us during the final year of my PhD and offered his expertise in geographic map projections and relevant empirical studies.

I am grateful to my thesis panel members Dr. Michael Wybrow, Dr. Sarah Goodwin, and Dr. Jarrod Knibbe for valuable feedback of possible future directions beyond the work presented in this thesis, A/Prof. Bernhard Jenny for chairing the review of my doctoral work and offered valuable advice and discussion on geographic map projections and practical applications, Dr. Arnaud Prouzeau and Dr. Lonni Besançon for useful advice of statistical analysis.

I would like to thank Prof. Stephen Kobourov and Prof. Melanie Tory who reviewed this thesis and offered many in-depth insights and comments into improving this manuscript.

I would like to thank for A/Prof. Rashina Hoda, Prof. John Grundy, and Dr. Chakkrit Tantithamthavorn for giving me an awesome opportunity to disseminate my research with audiences from different background in Department of Software Systems and Cybersecurity (SSC) lunch and learning seminar. I would like to thank Monash FIT academic language specialist Julie Holden who supported me for academic writing tips, structuring this thesis, and how to disseminate my research through Monash FIT Three Minute Thesis (3MT) competitions. I am grateful to Monash graduate research and FIT administrative team, including Ammie Julai, Sunny Singh, Sidalavy Chaing, and all the fellow colleagues in Data Visualisation and Immersive Analytics (DVIA) group who offered feedback and supports for my research and studies, with special thanks going to fellow students Sarah Schöttler, Kashumi Madampe, Stanislav Pozdniakov, Chunlei Chang, and Ishwari Bhade who actively participated in numerous pilot studies and provided their professional feedback.

I would like to thank my parents, wife, and daughter for their wholehearted support during my PhD, in particular, my wife Elaine who pilots all of my studies and offers helpful advice on turning tough and boring user study materials into fun and easier

to understand ones, and my daughter Elizabeth who is always a great supporter for her dad.

Contents

List of Figures

Introduction

> *You can't do better design with a computer, but you can speed up your work enormously.*
>
> — **Wim Crouwel**
> (Graphic designer and typographer)

Data analysis is essential to help make informed decisions in a variety of areas such as finance, market analysis, medical sciences, social sciences, and engineering. Data visualisation is commonly used to aid a user in the exploration and the comprehension of such data through visual representations in various forms such as statistical graphics, charts, networks and maps; to name but a few [Mun14; Liu+14; War19].

Traditionally, these visualisations are laid out on a fixed-size flat rectangular surface designed for ease of reading on a printed page or standard screen, but many types of data are not "flat" in the sense that when the data is presented on a flat surface or screen some information must be lost. As a consequence, visualisations that are represented in this way could be hard to understand.

For example, while most people agree that the world is spherical. To show maps of the entire earth's surface on a screen we have to somehow cut, stretch and squash it. Cartographers and mathematicians have developed many methods to *project* the surface of a sphere to a plane (i.e., techniques that connect the data points and graphical elements on a 2D plane by mapping their spatial relationship on a 3D spherical surface [Sny97]). However, due to the nature of a sphere, none of these map projection techniques can be considered optimal in depicting geographic information of a 3D globe on a 2D flat surface. Similarly, none of these projections are free of distortion and present the data without potentially introducing artefacts. Therefore, some projections cut, stretch pieces of maps, glue, and present the entire world in an interrupted view to preserve certain area or distance properties such as myriahedral projections [Van08]; with several other methods (described in section 2.2) achieving different trade-offs between (e.g.) area, distance, or direction preservation and geographic path discontinuities [Sny87].

Distortion
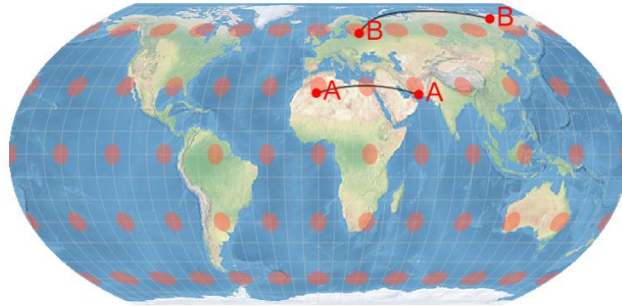
(a) Flat map - shape distortion

Visual cut
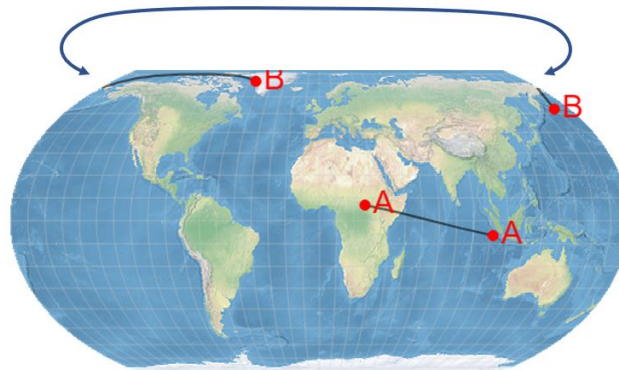
(b) Flat map - visual cut
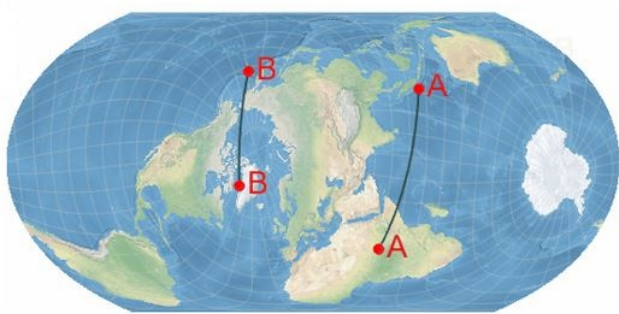
(c) Interactive spherical wrapping and rotation

Fig. 1.1.: Equal Earth map projection of pair of points (pair A and pair B). The black and curved line represents the geographical distance between each point pair on the surface of a globe. In such 2D flat maps, shapes are more distorted near the poles than the centre, as seen in the equal-sized orange dots (a). Path discontinuities (visual-cut) are inevitable in such maps (b). Interactive panning at the right allows a viewer to centre any region of interest for a less distorted view of the world. We discuss the effect of such interactive spherical wrapping on geographic comprehension tasks in chapter 7.

Fig. 1.2.: Bar chart and polar chart representation of hourly traffic flow over a 24-hour period in NYC in 2016 [Tro22]. To inspect whether the average hourly traffic accidents continue to increase or decrease from midnight till morning (upper-left), a user may need to mentally wrap the bars from 11pm at the right end across the page to 1am at the left end. Polar chart (lower-left) arranges the sequence of bars continuously, but certain sequences of bars (12pm to 5pm) are not aligned at the centre and thus a user may need to spin their head to make comparisons. Interactive wrapped and rotatable visualisations proposed in this thesis (upper-right, lower-right) allow a user to centre any group of bars while the visual cut is elsewhere in other hours (upper-right) and to have the same vertical baseline (lower-right) for comparing heights of bars. These images are created using our interactive tools described in chapter 3. We discuss the effect of such interactive cylindrical wrapping and rotation in chapter 4.

As a consequence, geographic distances and areas closer to the north and south poles may appear larger and the shapes more distorted than those at the centre [Jen+17] (Figure 1.1(a)). Furthermore, distances between geographic locations (e.g., Tokyo and Greenland) or areas of continents or seas that are wrapped across the projection boundary, or the direction of aeroplane's route across the boundary are also easily misinterpreted [Hen15; HAA16] (Figure 1.1(b)).

Most of these projections were originally developed for *static* display of the earth in printed maps and atlases. However, with modern computer graphics we can create interactive "wrapped" versions of these projections, such that they can be panned, reoriented and re-centred with simple mouse or touch drags [BD13; Dav13]. Such interactive panning allows the viewer to interactively rotate the 2D flat maps, i.e.

changing the orientation of the projection, and to wrap the projection around the 2D plane to centre a region of interest (and therefore the visual "cut" (Figure 1.1(b)) is elsewhere in other parts of the map), minimising distortion and discontinuities (Figure 1.1(c)). As we show in chapter 7, interactive panning of spherical maps provides promising results in terms of more accurate area, distance, and direction estimations than standard maps without such interaction.

Similarly, while often perceived as linear, many temporal phenomena have no beginning or end. Such data is intrinsically cyclical following cycles of day and night, seasons, or biorhythms. Common examples include time series of traffic flow over a twenty-four hour period; average temperature or birth rate over 12 months; or electrical and sound wave amplitude profiles. Non-temporal data can also be cyclical, such as average wind strength from different compass directions.

Often, cyclical data is presented in traditional linear bar and line charts. However, this ignores the cyclical nature of the underlying data and so understanding of cyclical phenomena in these charts may be hindered by "cuts" in the visualisation; i.e. the analyst needs to mentally "join" the left and right sides of the visualisation together, as seen in Figure 1.2-Top-Left. This example also shows that an arbitrary cut could potentially make it hard to understand trends (e.g., what happened after 11pm? Does the number of hourly traffic accidents continue to decrease after 11pm till midnight?) or patterns that wrap across the boundary of the chart as well as to compare bars that are far apart in the chart but whose data is temporally close, e.g. comparing hour 23 with hour 0 on a 24-hour chart (Figure 1.2-Top-Left) [TSA14].

One common solution to overcome this problem is to represent cyclical data in a polar (radial) representation. Like an analogue clock, polar visualisations show time in a circle, allowing for continuous representation of patterns and trends in any part of the cyclical data (Figure 1.2-Bottom-Left). The disadvantage is that comparing lengths of bars in a polar visualisation is less effective than comparing bar heights in traditional linear bar charts where the bars are bottom-aligned [Wal+19; AJB16; Bre+18].

To overcome the respective limitations of bar charts and polar charts, it is also possible for a "flat" or linear chart to be perceived as continuous when interactively connecting their left and right boundaries. That is, for linear charts, a novel "wrapped" panning interaction allows the viewer to pan the visualisations such that panning left causes marks which disappear from the left side to reappear on the right (and vice versa when panning right) (Figure 1.2-Top-Right). Thus, bars that might otherwise be at opposite ends of the chart can be brought closer together for viewing. Similarly, a novel "rotate" gesture can be applied to polar charts, which allows
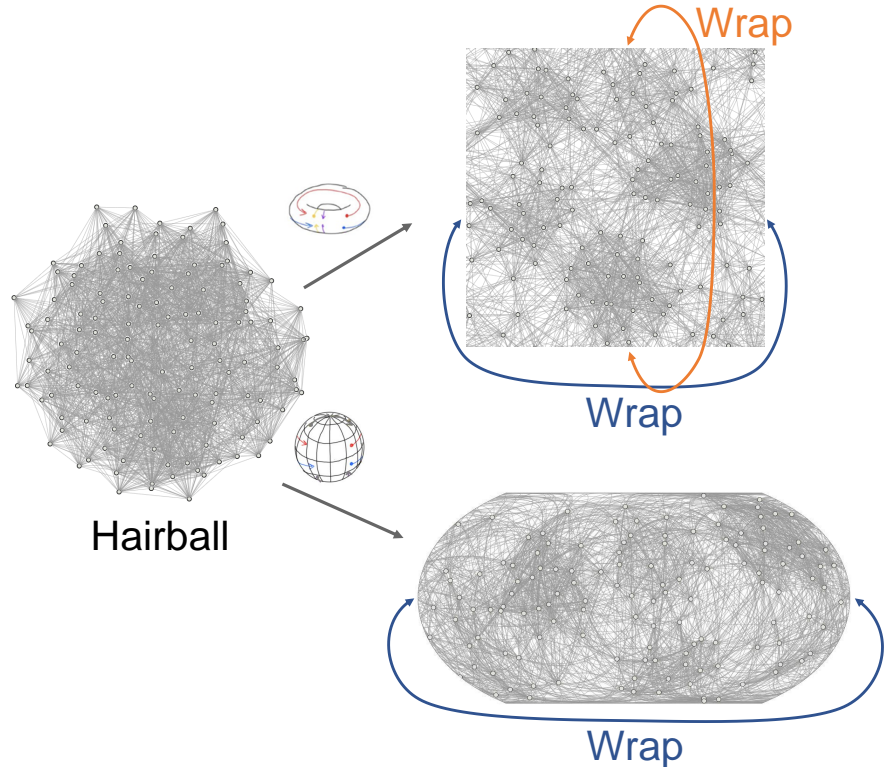
Fig. 1.3.: Node link representation of a densely connected network with a community structure. This network is created from public social network generators that simulate community structure. There are 128 nodes, 2090 links, and 6 clusters. The figures at upper-left, upper-right, bottom are laid out using a standard force-directed layout approach (discussed in chapter 6, torus-based (upper-right) and spherical-based layouts (bottom).

the user to bring any bar to the top or centre bars for more aligned comparison (Figure 1.2-Bottom-Right).

The use of such wrapped panning interactions for polar and linear chart representations of cyclical data have not been previously studied. We investigate the effect of such novel wrapped panning on cyclical data analysis tasks in chapter 4. Such one-dimensional wrapping or rotation described above can be considered to exist in a cylindrical topology[1] (before being projected to the screen), since topologically data points continue to wrap at the sides or rotate along the circumference of a cylinder when panned.

---

[1] In this thesis, we use the term "topology"' somewhat informally to refer to a small set of topological manifolds which correspond to the surfaces of shapes in R^3 (cylinder, sphere and torus). Points on these surfaces can be mapped (projected) to bounded Euclidean spaces in R^2. However, the different shapes ("topologies"), have different connectivity at the boundaries of these projections. This connectivity is interesting to us, because it allows us to have interesting interactive "wrapping" behaviours in our projected visualisations.

For more complex abstract data (data without inherent geometry, i.e., no information about mapping datasets to spatial position is provided), such as networks, highly connected networks with hundreds or thousands of links between data points lead to a densely connected structure that is even less "flat" when visualised in network diagrams (described below) and needs to wrap around in multiple dimensions.

Node-link diagrams are the most common way of visualising network data (or graphs) such as social and trade networks, software architecture, and biological networks across many domains [Mun14; Yog+20; Bur+20]. They are traditionally laid out on a flat rectangular viewport corresponding to a printed page or computer display. Many networks are found to have a small-world property, i.e., the average shortest path length between nodes in a network is a few hops, defined by Watts and Strogatz [WS98], or a community structure, i.e., subsets of nodes are more tightly connected within the subsets than outside the subsets, defined by Newman [New06].

However, in these networks, it could become too complex to understand or perceive the data's underlying structures or patterns for larger and denser datasets, because as the number of nodes and especially the number of links increases, their node-link diagrams become cluttered [VW08; Mun14; Sak+14; Wan+15]. Highly connected network structures translate into diagrams with many links that tend to cross the diagram leading to a dense, cluttered "hairball" (Figure 1.3-upper left) [Yog+18; ZPG18; RB20; Bur+20; Yog+20]. Figure 1.3-upper left shows a typical dense network (128 nodes, 2090 links) with a community structure, which becomes difficult both to disambiguate individual links but also to discern the high-level structure of the network, for example, to make out local clusters (defined shortly) of high connectivity. In order to better reveal and perceive such data's high-level network structures, its network layouts need to wrap around in multiple dimensions.

Discerning high-level structure, such as network clusters, is an important task in many domains, e.g., community structure in social networks[Rot21; New06], or biological networks [GN02]. Network clusters are loosely defined as subsets of nodes within the graph that are more highly connected to each other than would be expected from chance connection of edges within a graph of that density. More formally, a clustered graph has disjoint sets of nodes with positive modularity, a metric due to Newman which directly measures the connectivity of given clusters compared to overall connectivity [New06]. Visual cluster analysis is an important application of network visualisation. For example, an analyst may need to visually verify whether the output of an automatic cluster labelling algorithm makes sense

with respect to the network structure. For this task, they want a layout algorithm that displays the connectivity structure of the network as clearly as possible.

However, visual clutter (due to the hairball effect described earlier) makes cluster disambiguation difficult with standard node-link diagrams. As we explore in this thesis, there can be advantages to laying out networks on the surface of a 3D geometry, such as a torus or sphere, and then projecting the surface onto a 2D plane for reading on a paper or flat screen, such that in the case of torus, it has extra flexibility in avoiding or reducing crossings between links (edges) by wrapping some of them horizontally and vertically around the boundary of the display to reduce visual clutter (Figure 2.4, detailed in subsection 2.1.5) [MT01; KK16], and in the case of a sphere, it provides a layout that has no arbitrary edge to the display or privileged centre, as the network layout could be centred at any node of interest (Figure 2.2, detailed in subsection 2.1.4), as demonstrated by Perry et al. [Per+20] and Rodighiero[Rod20].

In particular, node-link representations of highly connected network data (which is ubiquitous in the world around us, from protein-protein interaction (PPI) networks, to social, communication, trade or electrical networks) have no real "inside" or "outside". Arguably, they can be better distributed across the surface of a topology which "wraps around". Similar to geographic maps and cyclical data, the toroidal and spherical projections of networks onto a 2D plane result in network layouts that wrap around the plane's boundaries, that is, visualisations which when interactively panned (i.e. using simple mouse or touch drags) such that nodes or links disappear from one side of the display, these marks reappear on the opposite side. In Figure 1.3 which we described earlier, toroidal (Figure 1.3-upper right) and spherical (Figure 1.3-bottom) layouts using our algorithms presented in chapter 6 and chapter 8 made clusters more distinct and easier to identify than the flat node-link layout (Hairball), and thus supports cluster identification tasks.

1.1 Research Challenges

To summarise, in all of the example data types shown above—including spherical maps, cyclical quantitative data, and networks—the visualisations can be wrapped around for better understanding about the patterns, structures or relationships within the data. Other examples of visualisations that wrap around will be explored in chapter 3.

However, as we discuss in detail in chapter 2, despite widespread use of map projections, spherical network layouts in 3D or immersive environments (such as Virtual Reality) [Kwo+16], and work in graph theory and algorithms on how networks for certain classes of graphs, e.g., non-planar graphs which can be drawn on a toroidal surface without any link crossings [KK16], there has been no research exploring interactive wrapped visualisations on a more general and generalisable level, nor is there available software to create such visualisations for arbitrary data types. In addition, the evaluation of human readability studies of these types of displays has not been presented.

Thus, we argue that despite the multiple potential applications for interactive wrapped visualisations, prior to the work presented in this thesis there was a lack of research linking these concepts and evaluating this potential from a human-centred design point of view. This work aims to provide a unified view of interactive wrapped visualisations and proposes novel interactions (Figure 3.1) that allow people to explore them and more effectively perform data analysis tasks.

1.2   Research Goals

The overarching design of this research follows five research goals (RG). We outline the research goals. The specific research questions are described further in each respective chapter (chapter 4-chapter 8).

1.2.1   RG1: Design space for interactive wrapped data visualisations

We identify the types of real-world data whose visualisations can potentially benefit from being explored with one-dimensional and two-dimensional wrapping on a 2D plane of a 3D geometry such as a cylinder, torus or sphere. We describe a design space based on the concept of one-dimensional wrapping and two-dimensional wrapping, linked to 2D projections of three common 3D topologies: sphere, cylinder, and torus[2]. This design space aims to consider a variety of real-world data types whose wrappable visualisations potentially benefit from being understood as 'spherical',

---

[2]While there are other shapes with interesting topological connectivity, we confine our exploration here to these three (cylinder, sphere and torus) because of their obvious wrapping properties when mapped to a 2D plane. For example, Klein bottle or Möbius strips are not applicable to wrapped displays, as these surfaces are not orientable (the orientation is reversed when moved across the edge of the 2D plane unlike cylinder, torus, or sphere)

'cylindrical' or 'toroidal' if it helps understand connectivity and patterns of the original data in these visualisations.

RG1 is an exploration of the design space informed by a systematic literature review of related work as well as our own examples and exploration.

1.2.2 RG2: Web tool for creating wrappable visualisations

We design and develop new web tools capable of creating wrapped visualisations for arbitrary data types, which can be interactively panned on a 2D projections of cylindrical and toroidal topologies. We apply cylindrical and toroidal topologies to a common range of data visualisations that allow a user to create wrappable visualisations for data types that make the most sense to be wrapped in this way.

The research method for **RG2** is an experimental analysis method. We design and develop simple web tools that create arbitrary visualisations with wrapped interactive panning. We then analyse the types of real-world data wrappable visualisations that make the most sense.

1.2.3 RG3: Cylindrical wrapping of cyclical data

To test the feasibility of the proposed interactive wrapping, we first investigate whether interactive cylindrical wrapping and rotation of cyclical data, in particular, cyclical time series, represented in traditional linear bar charts and polar bar charts, provides benefits in terms of user task performance (e.g., task completion time and accuracy) and user reported feedback and preference, compared with standard non-wrapped linear bar charts and polar bar charts without such interaction.

The research method for **RG3** is a mixed analysis method (i.e., combining quantitative and qualitative data analysis methods, described below). We pre-register our research questions, research design and hypotheses in Open Science Foundations (OFS). User studies of visualisation tasks, including time, accuracy, and subjective user rank, provide us with quantitative data. We collect participants' post-study qualitative feedback. We then perform statistical tests such as parametric and non-parametric tests to evaluate the quantitative data and collate the qualitative feedback.

1.2.4 RG4: Torus wrapping for network visualisations

To further explore the potential benefits of interactive wrapped visualisation, we explore and develop network layout algorithms that take advantage of the extra flexibility in a torus topology. We explore how to optimise node positions while wrapping certain links around the plane's edges horizontally and vertically to reduce crossing links (edges) and better capture the underlying data's structures and relationships for general networks (i.e. networks not just restricted to a certain class such as non-planar networks). We explore how best we can visualise the layout of a node-link diagram on the surface of a torus on a printed page or 2D flat screen. We explore whether toroidal network visualisation provides benefits in terms of graph layout aesthetics compared with standard non-wrapped layouts. We also evaluate whether toroidal layouts are more effective than standard non-wrapped visualisations in terms of network visualisation tasks.

The research method for **RG4** includes an experimental analysis method for graph layout aesthetics and a mixed analysis method. For the former, we first use standard simulators to generate a variety of random abstract data (networks) that simulates real-world social and biological networks. We then evaluate the algorithms using performance indicators such as standard graph aesthetics that reflect readability of a given layout, as well as the algorithm's runtime performance. For the latter, the mixed analysis method is the same method as described in **RG3**.

1.2.5 RG5: Geographic and network structured data that wraps around

We investigate the effectiveness of introducing interactive panning, combining wrapping and spherical rotation (described earlier) to a range of static map projections. We evaluate perceptual benefits of interactive spherical wrapping and rotation against static representations for geographic comprehension tasks.

To study whether interactive panning on 2D spherical projections are also useful for network data, we explore how to develop and adapt spherical network layout methods to find node positions that take advantage of the spherical surface. We explore the best interactive spherical projection suitable for node-link visualisations. We compare perceptual benefits of spherical network layouts with non-wrapped flat arrangements, and 2D projection of other 3D topologies, such as a torus for cluster understanding and network exploration tasks.

The research method for **RG5** includes a mixed analysis method (as described in **RG4**) for maps and networks, and an experimental analysis method for evaluating the proposed method for improving spherical network layouts.

1.3 Contributions

In this thesis, we contribute a novel class of interactive wrapped data visualisations, i.e., visualisations that wrap around two-dimensional projections of cylindrical, toroidal, or spherical surfaces. Such pannable wrapping is widely applicable to many data types, in particular, cyclical data, geospatial data, and networks. We offer the first web tool capable of creating pannable wrapped visualisation for arbitrary data types, and the first general-purpose torus network layout algorithms that afford better aesthetics and are more task-effective than standard non-wrapped representations. We also contribute to an adaptation of existing spherical network layout algorithms for spherical wrapping.

We show that these interactive visualisations better preserve the spatial relations in the case of geospatial data, and better reveal the data's underlying structure in the case of abstract data such as networks and cyclical time series.

1.3.1 Design space for interactive wrapped visualisations

We contribute the first systematic exploration of the design space for interactive wrapped visualisations. Cylindrical topologies are well suited to data that is cyclical in one dimension. Toroidal topologies are useful for relational data and when the data has two cyclical dimensions. Spherical topologies are well suited to geographic data and relational data that can be arranged onto such a topology.

We focus exclusively on the 2D projections that can be created from these views. This topological understanding of wrapping visualisations provides a unified view of wrapped panning interactions (using mouse or touch drags) and suggests new wrapped interaction designs (e.g. Figure 3.1, Figure 3.9, and Figure 3.14). Our design space can be a powerful tool to: (a) understand relations between visualisations (e.g., spherical maps, linear and polar bar charts, and networks), (b) apply interactive wrapping to a range of visualisations, and (c) to create new visualisations by mapping existing visualisations onto cylinder and torus visualisations to obtain wrappable projections.

The exploration of this design space was published in [Che+21b].

1.3.2 Web tool for creating wrappable visualisations

To create wrappable visualisations, we contribute a small web tool that can present arbitrary 2D visualisations with one- or two-dimensional wrapped panning and allow for the selection of panning constraints in accord with the design dimensions, i.e., wrapping topology and projections. Web developers can prepare separate images such as their own static images of charts or other visuals for the visualisation, and optional axes-labels that remain fixed at the sides while the visualisation is being panned.

The web tool was published in [Che+21b].

1.3.3 Cylindrical wrapping of cyclic time series

We show that data that is cyclic in one dimension is suitable to be represented on a 2D plane with cylindrical wrapping. In a crowdsourced controlled experiment (defined in section 2.4) with 72 participants, our study offers the first evidence that pannable wrapped visualisation provides better task performance than standard non-wrapped representation for reading cyclic time series data.

Specifically, we find that bar charts with interactive cylindrical wrapping lead to less errors than standard bar charts for trend identification, and less errors than standard bar charts, standard polar charts, and interactive polar charts for pairwise value comparisons but slower than the standard charts.

The result of detailed evaluation was published in [Che+21b].

1.3.4 Torus wrapping for network visualisations

We offer the first general-purpose network layout method that we are aware of that is capable of laying out arbitrary networks on a 2D projected torus topology. Such topology permits additional spreading of node positions by wrapping certain links around the boundaries of the display to reduce visual clutter. We use this to find layouts affording better graph layout aesthetics in terms of more equal link (edge) length, fewer link crossings, less deviation of incident link angles than standard non-wrapped representations. These aesthetics metrics have been shown relative to

readability of network diagrams. In two controlled user studies with 48 participants, we find that while torus layouts impose significant cost in accuracy and speed for understanding link wrapping across the plane's edges, adding either additional replicated context (to provide full network connectivity on the edge of the display, defined in subsubsection 2.1.5), or interactive wrapping (panning interaction that allows for centring any region of interest) improve its performance to a point that it is comparable to standard non-wrapped layouts. The result of the algorithm and evaluation were published in [Che+20].

To further study realistically larger networks, we offer two new algorithms for improving toroidal layout that is completely autonomous and automatic panning the viewport to minimise the number of link wrappings across the boundary. The resulting layouts afford fewer crossings, lower "stress", and greater visual cluster separation (i.e. less visual cluster overlapping). In a study of 32 participants comparing performance in cluster understanding tasks, we find that toroidal visualisation offers significant benefits over standard unwrapped visualisation in terms of improvement in error by 62.7% and time by 32.3%. The results of the layout algorithms and evaluation were published in [Che+21a].

1.3.5 Geographic and Network Structured Data that Wraps Around

To our knowledge, we offer the first systematic evaluation of the effect of introducing interactive spherical wrapping on different geographic map projection techniques. We find that interaction overwhelmingly improves accuracy and subjective user preference compared to static projections across all projection methods and tasks considered, at the cost of increased time due to interaction. This suggests such interactive spherical wrapping should be routinely provided in online maps to alleviate misconceptions arising from distortions of map projection. The result of the evaluation was published in [Che+22].

For networks, we present the first evaluation comparing spherical network projections against toroidal and non-wrapped flat layouts. We contributed an adaptation of existing spherical network layout algorithms. To mitigate the link wrapping across the boundaries, we proposed algorithms to automatically centre the content of interest in the display. Our study confirms the benefits of topologically closed surfaces, such as the surface of a torus or sphere, when using node-link diagrams to investigate network structure. This finding suggests that interactive wrapping of networks arranged on 2D planes of 3D surfaces should be more commonly used for cluster analysis tasks and further, that toroidal layouts may be a good general

solution, being not only more accurate than standard non-wrapped flat layout for cluster tasks but also at least as accurate for path following tasks.

The result of the adapted layout algorithms and their evaluation were published in [Che+22].

1.4 Thesis Structure

We begin this thesis by reviewing the related research on data visualisation, revisiting existing data that wraps around topologies, network layout algorithms, and empirical evidence (chapter 2). Afterwards, we present a design space exploration which identifies various data types that wrapping visualisations makes sense, and our interactive web tools that create novel applications (chapter 3). We then discuss each of the wrapping topologies and effectiveness evaluation: cylindrical (chapter 4), toroidal (chapter 5 – smaller networks and chapter 6 – larger networks), and spherical topologies (chapter 7 – maps and chapter 8 – networks). We conclude this dissertation with a discussion of the benefits and limitations of interactive wrapping and provide our view on the future of pannable wrapped visualisations and the potential challenges that may be explored by future researchers in this area (chapter 9).

Background and Related Work

<div style="text-align:right">2</div>

> *A picture is worth a thousand words. An*
> *interface is worth a thousand pictures.*
>
> — **Ben Shneiderman**
> Distinguished Professor of Computer Science
> (University of Maryland) and influential
> Human Computer Interaction researcher

In this chapter, we examine traditional visualisation from the perspective of different data types that are laid out on a 2D rectangular plane, in particular, network visualisation (section 2.1), global map layout (section 2.2) and cyclical temporal data visualisation (section 2.3). We also summarise relevant empirical evidence of the effectiveness of these types of data visualisations. These are the data types that we argue can be interactively wrapped around a 2D display of 3D surfaces: cylinder, torus, or sphere, to better understand the relationships within the data. As we discuss each of these data types, where applicable we also look at existing work that has considered projections of these types of data from their arrangements on 3D sphere or toroidal topologies (defined in chapter 1) and the very limited existing consideration of interaction with these projections. In each case we discuss how this thesis relates and contributes to the existing literature.

2.1 Network Visualisation

Relationships between people or things can be modelled as a network of nodes (representing the things) and links between pairs of nodes (representing the relations between pairs of things). Examples include social networks, electrical networks, software dependency networks [Dwy+13], biological reaction networks [Tho+03], and so on. When it comes to visualising such networks, if the things represented by the nodes have no physical position in space (for example, software entities) or any physical position of those things is not relevant to the analysis, then we are free to

place the nodes anywhere in the display. That is, we are free to arrange the marks (e.g., nodes and links) corresponding to elements in the data in a way that is as readable as possible, in order to reveal important patterns or features.

We consider data structures, such as networks, that have no inherent spatial structure to be "abstract". The most common way to visualise networks are node-link diagrams where the nodes are drawn as boxes with meaningful labels and the links are drawn as lines between the nodes. If the aesthetic criteria for placing the nodes and routing the link lines can be precisely formulated then the problem of finding an aesthetically pleasing and readable layout can be considered as an optimisation problem. A wide variety of automatic algorithms for finding readable and aesthetic layouts for these diagrams have been developed along with an extensive underlying mathematical theory [Bat+98], such as layered layout by Sugiyama et al. [STT81], orthogonal layout by Batini et al. [BNT86; Kie+15], and many others [Bat+98].

2.1.1 Force-directed network layouts

Force-directed layout methods, which simulate a physical model of a network with springy links in the layouts, are the most widely used layout algorithm for general purposes in practical network visualisation applications [Mun14; Liu+14; BP08; Dwy09; Dev+19]. Such methods seek to optimise node-placement in node-link diagrams to support perception of clusters (as defined in chapter 1) [LS20; Suh+19; DMW08; Dwy+08], paths [ZPG18], and spatial distances between nodes [Dev+19; BP08].

One of the most popular methods following this approach is by Kamada and Kawai [KK+89] who simulate graph theoretic distance between nodes (computed by, e.g., all-pairs shortest path algorithms) in the layout by optimising energy functions where nodes with theoretic shorter path length are moved closer with a simulated spring systems and electrical forces, while nodes with larger shortest path length are placed further apart.

The precise goal of the optimisation and the quality of its solution determine how well we can perceive patterns in the resulting visualisation and understand the data. Unfortunately, force-directed placement is a non-trivial optimisation problem, i.e., it is extremely difficult to find a global optimum with respect to the layout goals, and so imperfect heuristics must be used and the results are often sub-optimal. Such optimisations do not guarantee to deliver optimal solutions that are free of ambiguity and present the data without potentially introducing artefacts [Bin+19; Mun14].

(a) Football-NoTorus   (b) Recipe-NoTorus
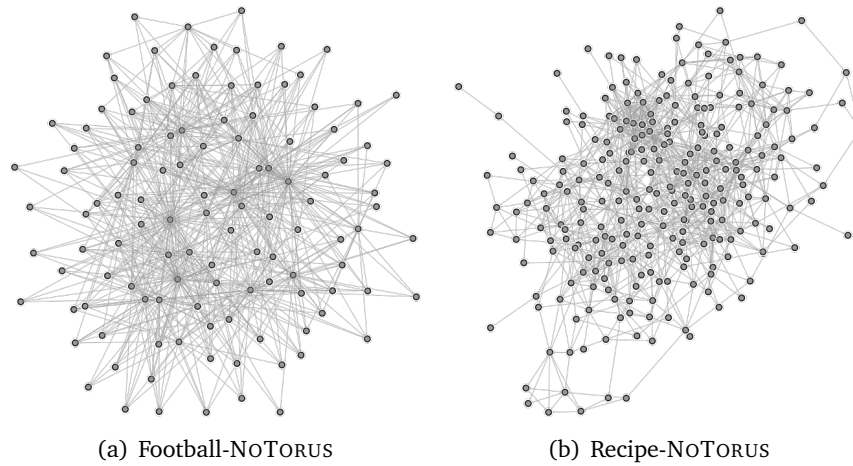
Fig. 2.1.: Node-link representations of (a) an American College football network (115 nodes, 669 links) with links representing regular-season games between Division IA college teams (nodes) they connect during Fall 2000 [GN02]. (b) a recipe network (258 nodes, 1090 links) with links representing pairs of ingredients (nodes) frequently used together for cooking [Ahn+11]. The hairball in (b) makes it difficult to identify the main clusters in the network, which may impede cluster understanding tasks.

While many of these methods can be applied to any type of networks and they typically improve the readability of network diagrams by reducing crossing links and node overlaps, there are certain types of dense network structures that can be particularly problematic for force-directed layout, producing messy "hair-ball" visualisations that are difficult to read [Mun14; Bur+20] (Figure 1.3-Left and Figure 2.1). Particularly problematic are:

- scale-free networks (where connectivity of nodes is distributed according to a power law [Tho+03], i.e. most of the nodes have low connectivity but a few nodes have high degree of connectivity) as defined by Barabási and Albert [BA99];

- small-world networks (most nodes are reachable by most other nodes with only a small number of hops) as defined by Watts and Strogatz [WS98]; or

- highly connected network structures such as tightly clustered nodes within subsets of nodes compared to their connectivity by chance in random networks, i.e., community structures defined by Newman [New06],

The approach to layout considered in this thesis is a variant of force-directed layout, but extended to perform the unravelling of the network on the surface of a wrap-around topology such as the sphere (subsection 2.1.4) and torus (subsection 2.1.5).

Performing layout on the surface of torus or sphere topologies potentially allows for better "unfolding" of dense networks, as seen in Figure 1.3-Right.

2.1.2 Clustered network layout

Since force-directed layouts of highly connected dense network structures tend to become difficult to disambiguate individual links and to discern the high-level structure of the network such as clusters, there are techniques which explicitly encode groups (subsets of nodes) or clustering information in visualisations, e.g. based on pre-identified clusters using community detection or clustering algorithms such as modularity clustering [New06]. These include methods displaying clusters in map-like layouts that highlight the visual cluster boundary, such as GMap by Gansner et al. [GHK09], Bubble Sets by Collins et al. [CPC09], and MapSets by Efrat et al. [Efr+14]. Similarly, to highlight common visual ambiguities such as visual overlap between community structures, an issue caused by the problematic hairball layout described in subsection 2.1.1, Wang et al. [Wan+15] encoded known clustering information in the visualisations to highlight visual cluster boundaries. Previous work also presented evaluations of task-performance for such group-level visualisations [Sak+14; OJK18; VBW17].

All of these algorithms rely on pre-identified cluster information—either from categorical variables within the data or pre-computed community detection—to highlight the cluster boundaries via layout or visual cues. However, for the layout approach considered in this thesis, we do *not* require knowledge of clusters and we do *not* specifically optimise the network layout based on clustering information. Instead, our approach, like multidimensional scaling (defined in subsection 2.1.8) and force-directed network layout more generally, implicitly groups nodes by minimising a cost function over difference between ideal graph theoretic distances and actual node separation in the 2D wrap-around drawings of projected 3D torus and sphere topologies [GKN04; Mun14]. This is different from a traditional non-wrapped node-link layouts. These topologies allow for extra flexibility in better distribution of node positions across the entire surface of a topology [Rod20; Per+20]; and reducing link and node overlap (in terms of torus, subsection 2.1.5) and eventually optimise graph readability.

2.1.3 Network layout with data aggregation

To improve the usability of node-link diagrams for large or dense networks, many data aggregation layout methods reasonably reduce visual clutter and reveal high-level patterns by showing an aggregated view of nodes and edges. Of course, they achieve this at the cost of reducing the level of detail present in the diagram, or requiring interaction to enable the user to recover such detail. Examples of such aggregated layout techniques include: link aggregation for creating high-quality layouts of network diagrams for software applications by Dwyer et al. [Dwy+13]; edge bundling which coalesced links with common direction (common start and end nodes similar to bundled train tracks) to reduce link clutter, such as confluent drawing by Bach et al. [Bac+16]; an interactive lens approach where edge bundling is based on selective areas of interest while the edges outside the lens are less bundled and retain the original link information, i.e., Moleview by Hurter et al. [HTE11]; and work that provides a clean overview of large networks for identifying and comparing key structural information about a network, in GraphThumbnail layouts by Yoghourdjian et al. [Yog+18]. However, many of these methods reduce the amount of information perceived by a user.

While such aggregation techniques may be compatible with the interactive wrapped visualisations introduced in chapter 3, the examples we explore present the full set of node-link data, taking advantage of the wrapping topology to reduce visual clutter.

2.1.4 Spherical network embedding

In previous subsections, we described how traditional network visualisations are laid out on a 2D flat screen to present a readable layout with meaningful patterns. Researchers have also explored visualisation of networks in three-dimensional views, with evidence that the third dimension can be used to improve readability of network clusters (Lu et al. [LS20]), or how to improve readability by removing link crossings for node-link diagrams [WM08; GPK11]. However, in such 3D representations occlusion is a significant problem [Alp+13], because with introduction of the third dimension, some nodes and links can be occluded completely by others facing a viewer.

Visualisation researchers have also explored the benefits of laying out networks on the surface of a sphere, or a torus (defined in subsection 2.1.5). However, in the case of laying out a network over the surface of a 3D globe, for instance, one hemisphere

is completely occluded by the other (Figure 2.2(b) and Figure 2.3-Left). A viewer needs to interactively turn to the opposite hemisphere to gain a full view of the entire network.

The potential benefit of laying out networks on the surface of a sphere or torus is that they are topologically closed surfaces: there is no centre or border to the surface, as the surface of a sphere (or torus) allows the layouts to be centred at any data point of interest [BM12; Per+20; Rod20] (Figure 2.2-c and Figure 2.3-Right). Thus, such spherical representation may allow the layout to better unravel the network and show its structure [BM12; Per+20; Rod20]. For example, Kobourov and Wampler [KW05] have investigated generalisation of force-directed algorithms to the surfaces of non-Euclidean (non-flat) geometries, e.g., the sphere; Perry et al. [Per+20] proposed a multidimensional scaling approach (defined in subsection 2.1.8) to produce drawings of networks on the surfaces of a sphere.

Such spherical network layouts are most commonly viewed in immersive virtual reality (e.g. [Kwo+16]) or as perspective projection of a 3D globe on a standard 2D monitor (e.g. [BM12; KL08]). The former provides the user with an immersive view from outside the sphere (*exo*-centric) or *within* the sphere (*ego*-centric). Ego-centric views are analogous to a planetarium, as users rotate their heads within the sphere, they see different parts of the network, effectively performing *panning* across its surface. Compared to more traditional exo-centric 3D views, ego-centric views have been found to be less efficient [Yan+18].

Note that in this thesis, we are **not considering direct 3D visualisation of surfaces of spherical and toroidal 3D shapes** (subsection 2.1.5). While such 3D visualisation may be interesting in immersive environments (for example, headset virtual reality with stereopsis and true perspective views which respond to head position), it is much less common for network layouts to be projected onto a 2D plane using a map projection due to the inherent problems in 3D with distortion and occlusion (detailed later, though some static map projections of graphs were demonstrated by [Rod20]). However, map projections have the great advantage over simple perspective rendering of a 3D globe, that the whole network can be seen at once (with Orthographic Hemisphere projection being the closest to a direct rendering of the 3D globe, but with both sides shown simultaneously). Furthermore, in this thesis, we focus on exploring the effectiveness and usability of projecting from such 3D shapes onto 2D wrapped display since we find that interactive 2D displays of such spherical layouts of data (whether geographic or network), and toroidal layouts of networks are not well studied.
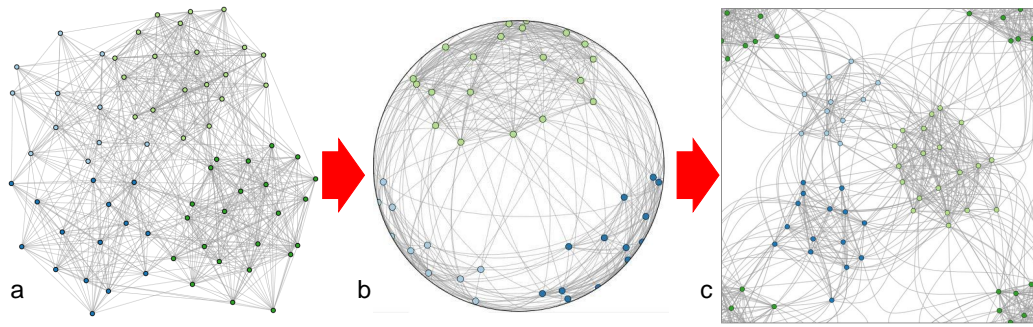
Fig. 2.2.: An example of a social network with community structures laid out using the traditional standard non-wrapped network layout (left); spherical network embedding on the surface of a 3D globe (middle) provides a better use of available screen space; the surface of a sphere projected onto a 2D plane to view the entire network using Charles Peirce's Quincuncial projection (right) [Pei79]. All of the network layouts are created using the layout algorithms described in chapter 8. The dataset is created using social network simulators by Fortunato et al. [For10]
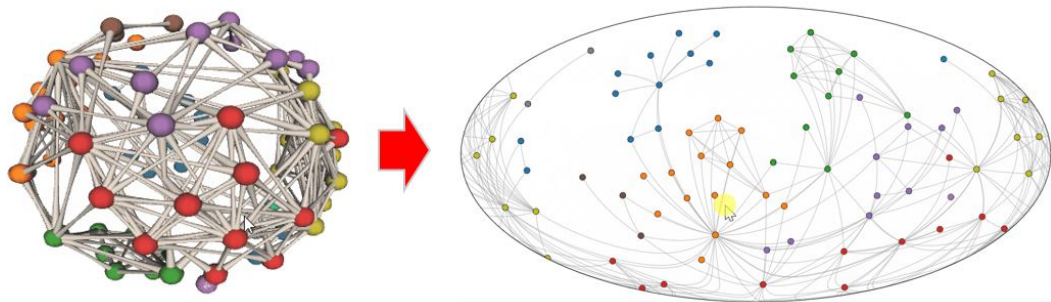
Fig. 2.3.: Spherical network layouts for a network of co-appearances of characters (nodes) in *Les Misérables* by Victor Hugo [Knu93], where edges indicate that two characters coexist in one scene and colours indicate group affiliation of the characters (nodes) in the novel; Spherical layout (left) developed using the methods described in chapter 8, and its 2D Hammer projection (right, described in section 2.2)

Likewise, projections from *inside* the 3D body (e.g., [Yan+18; Kwo+16]) are not considered here. Nor do we consider other exotic projections of 3D surfaces such as *Fish-eye projections* [Du+17] and *Hyperbolic* visualisations [LRP95; Mun98], which require continuous morphing as they pan.

Furthermore, despite the advantages of representing network layouts in 3D, visualisation design literature cautions against such use of 3D for visualisation if there are layout approaches better suited to the plane [Mun14, Ch. 6].

While one could simply use linear or perspective projection to project a 3D sphere (or torus, as described in subsection 2.1.5) onto the 2D plane, occlusion (defined earlier) and distortion of the surface are significant problems [Alp+13]. To minimise the problems of occlusion and perspective distortion when projecting the spherical

surface of the earth onto 2D maps, cartographers have developed map projections (defined in section 2.2) and visualisation researchers have applied such cartographic projection techniques to create 2D views of graphs on a sphere. Spherical projections, i.e., techniques that connect the data points and graphical elements on a 2D plane by mapping their spatial relationship on a 3D spherical surface [Sny97], have the great advantage over the aforementioned simple perspective rendering of a 3D globe, that the whole network can be seen at once while wrapping specific links around the plane's boundaries. For example, Orthographic Hemisphere projection (described in section 2.2) is the closest to such direct rendering of the 3D globe, but with both sides shown simultaneously for viewing the entire network while the nodes and links across boundaries wrap around.

While it is much less common for networks to be projected onto a 2D plane using a map projection, some static map projections of graphs were demonstrated. Examples include map style network drawing on a sphere, projected onto a plane where nodes and links wrap around by Perry et al. [Per+20]; spherical network embeddings with non-privileged views of social networks by Rodighiero [Rod20] (Figure 2.2(c)); and 2D projected sphere layouts using self-organising maps based approach (defined in subsection 2.1.9) by Wu and Takatsuka [WT06b], visualising multivariate networks (a type of network with multiple attributes associated with nodes and links such as social, cultural and spatial relationships between people in a social network).

The downside of such 2D projected spherical layout and map projections is that the resulting visualisation introduces discontinuities in the layout which now wraps around the boundaries of the plane, potentially confusing the viewer. In the case of networks, the connectivity of wrapped links in such displays (such as Figure 2.2-right and Figure 2.3-right) may not be obvious to all viewers, so the additional need for the viewer to understand link wrapping is the potential disadvantage of this style of visualisation [Hen15; HAA16]. In this thesis, we explore the use of rotate-able spherical layouts and evaluate their readability for network visualisation tasks. To the best of our knowledge, while interactive applications exist that allow the viewer to interactively pan to rotate a geographic globe projection, we do not know of previous research on interactive panning of spherical network diagrams when they have been projected onto a 2D plane using a map projection. However, as we show in this thesis, interactive wrapping provides benefits for perception for visualisation tasks. An exception is a web-based implementation of 2D rotate-able force-directed spherical layouts of networks by Manning [Man17]. However, in Manning's implementation the distance between nodes uses Euclidean distance not
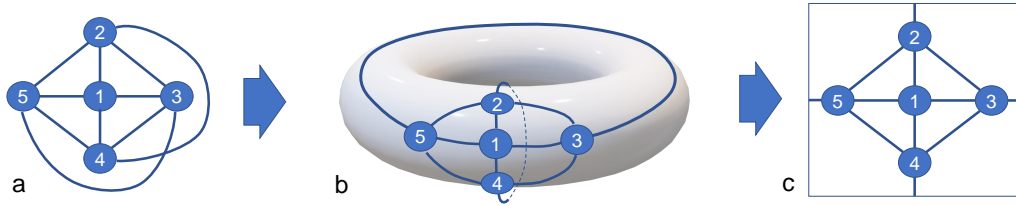
Fig. 2.4.: (a) A small complete graph K5,5 cannot be drawn on a 2D plane without line crossings. For example, there is an intersection of node pair (2, 4), and an intersection of (3, 5); (b) However, we can avoid such line crossings by routing the edge by connecting 2, 4 from top-bottom, and 3, 5 from left-right on a torus surface; (c) Cutting open the torus we can preserve the torus topology in a 2D plane. In chapter 5 and chapter 6 we propose torus-based network layout algorithms and evaluate their aesthetics compared with standard unwrapped representations.
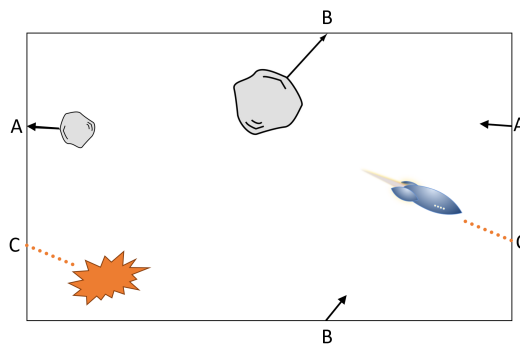
Fig. 2.5.: The video game "asteroids" is played on a flattened torus topology: here, we see that the path of the asteroid drifting to the left of the display will wrap around to the right at point A. Similarly, the asteroid drifting to the top will wrap around at B and the bullets of the space ship will wrap right-to-left at point C.

spherical great circle, which may not take full advantage of laying out a network on the surface of a sphere [Per+20]).

Beyond node-link diagrams, a range of other information visualisations can potentially be projected onto a sphere to minimise artefacts that occur when trying to find an optimal embedding for a 2D plane, discussed in our design space in subsection 3.2.1.

2.1.5 Toroidal network embedding

In addition to embeddings of graphs on the sphere, mathematicians and graph theoreticians have also considered embeddings of graphs on other topologies, such as the torus. For example, Mohar considers at length the theory of embedding graphs on various topological surfaces [MT01].

Mathematicians have long been interested in the difficult combinatorial problems associated with identifying the smallest number of link (edge) crossings required to embed a graph in the plane [Sch13]. Topological graph theory extends problems such as these into embeddings of graphs onto *higher-genus* surfaces. Higher-genus surfaces are essentially surfaces with holes which some of the links of a non-planar graph (i.e. the class of graphs that cannot be drawn in a plane without link crossings) can be routed through without intersecting other links [MT01]. For example, a spherical surface has genus zero (no holes), while a torus surface has genus one (possessing a single "hole"), and pretzels (double torus) have genus two, and so on. When a non-planar graph is drawn on the surface of a torus, it is possible to avoid or reduce crossings between links by routing some of them through the hole of the torus instead of around the outside, as shown in Figure 2.4(b). This figure also shows that it is possible to then "slice open" and flatten the surface of the torus so that it can be rendered on a screen or printed on paper, as seen in Figure 2.4(c). In such a flat rendering it is understood that edges which extend off the (for example) top, wrap around to the same horizontal position at the bottom of the drawing, and similarly for the left and right sides.

It would be fair to say that the majority of the work on higher-genus surface graph embedding has been theoretical or perhaps with application in areas such as circuit design. While some of this work has appeared in the Graph Drawing literature, which has cross-over between theory and network visualisation applications, graph drawing and visualisation researchers have not seriously considered practical network visualisation on higher-genus surfaces. Certainly, to the best of our knowledge, there was no available software that creates visualisations of arbitrary networks on higher-order surfaces. Algorithms capable of computing embeddings of certain types of graphs do exist [Yu14; DGK11], however, they are restricted to graphs that permit an embedding without crossings on the torus.

While it seems they are rarely actually implemented, nor used in data visualisation, there are very notable examples of classic 2D video games using torus wrapping in a way that players intuitively understand the topology. Consider the classic 1980s video game *Asteroids* [1] (Figure 2.5). In the Asteroids game, the video screen is like a box where players intuitively understand that an asteroid drifting off the left side of the TV screen box wraps around to the same vertical position at the right of the screen. Similarly, vertical wrapping occurs where the asteroid drifts off the top of the boundary and wraps around to the same horizontal position at the bottom of the screen. Such two-dimensional wrapping (wrapping vertically and horizontally) can be described as toroidal, since topologically (as defined in chapter 1) the data points

---

[1] https://en.wikipedia.org/wiki/Asteroids_(video_game)

are connected not only at the sides, but also at the top and bottom. In chapter 5 we explore whether the intuitive understanding of such torus wrap-around topology is also applicable to network visualisation.

The closest to network visualisation on higher-genus surfaces that we are aware of in the literature is illustrations in the kind of theory papers described above. That is, illustrations of the properties of embeddings on surfaces. 3D drawings of tori or higher-genus surfaces are sometimes used, i.e. doughnuts, pretzels and other holey pastries. While Kobourov and Wampler [KW05] have presented a "non-Euclidean spring embedder", i.e., a generalisation of force-directed algorithms for non-Euclidean surfaces, which may be capable of producing 3D torus drawings, it has not actually been tested with a torus distance metric. Also, visualisation in their work is generally intended for 3D rendering and does not achieve a 2D embedding. That is, it does not consider the question of how to project the torus surface back to a 2D display without occlusion.

An interesting research question may be whether such 3D representations might be usefully used in immersive environments, but this question is beyond the scope of this thesis. Rather, torus embeddings have the interesting property that they can also be represented in a two-dimensional diagram or visualisation, where it is understood that the left and right sides of the drawing connect, as do the top and bottom - as per the asteroids game mentioned earlier. In this thesis, our central interest is whether these types of 2D toroidal drawings permit improved graph layout aesthetics (described in subsection 2.1.6) and readability of the connectivity of the network, such that they can usefully be used in real-world network visualisation applications.

Beyond node-link visualisations, other examples of information visualisations can potentially be arranged onto a torus topology, discussed in our design space in section 3.4.

Tile-display and interactive wrapped panning

As already mentioned in relation to spherical wrapping subsection 2.1.4, one potential disadvantage of either toroidal or spherical wrapping is that this can break edges which now wrap horizontally and/or vertically around the layouts. To aid comprehension of wrapping around the border, in some illustrations of toroidal graph embeddings the layout is replicated on the edges of the original layout to provide partial or full context to the static diagram, such as networks by Kocay and Kreher [KK16] (Figure 2.6). This means the edges or maps are no longer
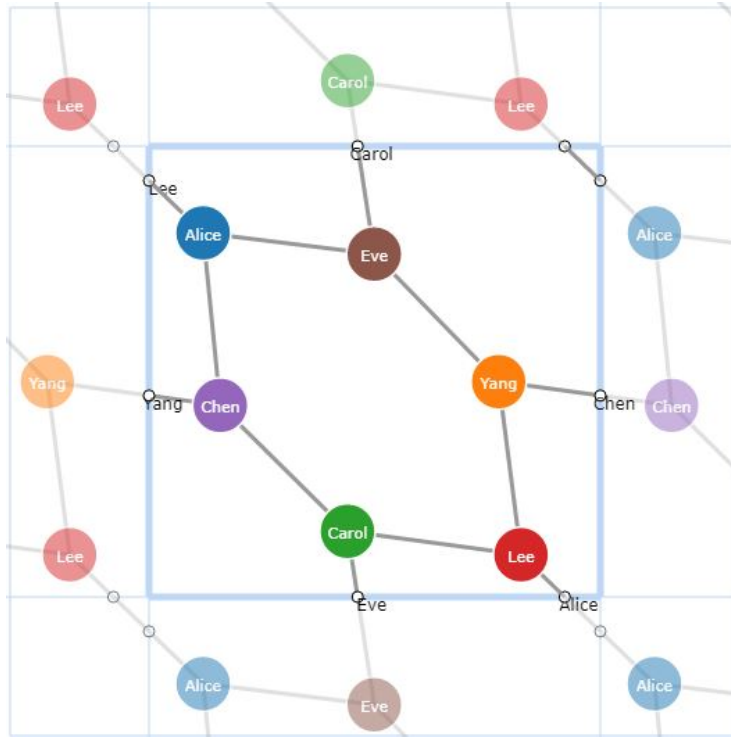
Fig. 2.6.: Partial context showing replicated network layout along the 2D plane's edges where a node-link diagram of K3,3 graph is laid-out on a 2D torus topology (Kocay and Kreher 2016), image reproduced using our torus-based layout algorithm described in chapter 5, where small circles intersecting the boundary shows which node (name) the node connects to. The layout is created using our algorithm in section 5.1.

broken at the boundaries, but at the cost of replication consuming additional screen space. Another example of similar tile display illustration has been presented in self-organising maps (defined in subsection 2.1.9) display of high-dimensional data by Ultsch [Ult03], which is explored in our design space in section 3.4.

Another potential aid to understanding edge wrapping is interactive panning. In map projections (section 2.2), interactive panning allows a user to use simple mouse or touch drags to pan the diagram to move any region of interest to the centre. Figure 3.2-top shows that when the data points are panned off the left side of the diagram, they reappear on the right side while the layout interactively wrap around in other regions, and vice versa [Dav13].

We might hypothesise that the user seeing the additional replicated wrapping context around the border in static diagrams aids comprehension. Similarly, we might also hypothesise that the user seeing the detail wrap around as they interactively pan will reinforce understanding. However, neither of these hypotheses have been tested before this thesis, which we explore in chapter 5. In chapter 3, we systematically

explore tiled representation and interaction as two wrapping approaches in our design space. This exploration leads us to create interesting wrapped visualisations for maps, cyclical data or relational data that may improve understanding of their data's relationships or connectivity.

2.1.6 Aesthetics and other empirical evidence on network visualisation

This thesis explores networks, an important type of abstract data commonly used in social science, medical science, and engineering. Effectiveness and usability are the key factors determining whether a given network layout and the patterns that it reveals can be of use for wider audiences and practitioners. The graph drawing community has a long history of establishing tools such as aesthetic metrics that measure the quality of network layout. These metrics give a quantitative model for aesthetic and readability requirements and have been correlated with user preference and performance in readability tasks. For example, fewer crossings between links and greater incidence angle between links where they connect to the same nodes have been shown to improve network path following tasks [HHE08; PCA02; Pur97; Yog+20]. Furthermore, networks with more uniform link length have been found to be preferred by readers [Dwy+09]. We demonstrate that such graph aesthetics measures can be improved by laying out networks on a 2D wrapped torus topology in chapter 5 and chapter 6.

Furthermore, unlike drawings of networks in a traditional non-wrapped 2D plane, our graph layout approach is based on 2D drawings of projected 3D topologies. To evaluate the readability of wrapped graph layouts, we define new metrics that capture the amount of link wrappings across the boundaries, detailed in section 5.2, and the distance between clusters of nodes in the network, measuring how well a layout "unfolds" a network diagram with dense structures, detailed in subsection 6.3.3.

Researchers have also explored readability of node-link diagrams across a variety of graph sizes and densities (e.g. Yoghourdjian et al. [Yog+20]), and with a variety of network comprehension tasks (e.g. by Saket et al. [Sak+14] and Okoe et al. [OJK18]). For example, Yoghourdjian et al. found that reading a node-link diagram is error-prone (more than 50% error rate) when graph density (ratio of edges to nodes) is equal to or greater than 6 for medium sized graphs (i.e., graphs with more than 50 nodes) [Yog+20]. Therefore, improving network layout quality and its usability remains an open question.

(a) Dense network   (b) Scale-free network
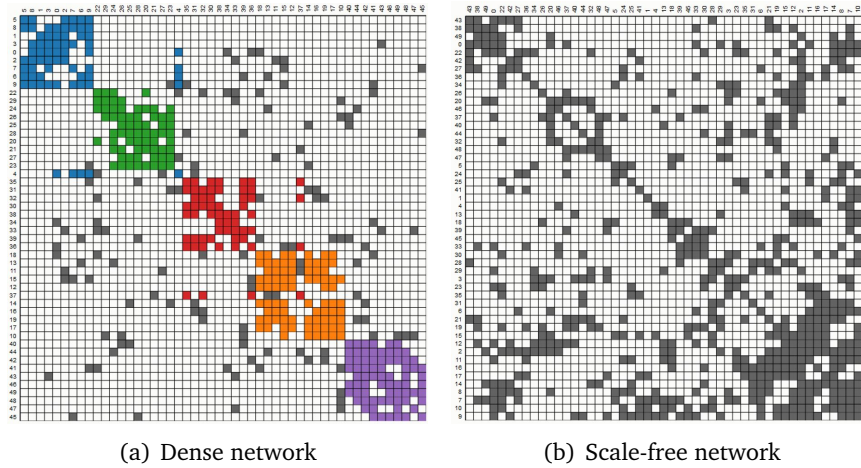
Fig. 2.7.: Adjacency representations of (a) a dense network with community structures generated using Fortunato et al.'s community network model [For10] (50 nodes, 211 links) and (b) a dense network generated using Barabási and Albert's scale-free network model [BA99] (50 nodes, 301 links); for a given dense network proper arrangement of rows and columns makes it easier to find that there are five main clusters (left) than a random order of rows/columns in the same network. Layouts were created by adapting existing library reorder.js by Fekete [Fek15].

2.1.7 Adjacency matrices

Since even the best layout algorithms must render very dense networks as hairballs (subsection 2.1.1), alternatives to node-link diagrams such as a tabular or adjacency matrix format can be used [Liu+14]. In an adjacency matrix, there is both a column and a row for each node, and the cells at the intersections of columns and rows of different nodes can hold a mark indicating the presence, absence, or weight of a link between those nodes. Such representations are capable of completely eliminating occlusion of node-link views for large and dense networks since there is a distinct cell for each possible link [Mun14; Liu+14].

In such tabular and matrix representations, there is also a significant degree of layout freedom, in terms of how rows and columns may be ordered. For example, if the rows and columns can be ordered such that the cells representing links involved in network clusters can be grouped together, then the cluster structure may be more evident than a random ordering [Beh+16; Bur+20]. Examples include determining such aesthetic orderings of rows and columns that reveal visual cluster patterns using "optimal leaf" ordering (Figure 2.7) or Cuthill Mckee ordering [Fek15].

While adjacency matrices have been shown more effective than node-link diagrams for identifying connectivity [OJK18; Bur+20], cluster identification tasks [OJK18], and weighted graph comparisons [Alp+13] for large and dense networks, adjacency

matrices suffer from the disadvantage that they are less familiar than node-link diagrams to most people and they can also be less intuitive to read [OJK18; Mun14].

In chapter 3, we demonstrate novel use of interactive wrapping for matrices. That is, we consider interactive column and row reordering for matrices that wrap around – again as if the matrix were projected onto 2D from the surface of either cylindrical or toroidal topologies.

2.1.8 Multidimensional scaling

Network layout algorithms have a strong connection with multidimensional scaling (MDS).

Many real-world data structures contain objects that have multiple attributes (e.g., a patient has temperature, weight, height, blood pressure, etc), however, when it comes to visualisation, people's ability to perceive such multidimensional data may be limited [BG05; KV07]. Therefore, in multidimensional data analysis, such data is commonly modelled using dimension reduction methods to preserve meaningful structures of a dataset when mapped to lower dimensions (e.g., 2D flat screens or 3D view) for ease of consumption by humans [IMO08; CC08; Bia+20]. Examples include MDS which models high-dimensional data as data points (representing objects) and the distances between pairs of data points (representing similarity or dissimilarity between pairs of objects) in low-dimensional space, and self-organising maps (defined in subsection 2.1.9). These techniques have been used in a variety of fields such as social sciences and biological sciences [CC08; Bia+20].

In MDS, the most common way of showing the resulting low-dimensional data points is scatterplots in 2D (or 3D) (i.e. X-Y or X-Y-Z dots plots). When visualising the resulting low-dimensional data points, similar to networks, if there is no physical positions that are relevant to the analysis of these data points, then it is considered "abstract data" and thus we have the freedom to lay out the data points to represent patterns or features corresponding to the high-dimensional data. One important goal of MDS is to preserve the high-dimensional data's underlying structure in low dimension as well as possible in order to faithfully represent the structures, relationships and patterns of the data [BG05; IMO08; DM09].

Examples of MDS plots that seek to arrange data to reveal structures, patterns or relations within the data include correlation between units sold and number of marketing promotions [CH18, Ch. 9], voters' similarity ratings on political candidates,

and embeddings of MDS plots for a better distribution of the datasets across the surface of a sphere by Lu et al. [LCY19].

Methods that seek to optimise the aforementioned arrangement of MDS plots typically solve a goal function called "stress", i.e. disparity of the summed distance between data points in low dimensions and the summed distance between original data points (defined more formally in chapter 5). These techniques solve exactly a variant of force-directed node-link layout problems (described in subsection 2.1.1) where the goal is to preserve the structure of the original data in the layouts. In fact, by specifying a complete distance matrix between all pairs of nodes of a network, the process of laying out node positions that simulates its graph theoretic shortest path distance is equivalent to the process of MDS finding suitable coordinates of data points in lower dimension that simulates its original high-dimensional distance [IMO08; Mar+12; ZPG18]. Similarly, by mapping high-dimensional distance onto weighted edges between all pairs of objects, the process of MDS is the same as laying out a weighted graph. However, as mentioned in subsection 2.1.1 and as we show in our evaluation (chapter 6), for larger and denser networks, such traditional force-directed based approaches do not do a good job of untangling a network.

In this thesis, like Gansner et al.'s stress majorization (i.e. an optimisation method for minimising stress) [GKN04], Martins' social network layouts using MDS-based approach [Mar+12], and Zheng et al.'s stochastic gradient descent (i.e. an optimisation method for minimising stress) for flat node-link layouts [ZPG18], our layout approach (chapter 5, chapter 6 and chapter 8) is also based on minimising this stress (defined earlier in this subsection) function, but on a more complex topology that provides us extra flexibility in laying out networks to reduce visual clutter.

MDS has also been embedded on the surface of a sphere to achieve a cleaner view of the complex data structure and better reveal clusters, by Lu et al. [LCY19]. However, the results were presented in a static 3D globe which may suffer from occlusion (defined in subsection 2.1.4) when viewed on a 2D flat screen or paper. Such spherical layouts of high-dimensional data have not been evaluated by human readability studies. In chapter 3, we explore examples of interactive wrapped MDS plots on a torus topology using our network layout approach discussed in chapter 6.

2.1.9 Self-organising maps

Self-organising maps (SOMs) are another example of arrangements of abstract data (defined in section 2.1) that also seek to optimise the layouts to better understand their relationships or the structure.

Introduced by Kohonen [Koh82], SOMs have been popular for social sciences, web-document mining, and other domains. SOM refers to dimension reduction techniques commonly used in multidimensional data analysis (described in subsection 2.1.8) in such a way that the original data's relationships and structure are preserved in lower dimensional projections (usually on a 2D hexagonal or rectangular grid) for ease of perception [KV07]. SOM models high-dimensional data as map units (representing objects) arranged in a 2D grid, where each unit is associated with a weight vector (e.g., with its cardinality corresponding to the number of dimensions in the original data). At each layout iteration, SOM distributes the positions of the map units by first selecting a best matching unit where its weight vector is closest to an input training vector of the original high-dimensional data. Then, the selected unit and its neighbouring units are moved such that the difference between the weight vector and the training vector decreases [Squ+05; WT06a].

Discerning high-level structure, such as clusters, is an important task for SOMs. In the process described above, map units that are grouped spatially nearby (into clusters) have more similar values than those in distal clusters [Koh82; WT06a; KV07]. Examples include studies showing SOMs revealing structural cluster changes and supporting visual comparisons tasks by Denny and Squire [Squ+05], studies showing participants distinguish clearly groups within the data using SOMs for a real-world dataset by Kirt and Language [KV07], and a qualitative study showing superiority of the SOM for visual financial performance analysis by Sarlin [Sar15].

SOM has also been adapted as a general force-directed (defined in section 2.1) layout approach for node-link visualisation, such as optimisation of the arrangements of data to reveal network clusters by Bonabeau, Eric and Hénaux [BH98].

SOMs have also been embedded on the surface of a sphere by Wu and Takatsuka [WT06a; WT06b] or laid out onto a torus topology by Ultsch [Ult03], as described in section 3.4.

However, self-organising maps-based node-link layout approaches belong to a general class of force-directed methods [BH98] and thus still suffer from cluttered layouts for larger and denser networks. In chapter 3, we explore examples of SOMs that are represented on topologies that wrap around.

3D globe   2D flat maps

unfolded
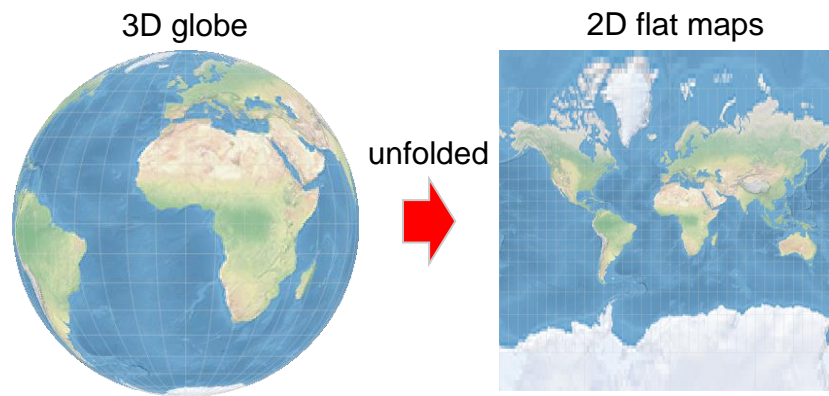
Fig. 2.8.: 3D globe and common 2D flat maps (e.g., Mercator projection commonly used in Google Maps).

2.2 Global Map Layout

While there has been limited research into wrapped network visualisations that are projected from 3D shapes onto a 2D plane (subsection 2.1.4 and subsection 2.1.5), creating 2D-projections from sphere typologies has been a challenge for map makers for centuries. Geographic information visualisation is the most common example involving visualisation of the 2D surface of a 3D shape. While sometimes geographic data is presented on a physical globe, cartographers have long ago found ways to project the surface of the globe to a 2D plane, e.g. for presentation on printed paper or more recently on computer screens. A plethora of methods for projecting a 3D globe onto a 2D map exist. Projecting the surface of a sphere onto a 2D plane is not novel: cartographers have been doing this for centuries.

Cartographers and mathematicians have devised hundreds of projection techniques [Sny97], including rectangular projections such as Mercator projection, commonly used in Google Maps, Quincuncial, Miller, Patterson projections, and non-rectangular projections, such as Hammer, Mollweide, and unusual projections such as Lee's conformal projection (i.e., map projections that preserve every angle between two curves that cross each other on the surface of a globe [Jen+17]) in a tetrahedron, with different mappings between data points across the map's boundaries. The reason for this diversity is that the surface of a sphere is "non-flat" and thus when geographic data is presented on a 2D flat screen, none of existing cartographic projection techniques can be considered optimal in depicting the original geographic information on the surface of a sphere [Sch+21]. Rather, each projection is a trade-off between preserving shape, area, angle, distance or direction [Sny97] as it is not possible for a 2D projection to do all of these simultaneously.
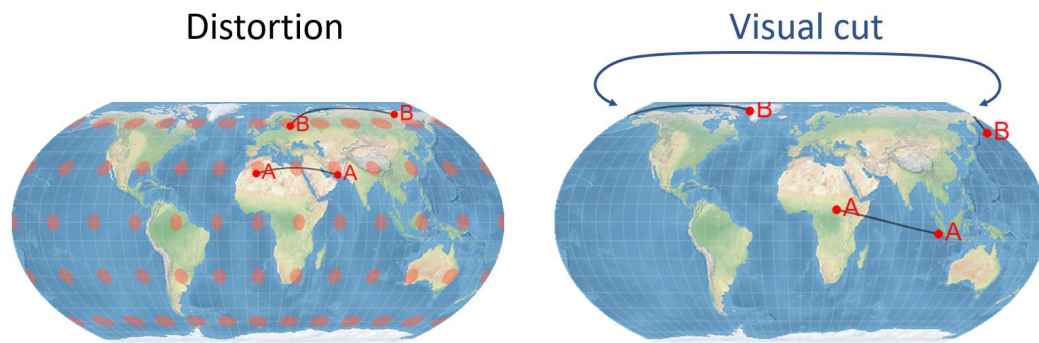
Fig. 2.9.: To judge which point pair represents the shorter true geographic distance on the surface of a globe, a user needs to take map distortion and discontinuities into account, which map may hinder geographic comprehension. Equal-sized dots on the left show the shapes are more distorted near the poles.

As a consequence, cartographers have invented what are called *equal area projections* that preserve the relative area of regions on the globe. These include EQUAL EARTH ⬭, HAMMER ⬭ and MOLLWEIDE HEMISPHERE ⬭ (see Figure 2.10) [ŠPJ19; Jen+17]. They have also invented *compromise projections* that do not preserve any of these criteria exactly but instead trade them off, creating a map that does not distort area, shape, distance or direction "too much." These include ORTHOGRAPHIC HEMISPHERE ⬭ and EQUIRECTANGULAR ▭ (see Figure 2.10).

To alleviate the strong distortions at the poles (top and bottom), non-rectangular projections have been developed, including *Fahey* projections (Figure 3.2-middle), Myriahedral projections [Van08], i.e. the globe is fitted to a polyhedron which, when cut open and unfolded along its edges, results in maps with many interruption but that are conformal (preserving angles) and conserve areas. Other examples include Dymaxion projections and many others, summarised by existing literature [Bos20; Sny97].

Globe projections also differ in the shape of the boundary of the projection. Some, such as EQUIRECTANGULAR ▭ are rectangular, others, such as EQUAL EARTH ⬭ or HAMMER ⬭, reduce distortion at the poles by projecting to a more ovoid shape. Some, such as the MOLLWEIDE HEMISPHERE ⬭ and the ORTHOGRAPHIC HEMISPHERE ⬭, resemble the front and back views of the 3D globe. Maps such as these in which the projection region is split are said to be *interrupted*. The ORTHOGRAPHIC HEMISPHERE ⬭, in particular, has a naturalistic appearance as it shows the Earth viewed from infinity [Jen+17].

There are several user studies on the readability and user preference of map projection visualisations [Hen15; HAA16; Šav+15; Car10]. For example, Hennerdal [Hen15] evaluated with five rectangular and non-rectangular projections,
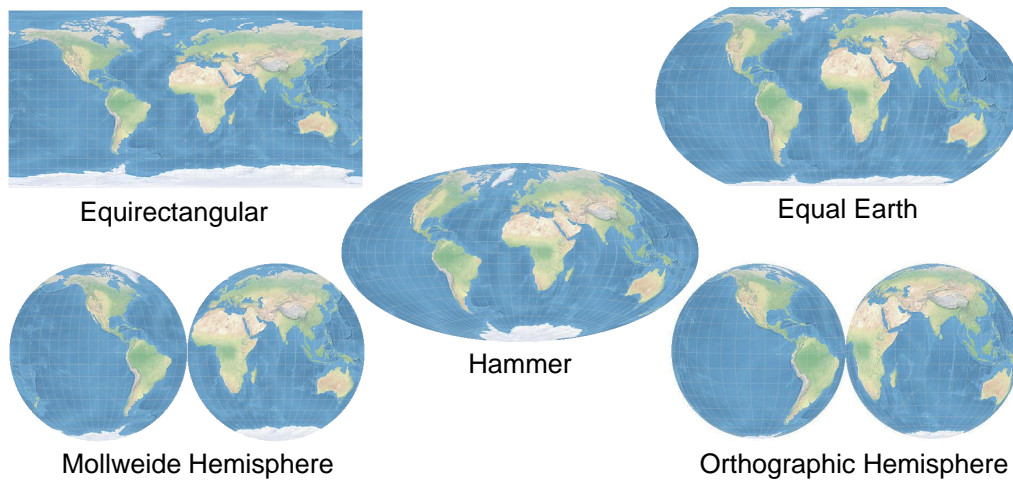
Fig. 2.10.: Five common map projections with a variety of area and/or distance preservation and shapes. The readability of these projections on geographic comprehension tasks are tested in chapter 7.

and found that people may find it difficult to understand path continuity across the edges of the map. Hruby et al. [HAA16] found that viewers find it difficult to understand the distance or direction between two points if this requires reasoning about the discontinuity in the projection and mentally wrapping the projection around a globe to understand their relative position. Avric et al. found users preferred continuous projection to interactive ones [Šav+15].

This suggests that allowing the viewer to interactively pan the map projection so as to centre a region of interest may improve their understanding of the inherent distortion introduced by the projection and of the Earth's underlying geography. For instance, this allows them to reposition two points so that they are no longer separated by a discontinuity. Such interactive panning, also called spherical rotation [Sny87], has been provided in many online maps for several years, e.g. by Bostock et al. [BD13], *Rotate the World* by Davies et al. [Dav13], and online interactive video maps showing how carbon dioxide travels around the globe over the course of one year by Jenny et al. [Jen+16].

One recent study of pannable terrain maps found that they perform more accurately than static maps but at the cost of additional time in task completion and concluded therefore that results of existing static map reading studies are likely not transferable to interactive maps [Her+18]. However, surprisingly, as far as we are aware there has been no evaluation of whether interactive panning of globe projections improves performance on standard geographical tasks such as estimating the distance or direction between two points or the relative area of two regions.

In chapter 7, we explore the effect of interactive panning on geographic comprehension tasks.

The only direct user research of pannable globes that we know of are two studies in virtual reality (VR) investigating different visualisations for understanding origin-destination flow between locations on the Earth's surface by Yang et al. [Yan+19]. While it was not their main focus, the studies revealed that interactive panning improved task accuracy at the cost of task completion time when viewing flow shown using straight lines on a flat map. However, it is likely that this was not because panning was used to reduce geographical distortion but rather that it was used to separate the flow lines which were the focus of the tasks. Furthermore, the static and interactive conditions were across different studies so comparison was between groups. In chapter 7, we present a more systematic and direct study of interactive panning for geographic tasks.

2.3 Visualising Cyclical Data

Cyclical data, such as: time series representations of average yearly rain-fall, monthly flu vaccines received by people, and average birth rate over months of a year; non-temporal cyclical data such as average wind strength around the compass directions; and so on, have no beginning or end. Such data are often represented in traditional static linear bar and line charts where the series is assumed to wrap around from one side to the other (e.g. January to December). However, this traditional "flat" representation may hide important information as it hides the nature of the underlying data's structure, making it difficult to inspect trend or value comparisons across the left-and-right edge of the chart where the data is actually continuous [TSA14]. Compared to linear bar charts, polar (or radial) visualisations have the advantage that cyclical data can be presented continuously without mentally bridging the visual 'cut' across the left-and-right boundaries (to be explored in this chapter). However, as we discuss in this subsection, in general they have been shown to be less effective than a linear bar chart.

Despite extensive consideration from designers of visualisation techniques, temporal data visualisation continues to pose challenges for effective representation. Temporal data can be very rich, including discrete events, duration, arrays of quantitative values (time-series), as well as more complex combinations with other types of data such as networks and spatial data (i.e. data combining spatial attributes such as locations), e.g. a geospatial map displaying hourly energy consumption in buildings
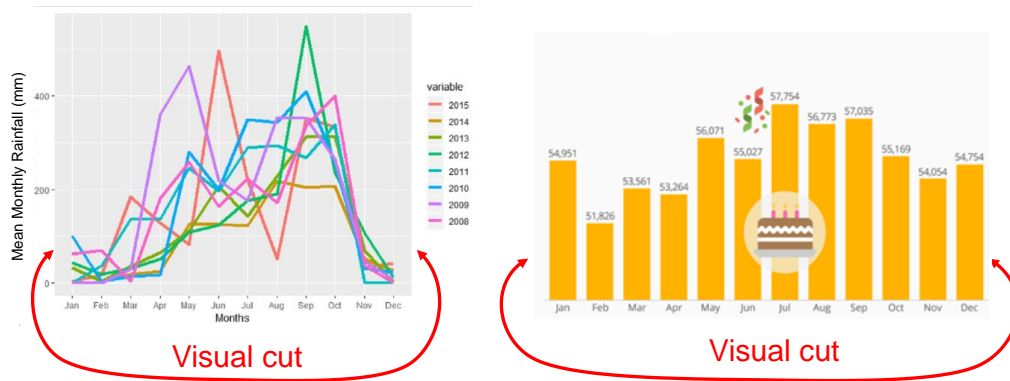
Fig. 2.11.: Average monthly rainfall over months of several years in Bangkok (image on the left adapted from Jain et al. [Jai+19]) and credited to Jain et al.; average UK birth rate over 12 months of a year (image on the right adapted from `https://www.statista.com/chart/5814/the-months-of-the-year-with-the-most-births/`.

within a city's downtown region by Zhang et al. [Zha+22]. Various surveys of temporal data visualisation exist, citing general techniques by Aigner et al. [Aig+11], timelines by Brehmer et al. [Bre+16], spatio-temporal data exploration techniques, for attributed trajectories by Tominski et al. [Tom+12], and more generally [AA06], and more abstract quantitative data that can be represented through space-time cubes (a three-dimensional (usually 3D Euclidean space consisting of a 2D geographical space and a time dimension) [Bac+17]. Interactive techniques for exploring general temporal data (without any particular allowance for cycles) range from multilevel zooming to scale to large data or minute detail [Zha+11] to sophisticated visual organisation of sequences of state data (such as brain activity) [Bac+15a].

Data for many temporal phenomena is characteristically cyclical, as nature follows the cycles of day and night, climate and seasonality, as well as biochemical processes that are continuous cycles. In the case of time series, polar visualisations have often been used either as simple cycles with radial bars or a line of varying distance from the centre. Multiple layers of bars or lines have also been proposed, e.g., silhouette graphs [Har99], overlaying multiple lines, e.g., one for every year, or a 'timeline' spiralling outside the centre of the polar chart [TS08]. A particular way to visualise cycles and repetition in temporal data is to abstract temporal change through multidimensional reduction methods [Bac+15b].

When comparing cyclical temporal data across 24 hours with static linear or polar bar charts, previous studies by Waldner et al. [Wal+19], Adnan et al. [AJB16], and Brehmer et al. [Bre+18] have shown that comparing lengths of bars in polar visualisation is less effective than comparing bar heights in traditional linear bar

charts. Likely (and in accordance with position encoding differences identified by Cleveland and McGill [CM84]), this is due to bars in the linear charts being parallel and aligned to a common baseline, as opposed to being at different angles and aligned to a circular base.

For example, Waldner et al. [Wal+19] found that for low-level tasks (e.g., locating extrema, reading values and comparing values at fixed 12-hour separation) a linear layout was significantly faster than a polar layout. Other studies on time series visualisation also found that for positional and length judgements, people make less errors using linear bar charts than when using polar charts for tasks of finding trends [AJB16], locating extrema [SED18; Wal+19], locating features at specific times [Wal+19; Bre+18], comparing values [Fuc+13], or proportion judgement [SH87]. However, there is no empirical evidence comparing the readability of cyclical temporal intervals split across the 'cut' on bar charts (Figure 1.2, Figure 2.11), compared with their corresponding polar bar charts showing the continuity. However, as we show in chapter 4, interactive wrapping of bar charts to centre any region of bars provides promising results in terms of more accurate trend identification and pairwise value comparisons compared with standard bar charts or polar charts.

2.4 Evaluation Methodology

In subsection 2.1.6, we described evaluation of network visualisations with established graph layout aesthetics metrics. This section describes other evaluation methods for visualisation performance discussed in this thesis.

User studies are the most commonly used evaluation method for data visualisations to gain insights from users, such as their preferences and reported qualitative experience on visual designs [Liu+14; Mun14]. They are also used to understand the limit of visual perception, cognition of graphical encoding, and visualisation performance (in terms of effectiveness and usability) of visualisation techniques or interactive graphical tools [Lam+11; Nob+20]. The above data is often collected in multiple ways, including questionnaires, surveys (quantitative or qualitative), controlled experiments, and interviews. In this thesis, we used mixed evaluation methods including questionnaires and controlled experiments. Controlled experiments are widely used to systematically test the effect of controlled conditions (also known as independent variables) on dependent variables. Examples of dependent variables, also known as uncontrolled variables, include user's task completion time, accuracy, number of mouse clicks, duration of mouse drags, and their gaze movement collected

from stationary or mobile eye trackers. A comprehensive review of experimental designs and practical visual examples has been presented by Purchase [Pur12].

There are two common ways for conducting controlled experiments: Laboratory-based or crowdsourced studies. Lab-based supervised studies typically involve experimenters supervising user studies in a quiet room with participants operating dedicated equipment in a controlled environment. Experimenters have full control over the course of experiments, ensure participants closely follow experimental procedures, and alleviate other confounding factors such as disruption perceived by participants during the experiments [Liu+14; Bor+18]. In section 5.5 and section 5.6 we present the user evaluation methods of lab-based studies. Supervised controlled experiments can also be done online, e.g., over a zoom session with participants sharing their computer screen operating dedicated experimental software. In section 6.4 we present an online supervised study.

Controlled experiments are also commonly conducted through crowdsourcing. It collects samples with a wilder demographic range than a lab-based study which sometimes helps improve generalisation of empirical results [Bor+18; OJK18; Nob+20]. However, in unsupervised studies, it often requires additional measures to ensure the quality of the data being collected, such as recruiting participants through established crowdsourcing platforms (Amazon Mechanical Turk [Pee+17], Prolific [PS18]) and testing the attention of participants conducting the study, to ensure experimental procedures are followed appropriately and that the data collected could be reliable. In section 4.2, section 7.2, and section 8.2 we present user evaluation results through crowdsourcing experiments. Gadiraju et al. [Gad+17] compared lab-based and crowdsourced studies, while Borgo et al. [Bor+18] surveyed visualisation studies in the past decade utilising crowdsourcing. They identified that more details need to be reported (such as within-subjects, where all the subjects are given the same sets of conditions and trials being evaluated, or between-subjects, where the comparison of conditions is between two different groups using different trials) that may affect reproduction and rigorousness of this type of study. Meanwhile, Nobre et al. [Nob+20] mentioned that crowdsourcing was less used for evaluating complex interactive visualisation techniques due to potential challenges of sufficient training with remote novice participants, and thus they proposed improved study designs with extensive and careful training which makes crowdsourcing complex and interactive visualisation techniques possible with novice participants.

In this thesis, we use crowdsourced studies in chapter 4, chapter 7, chapter 8; lab-based controlled experiments for chapter 5, and online supervised experiments for chapter 6 to evaluate human readability of interactive wrapped data visualisa-

tions. Other evaluation methodology such as experimental analysis (chapter 5 and chapter 6) can be found in the respective chapters.

2.5 Conclusion

In summary, we reviewed traditional visualisations from the perspective of different data types that are laid out on a 2D rectangular plane. When presenting such data with complex relationships (whether networks, high-dimensional data, geographic data or other data with cyclical relations), it can be difficult to place elements in such a way that all similar or connected elements are close together without creating excessive overlaps of network clusters (subsection 2.1.1) or loss of information when the data is split across the edges of the representation such as geographic maps (section 2.2) and cyclical time series data (section 2.3).

There can be advantages to 2D visualisation of 3D shapes. However, despite widespread use of projections in geovisualisation and the study of embeddings on spherical surfaces (subsection 2.1.4) and torus topology in graph theory (section 2.1), before the work presented in this thesis, there has been no research exploring interactive wrapped visualisations and evaluation from the perspective of human-centred design or human-computer interaction.

In this research, we aim to bridge this gap in literature by providing a unified view of interactive wrapped visualisations. In particular, we focus on 2D surfaces of 3D sphere, cylinder, and torus, because it allows us to have interesting interactive "wrapping" behaviours in our projected visualisations and improves data analysis tasks. To understand the concept of interactive wrapping and its potential applications we present a design space exploration in chapter 3, which unifies interactive wrapped visualisations and tools that allow a viewer to create them.

Design Space for Interactive
Wrapped Data Visualisation

<div style="text-align:right">3</div>

> *Q: What is a topologist? A: Someone who cannot distinguish between a doughnut and a coffee cup.*
>
> — **Paul Renteln and Alan Dundes**
> Paul Renteln is emeritus professor of physics at California State University; Alan Dundes is distinguished professor of anthropology at the University of California

Most data types are not intrinsically "flat", including spherical maps, cyclical quantitative data, and networks. By "not flat", we mean, informally, that visualisations of these data types can potentially be "wrapped around" the surface of different 3D shapes (cylinder, sphere, torus) and projected onto a 2D plane for better comprehension. By wrapping on a 2D plane, we refer to the fact that some visualisations can be perceived as continuous when connecting their left and right and/or their upper and lower boundaries (Figure 3.1).

To better understand the concept of wrapped visualisations and their multiple potential applications, this chapter first presents a systematic exploration of the design space for wrappable visualisations. The design space is informed by related work in the literature as well as our own examples and exploration.

We examine data types whose visualisations can potentially benefit from being laid out on 2D representations of a 3D shape, in particular cylinder, torus or sphere shapes. Naturally, spherical wrapping is well suited for geospatial data at a global scale (e.g., the world map) but may also be applicable to relational data types that have no spatial information that may be considered start and end points. Such relational data types include network structured data and self-organising maps [Rit99; Ito00; Ult03; WT06a] (Figure 3.1-sphere, detailed in section 3.2). Cylindrical wrapping is well suited to data that is cyclical in one dimension such that the representation benefits from panning continuously along that spatial dimension. Examples of such data include cyclical time series, but also directed networks with cycles, which can be

represented by Sankey diagrams[1] with cycles and metro maps with loops (Figure 3.1-cylinder, detailed in section 3.3). Toroidal wrapping is applicable when the data has two cyclical dimensions and for relational data that can be arranged freely onto such a topology for representation as node-link visualisations, adjacency matrices. High-dimensional quantitative data can also be mapped to a torus topology using multidimensional scaling (Figure 3.1-torus, detailed in section 3.4). This design space can be a powerful tool to: *(a)* understand relations between visualisations (e.g., linear and polar bar charts), *(b)* apply interactive wrapping to a range of visualisations, and *(c)* to create new visualisations by mapping existing visualisations onto sphere, cylinder and torus visualisations.

In section 3.1, we describe design dimensions that characterise visualisations that can be laid out on the surface of a 3D cylinder, torus or sphere shape. Besides geographic maps and network structured data that are suitable for spherical wrapping, we propose novel uses of cylindrical and toroidal wrapping for a common range of data visualisation techniques, including: bar charts; line charts; polar (radial) charts; networks represented by adjacency matrices or node-link visualisations; high-dimensional data visualised by self-organising maps and multidimensional scaling.

In section 3.5 we present new tools capable of creating wrapped visualisations for arbitrary data types, which can be interactively panned on a 2D surface with cylindrical or toroidal wrapping for arbitrary data visualisations.

3.1 Design Dimensions

In this section, we introduce a design space which maps the possibilities for 3D cylindrical, toroidal and spherical *topology* against the possibilities for mapping the topology to a two-dimensional visualisation (*projection*) and the (*wrapping*) affordances for showing how the visualisation wraps around.

Our design space is the composition of three dimensions:

---

[1]Sankey diagrams are a type of node-link diagram with directed links (arrows) used to depict flows between entities in which the width of the arrows is proportional to the flow rate

	3-dimensional Topology	Projection & Wrapping in 2D plane	Tile Repetition	Example Visualisation
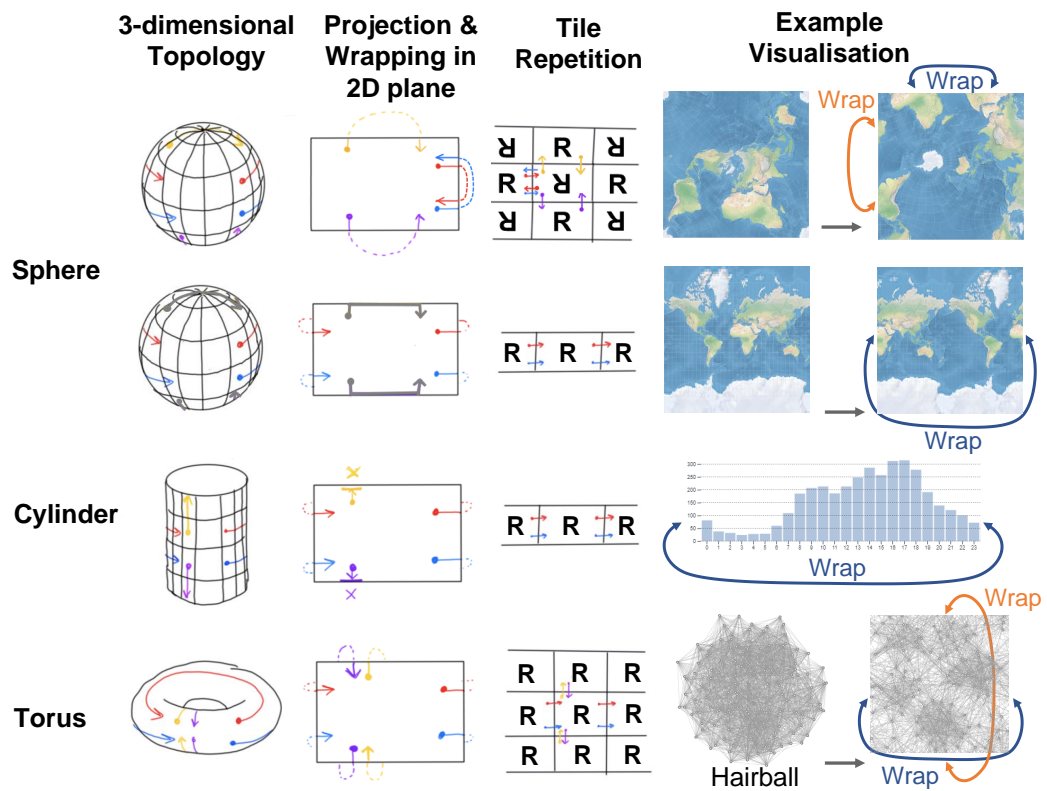
Fig. 3.1.: Design space of interactive wrapped data visualisations laid out on a 2D display of 3D cylinder, torus, and sphere geometries. The first column shows 3D topologies in our design space. The second column shows their resulting 2D rectangular projections. This also demonstrates how the points "wrap" around when panned. The third column shows tile replication which provides a continuous view of graphical elements wrapped across the boundary. The fourth column illustrates example visualisations of these data types applicable to each topology.

3.1.1 Topology

This dimension describes whether a visualisation (technique) is mapped onto the surface of a sphere (Figure 3.1-top), cylinder (Figure 3.1-centre) or torus (Figure 3.1-bottom). The first column in Figure 3.1 shows three-dimensional topologies. Different topologies provide different connectivity across their surfaces, and hence different connectivity of those surfaces when they are projected to the Euclidean plane [GAS17; Gar71]. We focus only on cylinder, sphere and torus topologies in this thesis because of their obvious "wrapping" behaviours when mapped to a 2D plane. They are called orientable surfaces because when data points are wrapped around the surface they maintain their orientation [GAS17; Gar71]. With non-orientable topologies such as Klein bottle or Möbius strips the orientation reverses as their surfaces are traversed. Such topologies are not applicable to wrapped displays as they would have undesirable wrapping behaviour, such as inverting the graphical elements across the edges of the projected view.

3.1.2 Projection

This dimension describes the projection method used to obtain a 2-dimensional representation from the 3-dimensional topology. The most common examples of 2D-projections from 3D sphere topologies were developed for creating flat maps of the spherical surface of the world. These include projections which result in rectangular 2D shapes, such as *Mercator* (Figure 2.8-right), *Equirectangular* projection (Figure 2.10) of a world map, *Miller* projection (Figure 3.5-top, an example of a spherical network layout), Pierceś Quincuncial projection [Pei79] (Figure 3.1-top row-example visualisation). They can also result in non-rectangular shapes, such as *Fahey* projection (Figure 3.2-middle), *Hammer* and *Orthographic Hemisphere* projections (Figure 2.10). In general, the different projection types strive to preserve different spatial aspects of the original sphere surface, such as area, distance or direction. For cylinder and torus topologies, we consider two projections motivated by our observation (detailed in subsection 3.3.1): the *side* and *top* of the 3D surfaces, respectively, which results in a rectangular (side projection, detailed in section 3.3) (Figure 3.9-centre) or a concentric radial (or "polar") (top projection, detailed in section 3.3) (Figure 3.9 and Figure 3.14).

3.1.3 Wrapping

Wrapping is the technique used to show how the projected visualisation wraps (or connects up) around its boundaries. We consider two approaches: (1) interactive wrapped panning (using mouse or touch drags) affordances for visualisations (which can continuously cover each of these topologies) when projected back onto a 2D plane; (2) tile-display, i.e., by repeating the 2D rectangular plane partially (Figure 2.6) or in full (Figure 3.3 and Figure 3.4) while preserving the view of linear point movement wrapped across a plane's edges.

The second column of Figure 3.1 demonstrates how wrapped panning differs across the topologies, using a set of coloured points (red, blue, orange, purple). That is, it demonstrates how these points would wrap "around" the borders of a 2D-plane if panned, as a result of projecting from the surface of the underlying topology. While horizontal pan (red, blue points) wrap the 2D rectangular plane horizontally, vertical pan (yellow, purple) only wraps vertically in torus and spherical projections, e.g., Pierceś Quincuncial projection (Figure 3.1-top row), as there are no data points wrapped (cut) across top or bottom boundaries in either cylindrical or other rectangular sphere projections, e.g., Mercator (Figure 3.1-second row). The third column of Figure 3.1 depicts how tile repetition differs across the topologies. While horizontal replication connects the graphical elements wrapped across the boundaries of horizontally adjacent tiles, vertical replication is only applicable to torus and spherical projections that are applicable to be tiled in vertical directions, e.g., Quincuncial projection (Figure 3.3).

3.1.4 Constrained wrapping

The key problem caused by projecting a 3D shape's surface onto a 2D display is the distortion (as seen in Figure 1.1) or discontinuity (i.e., visual "cut") created at the boundaries of the projection. Following features of the surface across the boundary may be assisted by interactive panning. That is, allowing the user of the visualisation to use a mouse or touch gesture to move (drag) and reorient centre projection such that features of interest are no longer split by the boundary. Depending upon the underlying topology of the 3D surface and the style of projection used, such wrapped panning interaction may require limitations (or constraints) on the panning direction. For example, wrapping direction is constrained naturally in cylindrical and spherical projections (Figure 3.2-second row and Figure 3.9-centre), while users can freely pan along both dimensions of the plane in torus topologies. To some extent, the constrained wrapping is also determined not only by the *topology*

and *projection* dimensions but also by the type of data and conventions within the application domain. We provide more details regarding the wrapping constraint in each respective topology later.

In the following sections (section 3.2, section 3.3 and section 3.4), we provide galleries of wrapped visualisations that emerge from three design dimensions described above. We discuss each of the three wrapping topologies and their projections in detail. We show how each topology can be applied to create wrappable visualisations in information visualisation and identify the types of data for which wrappable visualisations may provide benefits.

3.2   Spherical Wrapping

Designing effective 2D-projections from sphere geometries has been a challenge for map makers for centuries. As discussed in section 2.2, a plethora of projection techniques have been applied to **Spheres** by geographers, well summarised by Davies[2]. Spherical projections can be rectangular (Figure 3.2-top) or non-rectangular (Figure 3.2-middle and Figure 3.2-bottom), referred to in the related work in chapter 2.

In rectangular projections such as Pierce's Quincuncial projection [Pei79], data points can wrap around both horizontally and vertically. For example, in Figure 3.1-top row: example visualisation, the orange arrow shows that one tip of South America that wraps across the lower-left boundary is inverted at the corresponding upper-left position. Similarly, in the same visualisation, the blue arrow indicates that where one tip of Africa wraps around at the top-left boundary, it reappears and is inverted at the corresponding top-right position. The design of this inverted wrapping provides a seamless view in tiled repetitions (Figure 3.4 which reduces the discontinuities across the plane's edges [Pei79; Rod20; Hen15], discussed in **Tile-display**).

In other common rectangular spherical projections (e.g., Mercator), data points between the left and right boundaries are connected (red, blue in Figure 3.1-second row). Data points at the top (and bottom) of the 2D visualisation plane remain at the top while mirrored in a horizontal fashion (as shown in the grey arrow in Figure 3.1-second row). For example, in America centred world maps (Figure 3.2-top left), the continents of Asia wrap from the left end across the page to the right end.

---

[2]https://www.jasondavies.com/maps/transitions
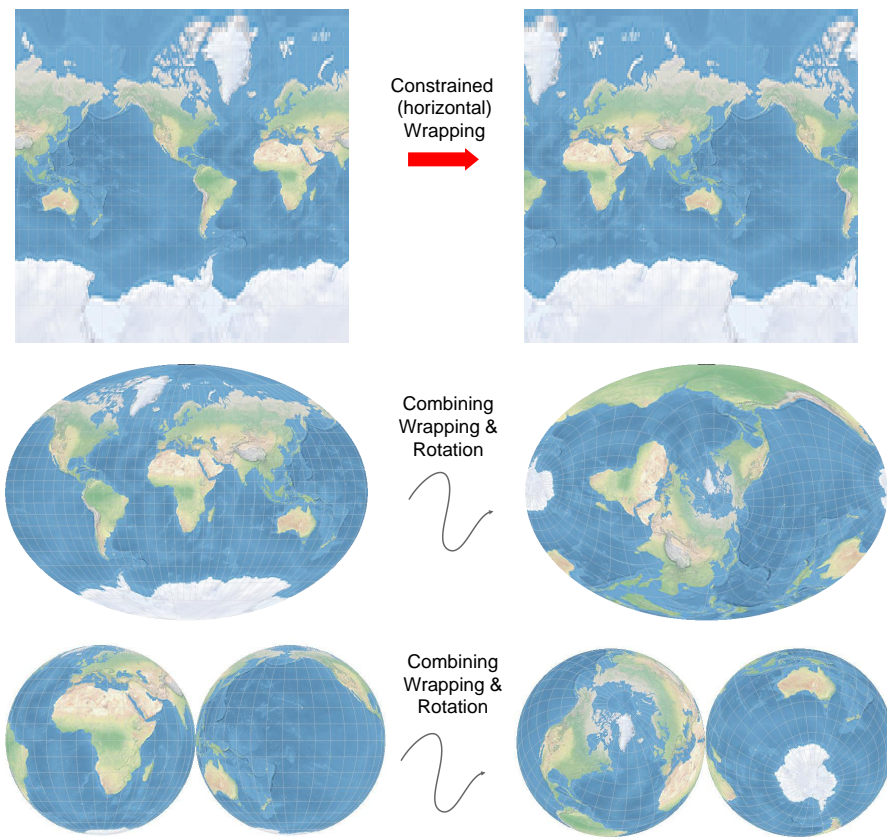
Fig. 3.2.: Examples of design space dimensions for sphere topology, including rectangular (top: Mercator), non-rectangular (middle: Fahey) spherical projections, and Orthographic Hemisphere (bottom). A constrained panning can be applied (top) such that the map is only wrappable in horizontal direction. A user can pan horizontally to explore the map. For Orthographic Hemisphere, we adapted D3's Orthographic Hemisphere projection to create two rotatable complementary Orthographic map projections, with one showing the western and the other showing the eastern hemisphere. They are placed close together. A user can drag one of the hemispheres, while the other hemisphere automatically adjusts rotation angles to show the opposite hemisphere. The evaluation of such interaction is presented in chapter 7.
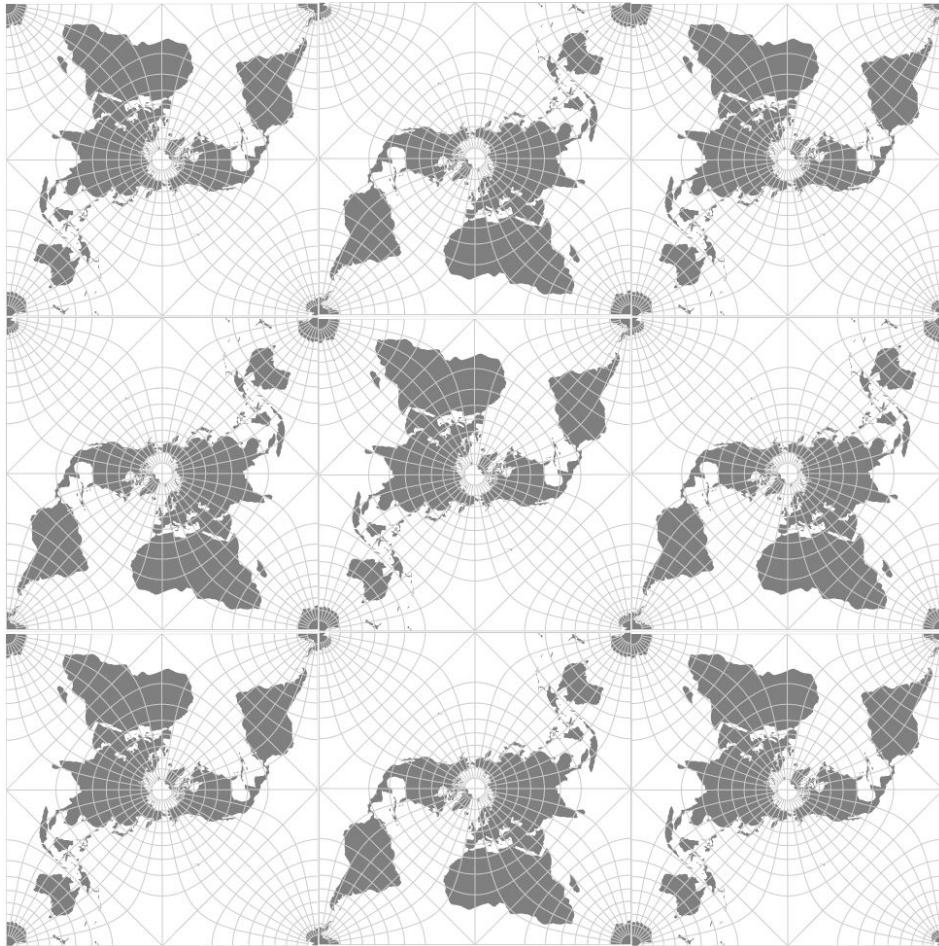
Fig. 3.3.: Tile display in both horizontal and vertical directions. Since the continents are wrapped on the same side of the plane's edge, we develop a simple tool to rotate the tile 180 degrees in (counting from the upper-left to lower-right) tile 1, 3, 5, 7, 9. The repeated tiles show continuity of data points at the boundaries which preserves the data point movement wrapped across the boundaries.

In a non-rectangular projection, the boundary could be curved (e.g., Equal Earth), oval (e.g., Hammer), and the projections could be interrupted (e.g., Orthographic Hemispheric), as discussed previously in section 2.2. There exist a range of different mappings between points across the boundary. Such non-rectangular visualisations still provide for wrapping. For example, in Orthographic Hemisphere projections, there is wrapping between inner sides of two circles in addition to the outer sides of the circles (Figure 3.2-bottom).

Constrained wrapping In common rectangular map projections, e.g., Mercator or Hobo-Dyer projections, the poles remain at the north and south, so the wrapped panning is usually constrained to the horizontal axis (Figure 3.2 - top row). In other words, the map can be panned horizontally with points on the right and left being close. This results in a cylindrical wrapping which is only pannable in the horizontal direction.

Combining wrapping with rotation Many spherical projections can be moved in different directions. For example, in Fahey or Orthographic Hemisphere projections, wrapping can be combined with rotating the projection. The rotation allows the user to reorient and re-centre the projected visualisation with simple mouse or touch drags while wrapping gives the "cut" of the projection, as seen in Fahey projection (Figure 3.2-middle). In the case of Orthographic Hemisphere projection, when one hemisphere is dragged, the Orthographic projection of the other hemisphere automatically adjusts three-axis rotation angles such that it shows the correct opposite hemisphere (Figure 3.2-bottom).

Tile-display If such a spherical projection were repeated to create a seamless tile display, it would show a simple repetition of the projection. In some projections, methods to reduce the discontinuities across the plane's edges exist, such as the 3 by 3 tile display of Pierce's Quincuncial projection [Pei79] (Figure 3.3). In common rectangular map projections, e.g., Mercator, Miller, Equirectangular, horizontal tile display preserves the view of the discontinuities across the left-and-right boundaries (Figure 3.4).

In chapter 7, we explore the usability and effectiveness of a range of spherical projections described above on geospatial data analysis tasks such as area, distance, and direction estimation tasks. We compare different map projection techniques with and without interactive spherical wrapping and rotation as described in our design space.
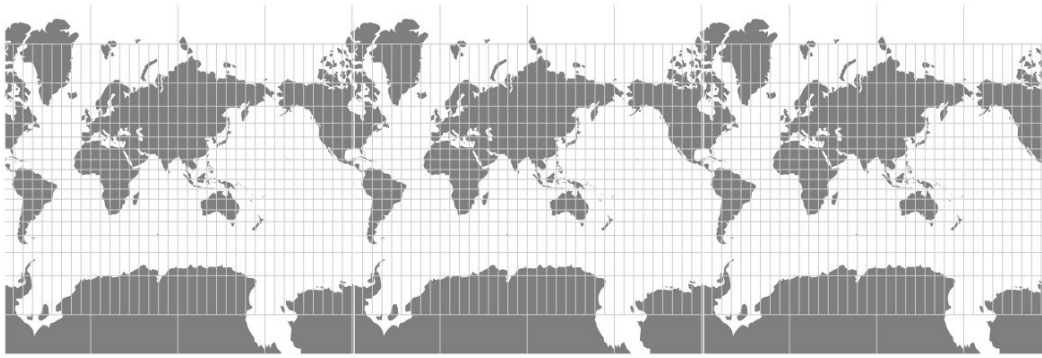
Fig. 3.4.: "Wrapping" dimension in horizontally repeated tiles. The repeated tiles show continuity of data points at the boundaries which preserves the data point movement wrapped across the boundaries.

3.2.1 Non-geographical examples

Beyond geographic visualisations, spherical geometries have been used to create embeddings for node-link graph visualisations. This includes: generalisation of force-directed network layout methods over non-Euclidean surfaces (e.g., sphere) by Kobourov et al. [KL08]; laying out social networks on the surface of a sphere to provide better use of available space, using multidimensional scaling by Martins et al. [Mar+12] and Perry et al. [Per+20]; 2D projections of spherical embeddings of networks without more centred or privileged nodes, compared to traditional flat node-link layout by Rodighiero [Rod20]; and force-directed implementation of 2D projected spherical network layouts by Manning [Man17].

Despite the work described above, spherical projections of networks are less common than their 3D representation in immersive environments, as described in chapter 2. When presenting node-link layouts in a 2D rectangular projection, e.g., Miller or Mercator, using our adapted spherical network layout algorithm (detailed in chapter 8), links between nodes wrap across the boundary with data points between the left and right meet up, while the links at the top or bottom remain at top or bottom being horizontally mirrored. Those links appear greatly distorted (Figure 3.5-top). In non-rectangular projections (e.g., Mollweide Hemisphere, introduced in chapter 2), the links wrap around the circumference between the inner (and outer) side of the circles.

Compared to geographic data, a network has no inherent spatial structure. Therefore, wrapped panning for a network has no inherent constraint. The wrapping and rotation interaction could be applied freely in both horizontal and vertical directions. This allows the viewer to move any region of interest to the centre, as seen in Figure 3.5.
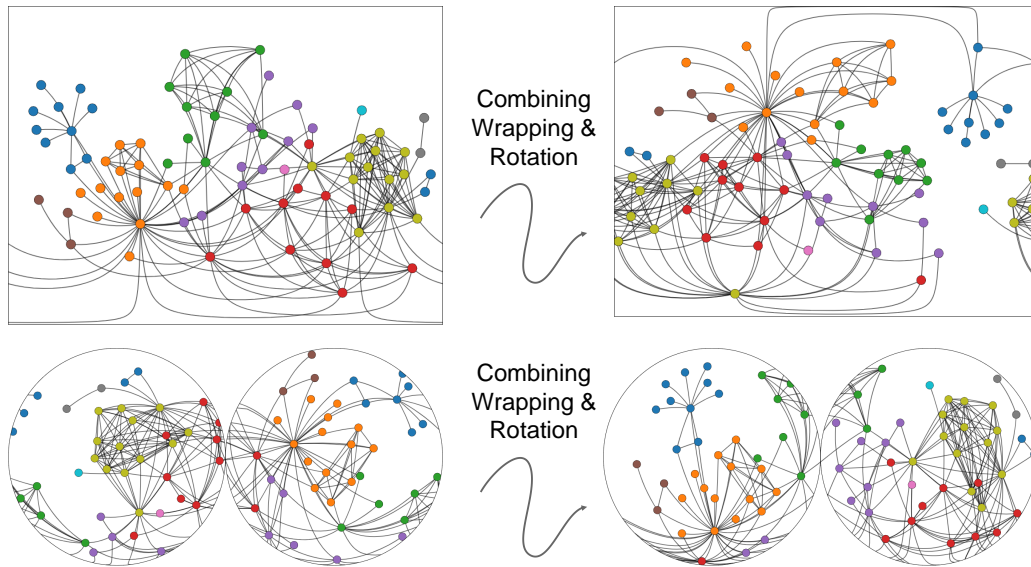
Fig. 3.5.: A rectangular (top: Miller) and non-rectangular (bottom: Mollweide Hemisphere) spherical projection for a network layout of co-appearances of characters (nodes) in *Les Misérables* by Victor Hugo [Knu93], where edges indicate that two characters coexist in one scene and colours indicate group affiliation of the characters (nodes) in the novel. The network is laid out using our layout algorithm described in chapter 8.

To show the link continuity wrapped across the boundaries of a 2D sphere layout, a network can be represented with tile-display. For example, tile repetition of Quincuncial projection of a network shows link connectivity wrapped across the horizontal and vertical boundaries Figure 3.8, while in Equirectangular (introduced in chapter 2) projected network layout it shows the link continuity wrapped horizontally (Figure 3.8).

Similar to node-link embeddings, spheres have been used to embed multidimensional scaling (MDS) plots for better capturing the data's underlying topology on the surface of a sphere such as international trade data between cities by Leeuw and Mair [DM09], and by Lu et al. [LCY19]. Spheres have also been used to embed self-organising maps (SOMs) with the advantage that there is no privileged centre (i.e. as shown in Figure 3.7-left, the regions at the centre have more neighbourhood relationships and thus are updated more often) than those at the centre by Wu et al. [WT06a], shown on both 3D representations (Figure 3.7-middle), and projected onto a 2D-plane (Figure 3.7-right).

Figure 3.7 shows a 3D globe representation of a SOM and its 2D pseudo-cylindrical projection showing that clusters between the left-and-right boundary wrap around [WT06a]. Beyond node-link diagrams and SOMs, other information visualisations can poten-
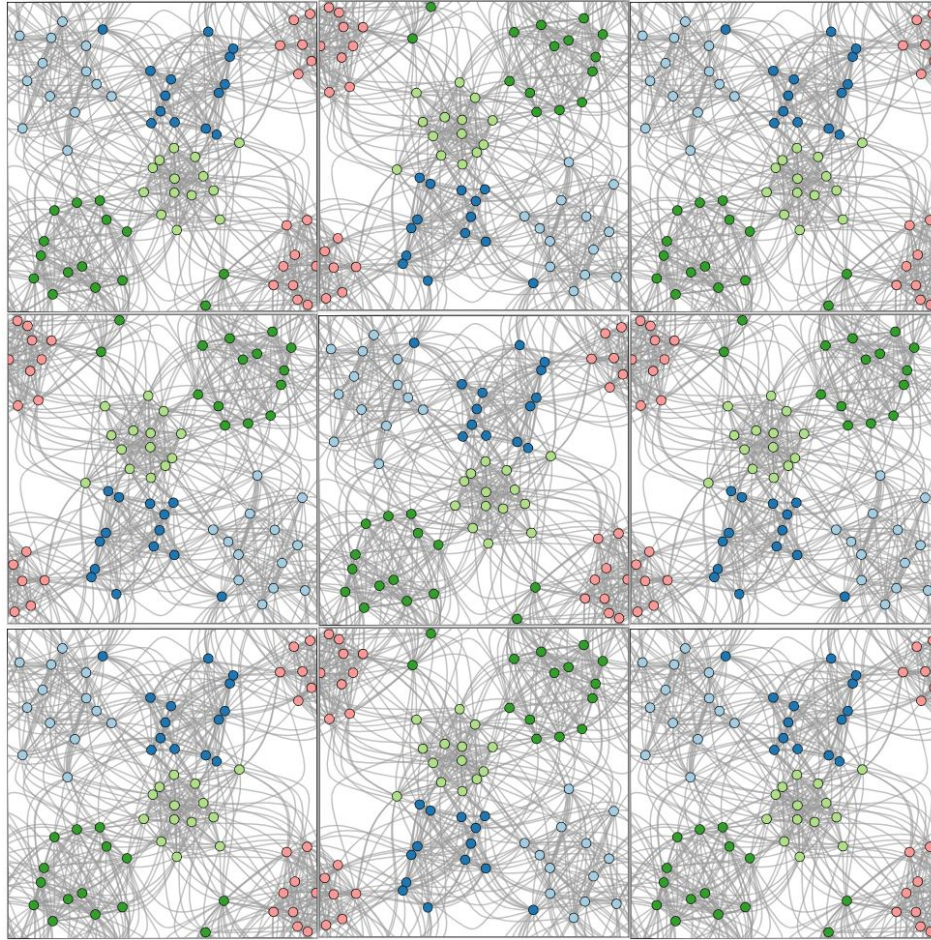
Fig. 3.6.: Pierceś Quincuncial projection of "Wrapping" dimension in both spatial directions: horizontally and vertically repeated tiles. The repeated tiles show continuity of data points at the boundaries which preserves the data point movement wrapped across the boundaries.
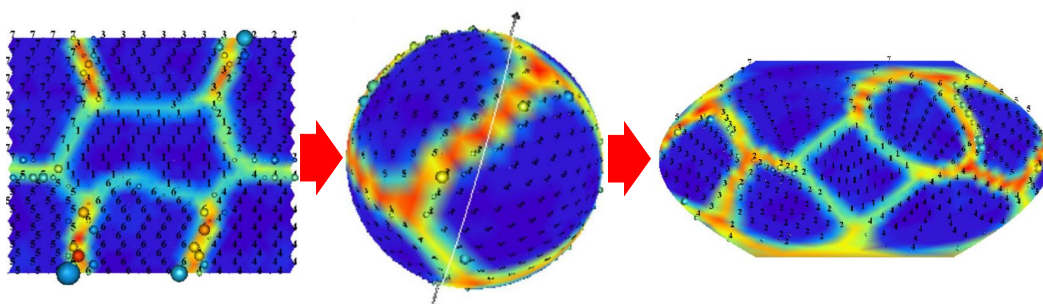
Fig. 3.7.: Traditional arrangement of SOMs on a 2D plane (left) are known to have a privileged centre (or "border effect"). This can be alleviated by rendering on a continuous surface – sphere (middle). The spherical SOM is then projected onto a 2D plane (right) where regions between the left-and-right boundaries meet up. The images are adapted from and credited to Wu et al. [WT06a].
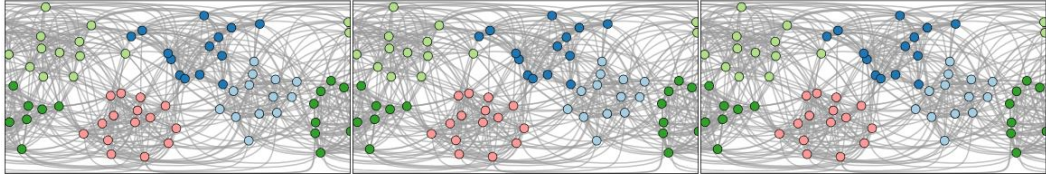
Fig. 3.8.: Tile repetition of an Equirectangular projected spherical network layout.
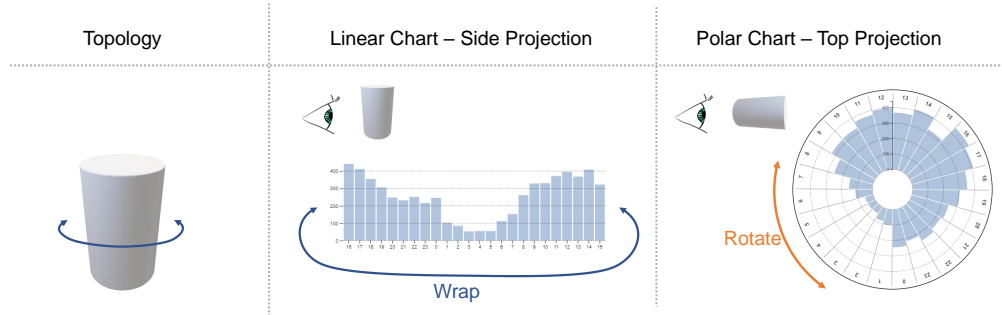
Fig. 3.9.: Rotate or Wrap: average traffic accidents per hour on Thursdays in Manhattan in 2016 (Troshenkov 2021), represented in a 2D projected cylinder with wrap (linear bar chart) and rotate (polar chart) interactions.

tially be projected onto a sphere such that there is no arbitrary edge to the display or privileged centre: multidimensional scaling by Papazoglou et al. [PM17] and Lu et al. [LCY19] and their extension time curves [Bac+15b]. However, we did not find any implementations or visualisations and their implementation might be non-trivial and thus to be studied in future.

In chapter 7 and chapter 8, we investigate the effectiveness of interactive spherical wrapping on maps and networks.

3.3 Cylindrical Wrapping

In this section, we motivate our design space of cylindrical wrapping by considering data types which are intrinsically cyclical (e.g., cyclical time series) which have no beginning or end but are often represented in standard linear bar charts or polar charts.

3.3.1 Motivation: wrap and rotate panning interaction

As described in chapter 1, the arrangement of cyclical data, such as time series in traditional linear bar charts, ignores the cyclic nature of the underlying data and thus may hinder identification of trends or value comparison of bars split across the left-and-right boundary (cut). Existing polar representation provides a common solution for showing time in a continuous manner without introducing the cut, but at the cost of reduced efficiency of data analysis performance.

In order to overcome respective limitations of linear charts and polar charts when representing cyclical time series, we propose bar charts with novel interactive wrapping and polar charts with interactive rotation.

An **interactive bar chart** is a linear bar chart (Figure 3.9-centre) where a user can drag horizontally using mouse or touch interaction to pan the visualisation, such that the chart wraps around from the left to the right and vice versa. This wrap interaction is intended to avoid the issue of bars at the extreme left of the chart being difficult to compare against bars at the extreme right, and making it difficult, for example, to detect trends that continue across this arbitrary cut. The user can simply bring such bars back together by re-centring.

An **interactive polar chart** is a polar bar chart like a static polar (Figure 3.9-right) while a user can drag on any part of the polar bar chart to rotate the chart. The chart spins around the centre in a clockwise or counterclockwise fashion as the user pans accordingly. This rotation interaction was motivated from the arrangement of bars in a polar chart that seemed easier to compare when they could be centred around the vertical centre line. When a pair of bars is centred in this way, the two bars have the same vertical baseline, meaning their heights can be compared directly. When they are off-centre, they are at an awkward angle. Rotation allows the user to centre the two bars they are interested in.

3.3.2 Cylindrical projections

The interaction techniques described in subsection 3.3.1 can be thought of as stemming from the connectivity in cyclical data being mapped to a 3D topology, as per Figure 3.9. For the data with a single cyclical domain, the underlying topology we consider for both linear and polar charts is a cylinder. Both the wrap and rotate interactions can be thought of as a rotation of the 3D cylindrical surface before projecting back to a 2D visualisation along one of two possible view angles of the cylinder, as follows.

Side projection: wrap interaction for bar charts A cyclical bar chart can be wrapped around the cylinder such that the full time period covers the circumference of the cylinder, and the range of the bars representing values will extend to the height of the cylinder. We can rotate the cylinder to choose the time interval that is centred in the field of view. To provide a 2D view of the whole data domain, we can "slice" the cylinder on the side farthest from the viewer and flatten the resulting sheet, as per Fig. 3.9-centre.

Top projection: rotate interaction for polar charts For a polar chart, imagine shrinking the base of the cylinder such that it forms a cone, with the bars extending from the narrow base, outward to the wider top. The polar chart is then simply a view of the cone with the point facing the viewer. Rotation of the cone now changes which bar is centred at the top.

3.3.3 Cylindrical wrapping

We now describe cylindrical wrapping that links to our design dimensions. In **Cylinders**, there is no movement from points from the bottom to the top, or vice versa, while points on the right and left of a visualisation plane wrap (Fig 3.1-centre). Cylinder wrapping can be applied in any visualisations that can benefit from panning continuously along one spatial dimension: vertically or horizontally.

Constrained wrapping For side projected cylinder (as described in subsection 3.1.1), wrapping direction is constrained naturally to only one dimension, but the orientation may be horizontal or vertical. Most interactive Mercator projections, such as Google's 2D map, are in fact cylindrical wrappings as panning is allowed only horizontally (infinitely) (Figure 3.2-top).

Rotational panning For a top-projected cylinder, a user rotates the resulting circular visualisation, so the panning is rotational.

Tile-display Similar to sphere topology, a visualisation can be shown with repeated tiles in horizontal way (Figure 3.4 and Figure 3.8).

In the following, we give further examples. We show how cylindrical topology leads to wrappable or rotatable visualisations and for which kinds of data wrappable visualisations make most sense.
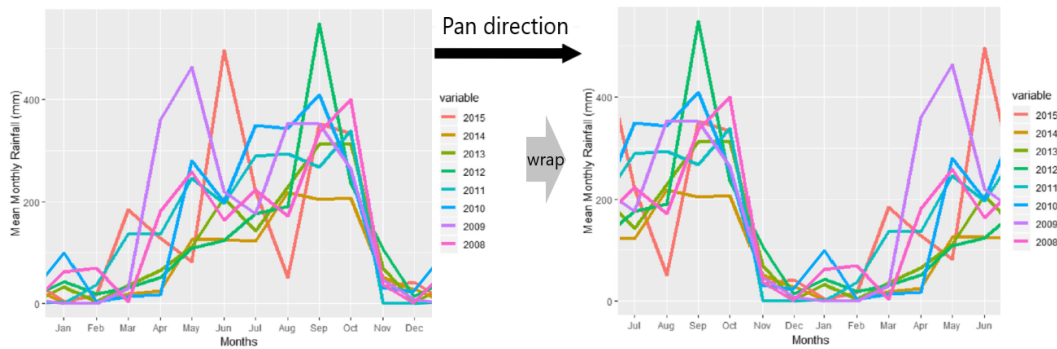
Fig. 3.10.: Cylindrical interaction for standard visualisations of Bangkok's average monthly rainfall from 2008 to 2015: a time series wrapping horizontally and allowing to perceive patterns within individual time periods (e.g,. year.) Image is adapted from Jain et al. citejain2019prediction and credited to Jain et al.

3.3.4 Cyclical time series

We have seen examples of cyclical time series on bar charts and polar charts in chapter 1. Such cyclical data with one periodic dimension (e.g., 24 hours over a day, or 12 months over a year) is suitable to be shown on a cylinder. Their interactive wrapping (for bar charts) or rotation (for polar charts) can be mapped to a cylinder with side or top projection.

Figure 1.2 shows hourly traffic accidents. Panning horizontally allows focusing on specific sections on the cycle, such as temporally (e.g., `Jan-Dec`, (left), or based on min and max values in the data (right). In Figure 1.2-upper left, having 0am on the left and 11pm on the right is completely arbitrary. If we want to look at the midnight (11pm, 0am and 1am) together, we can pan them so the wrap is elsewhere in the day (Figure 1.2-upper right).

Apart from bar charts, cylindrical wrapping can be applied to other types of visuals in a straightforward manner. Figure 3.10 shows a set of average yearly rainfall data shown as lines plotted across 12 months.

3.3.5 Parallel coordinate plots

Parallel coordinates plots are a common type of visualisation used to represent multidimensional data (i.e., mapping n-dimensional relations into 2D patterns), where the dimensions have a sensible cyclical ordering, can be mapped to the cylinder and be either side-projected (Figure 3.11a), or top-projected in a radial form (Figure 3.11b).

(a) PCP side-projection   (b) PCP top-projection
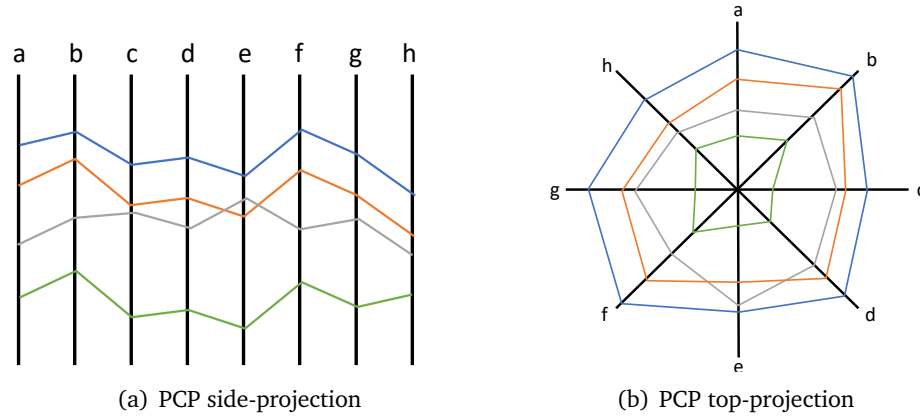
Fig. 3.11.: Cylindrical interaction for standard parallel coordinates plot (PCP) with cyclical dimensions in (a) side-projection; (b) the PCP in top-projection radial view.

3.3.6 Metro maps

The concept of interactive wrapping can equally be applied to other types of data. For example, Figure 3.12a shows a transportation map (two intersecting lines from the Singapore Metro) with a cycle. The transport map above shows a city loop interconnecting the red and green lines. Such a map often shows a cycle interconnecting individual lines. We show that by redrawing the map on a cylinder such that the loop is routed around the circumference, and then by a side-projecting, we can unwrap the cycle (Figure 3.12b). The projected view can then be endlessly panned left or right (Figure 3.12c). If we want to look at the remaining stations of the red line before a transit, we can pan them so the red line starts from the left and wraps around. Furthermore, such a representation could be used on (e.g.) a circle line train to continuously show the stations ahead in the order they will be visited. Such narrower maps can be shown more space-efficiently on, e.g., narrow static or digital displays above doors inside trains.

3.3.7 Cyclical Sankey diagram

In a similar (cyclical network) vein, Figure 3.13 shows a Sankey diagram depicting thermodynamic analysis of water injection in a micro gas turbine[Car+17]. Sankey diagrams are a type of node-link diagram with directed links (arrows) used to depict flows between entities in which the width of the arrows is proportional to the flow rate. Originally developed by Matthew Henry Phineas Riall Sankey to demonstrate flow of energy in steam engines in 1898, Sankey diagrams have become popular in

(a) Two connected lines from the Singapore metro map, causing a cycle.
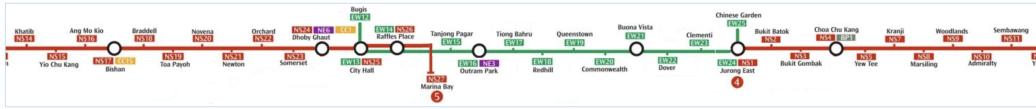
(b) The map from above, wrapped around a cylinder topology and side-projected.

(c) Result of panning the map above to the left.

Fig. 3.12.: Example of cylindrical wrapping: transportation map; image adapted from Singapore metro maps [Sin21]
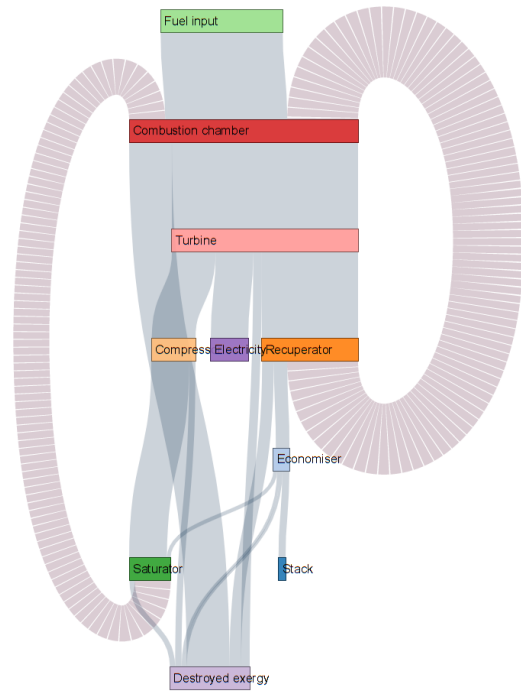
(a) Cyclical links are circular.
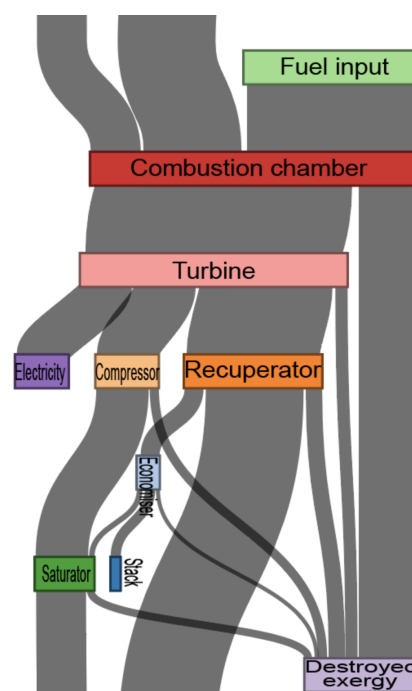
(b) Cylindrical projection to allow continuous vertical panning.

Fig. 3.13.: Examples of cylindrical wrapping: a Sankey diagram that includes a cycle (left) can wrap vertically to highlight the continuous nature of flows (right).
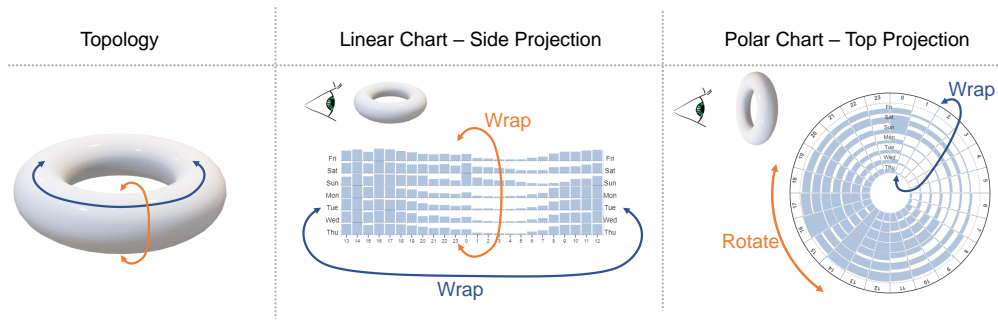
Fig. 3.14.: Rotate or Wrap: average traffic accidents per hour across the week in Manhattan in 2016; dataset taken from Troshenkov [Tro22].

information visualisation for showing movements of data elements between different groupings. For example, Google Analytics uses them to show click-through behaviour of users of web pages. However, in such abstract information visualisation they are rarely depicted with cycles, possibly because they start to look messy, as in Figure 3.13a[3]. In Figure 3.13b, we redraw the same Sankey diagram on a cylinder topology, and project from the side to afford vertical panning.

3.4 Toroidal Wrapping

In this section, we extend the concept of one-dimensional wrap and rotate to consider 2-dimensions for visualisations that can be wrapped both horizontally and vertically. We first describe toroidal interaction techniques. Next, we propose novel uses of toroidal wrapping for cyclical time-series data that have multiple levels of periodicity respectively (e.g. weekly as well as daily, as shown in Figure 3.14). Subsequently, we demonstrate examples of toroidal wrapping for networks represented by node-link diagrams or adjacency matrices [GFC05] and high-dimensional data visualised by self-organising maps [Ito00; Ult03] or multidimensional scaling (as defined in chapter 2).

3.4.1 Toroidal interaction techniques: combining wrapping and rotation

Side projection: 2D wrapped bar chart arrays The concept of a visualisation that can wrap in two-dimensions is best understood, initially, by extending our earlier charts. In Figure 3.14-centre, we further break down the hourly traffic accident data by day of the week. Each bar still represents one hour, but there are now seven small multiple visualisations, one for each day. Like the 24-hour day, the seven-day week is also cyclical. It's conventional to display the week from Sunday to Saturday, with the weekdays centred and consecutive. But what happens if an analyst particularly wants to compare Saturday activity against Sunday? Here is where we can introduce a second (vertical) dimension of interactive wrapping, such that vertical mouse or touch drags cause the day order to wrap, while still allowing horizontal drags to wrap the hours. Topologically, this can be seen as a side view of the torus, where now the hour dimension spans the long circumference of the torus, while the days progress around the circumference of a torus segment (as per Figure 3.14-centre).

Top projection: polar chart arrays that wrap and rotate In Figure 3.14-right, we introduce a new type of visual inspired by a top-down view of the torus. The result is a polar visualisation where, as before, the disc can be rotated through a mouse or touch drag tangential to the disc to change the hour that is centred at the top. In addition, a mouse drag outward from, or towards, the centre of the disc changes the day order. A top projection of a torus results in a polar visualisation, similar to top-down views on cylinder topologies. The difference is that panning can now happen in two ways: using rotation along one dimension and using wrapping to move elements from the *inside* of the polar visualisation to its outside (Figure 3.17). This kind of panning can also overcome the common problem of polar visualisations that visual information at its centre is rendered smaller than on its outer boundary.

3.4.2 Toroidal wrapping

We now describe toroidal wrapping and link visualisations to our design dimensions. In **Toruses**, a 2D visualisation can be panned (wrapped) both horizontally and vertically (Figure 3.1). In user interface design, the earliest example of such torus wrapping may be the classic Asteroids game, described in subsection 2.1.5. Torus topologies have also been explored for mouse pointers on computer screens by Huot

---

[3] generated with a fork of the d3-Sankey software: `http://bl.ocks.org/soxofaan/bb6f91d57dc4b6afe91d`
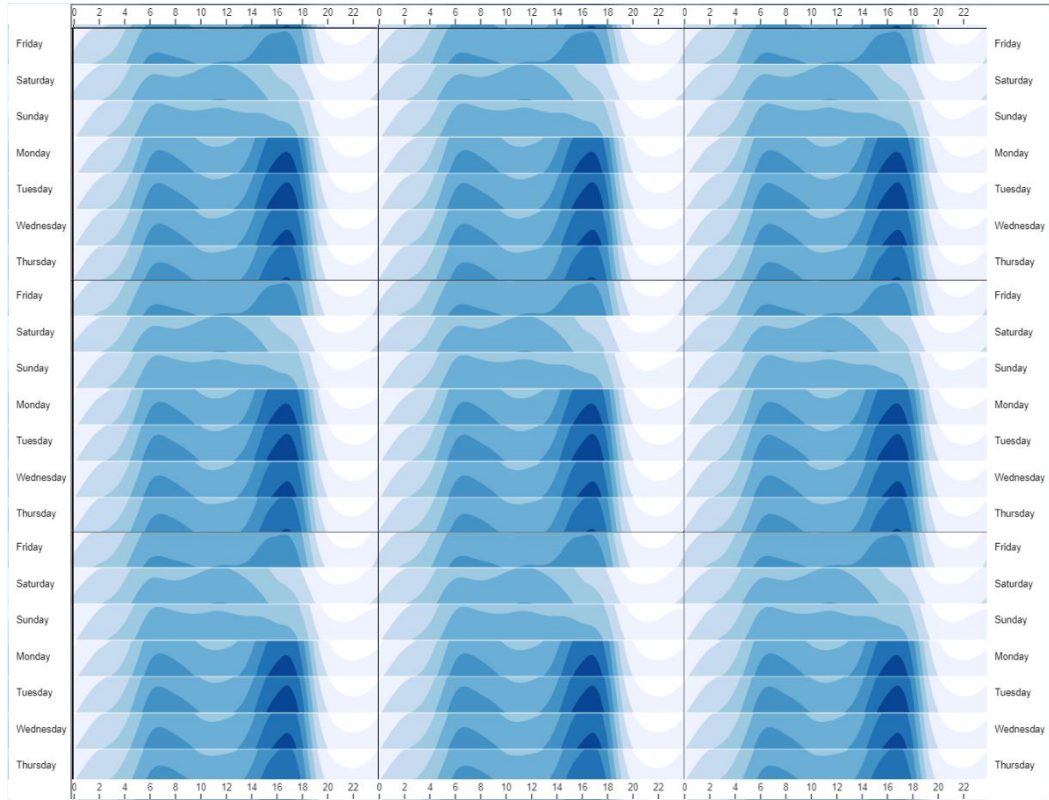
Fig. 3.15.: Example of cyclical time series of New South Wales traffic accidents (dataset taken from Roads and Maritime Services [RS22]) represented on a 2D projected torus topology with 3×3 tile display. Two periodic dimensions such as 24 hours of a day across 7 days a week repeat across the top, bottom, left, and right edges of the original chart (shown in the central tile) without repetition. It allows a viewer to observe continuous ranges of both dimensions across the chart's boundaries with any break in the temporal dimensions.
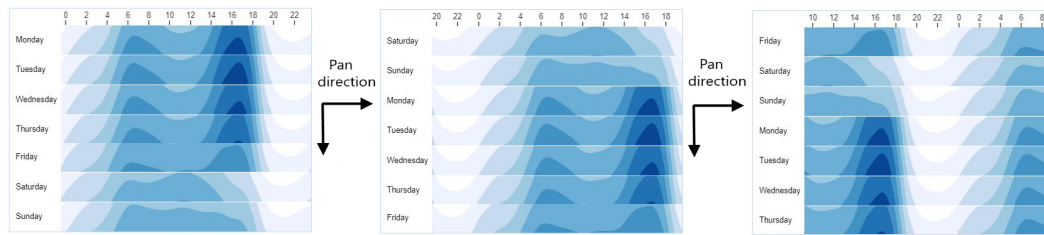
Fig. 3.16.: Example for torus horizontal and vertical torus wrapping, individually.

et al. [HCD11]. However, in Information Visualisation torus topologies have not been systematically explored. Some algorithmic work has been applied to create self-organising maps (SOMs) [Ito00; Ult03] on a torus, but visualisation has always happens in a static plane. Our simple web tool (described in section 3.5) allows us to create torus topologies for any static visualisation, given that axis labels and visualisation "content" are uploaded separately.

Constrained wrapping While panning direction is constrained naturally in cylinder topologies to only one dimension (e.g. horizontally or vertically), on toroidal topologies, users can freely interact with two-dimensional panning, in both horizontal and vertical for a side projection. Such free panning allows for rapid navigation, however, in many visualisations, spatial dimensions have meaning and constraining interactions to one dimension at a time facilitates exploration. In addition, for some toroidal projections of certain data types, it makes sense to further constrain the pannability. For example, a symmetric matrix could be panned both horizontally and vertically, but it usually makes sense to keep the matrix diagonal centred.

Wrapping combining rotation For toroidal topologies, users can freely interact with rotational and radial panning for a top projection.

Tile-display The (static) tile replication for a torus layout provides a continuous view of graphical elements that are wrapped across the boundaries, as seen in Figure 3.15. This allows a viewer to see the continuity of cyclical time series with two periodic dimensions.

In the following, we give further examples. We show how toroidal topology leads to interactive wrappable visualisations, static tiled display, and for which kinds of data wrappable visualisations make most sense.
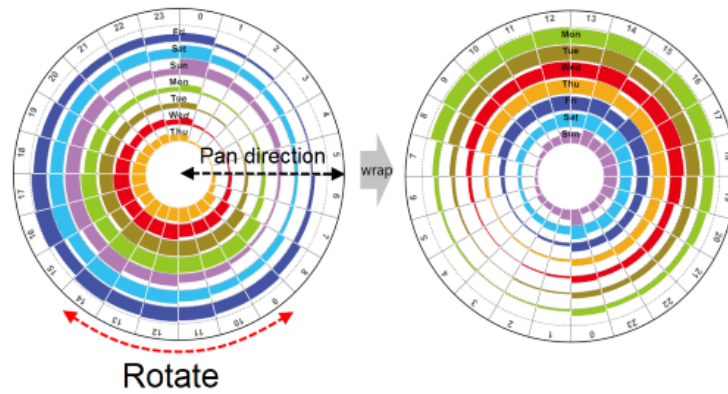
3.4.3 Cyclical time series

The first example that shows temporal dimensions have meaning and that constraining interactions to one dimension at a time facilitates exploration is cyclical time series. For cyclical data with two periodic dimensions, the charts can be designed to be arranged in a 2D bar chart arrays described in subsection 3.4.1, which can be interactively wrapped around horizontally and vertically in a side-projected torus, or arranged in a 2D polar chart arrays that combine rotating in one dimension while wrapping in another, in a top-projected torus, as described in subsection 3.4.1.

Figure 3.16 shows an horizon graph [HKA09] for aggregated time series data over days (vertically) and hours (horizontally). When represented on a side-projected torus, panning horizontally will cycle through the hours of the day (0-24h) and panning vertically will cycle through the days of the week (Mon-Sun). Panning in this example is restricted to one of these dimensions at a time to allow for exploring values across either days or hours without accidentally changing the context of the other dimension. For example, an analyst might be interested in exploring the exact time of peaks across days, and therefore requires the hour-dimension (horizontal) to remain fixed, while panning vertically through the days. The pannable version of this visualisation is created from our web tool. Detailed configuration is discussed in section 3.5.

Take another example. Figure 3.17 shows hourly traffic flow over seven days a week in New York City [Tro22], represented using our 2D polar chart arrays, introduced in subsection 3.4.1. Such a dataset has been evaluated by visualisation literature on a static 2D plane by Waldner et al. [Wal+19]. In a typical arrangement the hours are arranged analogous to a natural clock. However, to compare values between
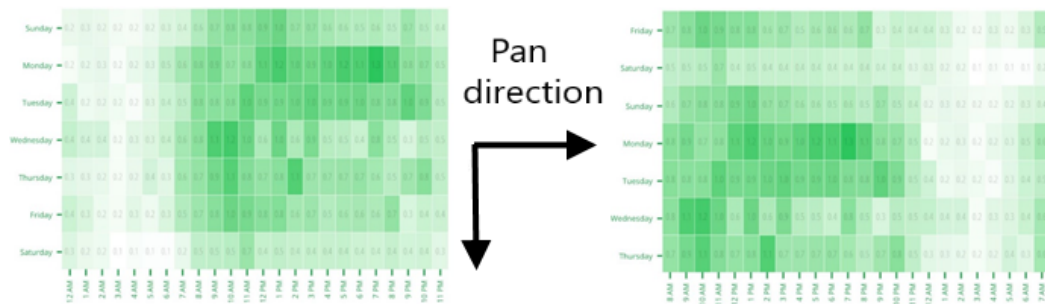
Fig. 3.18.: Average Reddit submissions by hour (horizontal) and day-of-week (vertical) represented on a heatmap can be laid out on a 2D side-projected torus. Images are adapted from average Reddit submissions `https://www.reddit.com/r/dataisbeautiful/comments/1pe4vm/` `heatmap_of_all_link_submissions_to_reddit_which/`

two bars or a range of bars in a day period (e.g., 7am to 6pm) one may need to turn the head around to mentally centre the bars (readability of polar charts compared to linear charts is discussed in chapter 2 and evaluated in chapter 4). In a top-projected torus, panning interaction is supported in two ways. A viewer can use rotation along the hour dimension to centre on the day period from 7am to 6pm for a more aligned comparison, while panning (wrapping) allows for moving, e.g., Monday from the inside of the polar visualisation to its outside, as seen in Figure 3.17-Right, to provide an ordering starting from the first day of a week beginning, as opposed to a typical arrangement shown in Figure 3.17-left.

3.4.4 Heatmaps

Similar to cyclical time series, heatmaps can also be used to represent cyclical time series with two periodic dimensions. The example in Figure 3.18 shows average Reddit submissions by hour (horizontal) and day-of-week (vertical). Since the data has two periodic dimensions, it is suitable to be laid out on a 2D side-projected torus. We specify images for bottom and left axes and no pan constraint.

3.4.5 Adjacency matrices

An example for a torus topology that uses constrained panning are adjacency matrices for network visualisation (Figure 3.19). Some matrix orderings are in fact *cycles* such as those based on the travelling salesman problem [Beh+16]. These matrix orderings make the matrix a torus since their start and end point have to be taken randomly from the permutation and matrices can be panned along the two
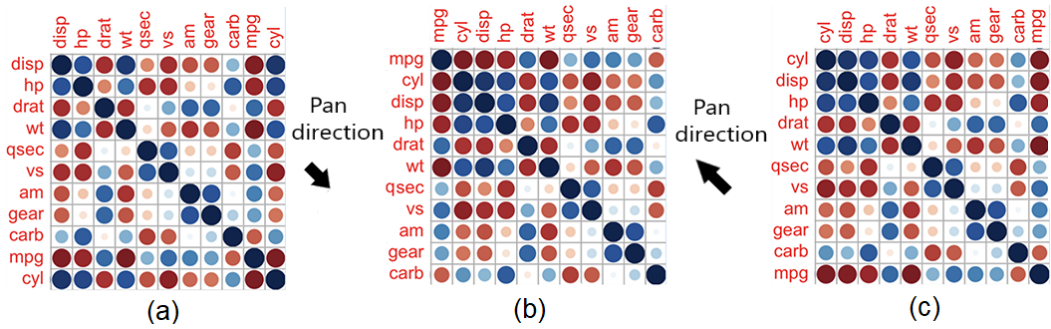
Fig. 3.19.: Effects of diagonal panning on a torus-topology matrix. Using the same row/column ordering: (a) random start and end row/column shows two clusters (blocks along the diagonal), (b) horizontal panning now shows a single cluster (top-left), and (c) panning again highlights strongly correlated `mpg` row/column. Taken from `http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram`

spatial dimensions. Which of the elements becomes the first and last row is usually determined using external heuristics such as the highest-degree node, however, there is no given starting point. Choosing a random starting point from a permutation can lead to arbitrary patterns in a matrix. For example, the matrix in Figure 3.19a seems to show two strong clusters, one on the top-left and one on the bottom-right. The torus panning for matrices therefore can serve two purposes: *(a)* learning and communicating the idea of ordering rows and columns and *(b)* exploring a specific ordering to avoid, e.g., overlooking cells at the margins of the matrix or misinterpreting clusters cut in half by the ordering (Figure 3.19a).

For undirected networks, matrices are symmetric to the diagonal (usually top-left to bottom-right diagonal, if both rows and columns run left-right and top-down respectively). To preserve this important feature of symmetry and to keep viewers' mental map of the matrix preserved, we can constrain the panning so that user can pan only along the diagonal, i.e., the diagonal is fixed while vertical and horizontal panning happen at the *same* time.

Panning the matrix diagonally (Figure 3.19b) reveals that the two clusters visible in Figure 3.19a are in fact one large cluster situated at the top-left of the matrix. Panning further along the diagonal for just *one* single row (Figure 3.19c) highlights the `mpg` row/column by emphasising its strong connections (correlations) with any of the other nodes in the network. The same "pan-configuration" also highlights the red cells in the `mpg`-row/column, which depict negative correlations. In contrast, the configuration in Figure 3.19b visually emphasises the *strength* of the correlation, rather than their *type*. In summary, torus wrapping can help exploration and
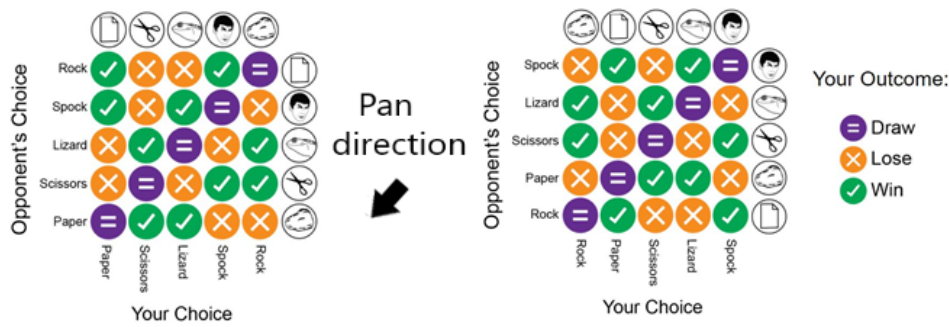
Fig. 3.20.: A directed graph with the special self edges (the draws) mapped to the anti-diagonal. An "antidiagonal" pan constraint keeps those draws where we can easily track them. Adapted from A Rock, Paper, Scissors, Lizard, Spock Chart In R. Images Wikimedia: Creative Commons `https://upload.wikimedia.org/wikipedia/commons/8/89/Rock_paper_scissors_lizard_spock.svg`
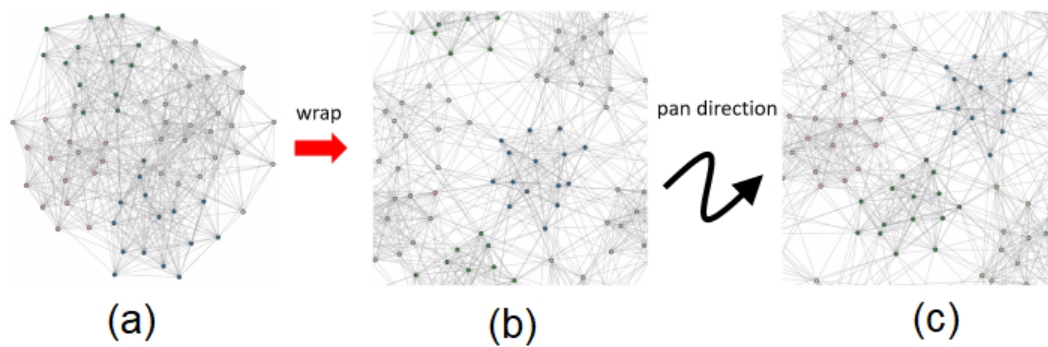
Fig. 3.21.: Examples of torus panning on a network using the random partition network data set [For10] with already known clustering information: (a) traditional (unwrapped) network on a 2D-plane; (b, c) two configurations wrapped network laid-out on a 2D torus, using the algorithm presented in chapter 6.

scrutinising data as well as finding the views most appropriate for a given task or message.

A matrix can also be panned along the anti-diagonal direction. Figure 3.20 shows a directed graph with the special self edges (the draws) mapped to the anti-diagonal. An "antidiagonal" pan constraint keeps those draws where we can easily track them.

3.4.6 Node-link diagrams

A node-link diagram of a network can be embedded onto a torus and side-projected onto a 2D plane, while wrapping specific links around the borders of the display. An illustrative example is shown in Figure 2.4, discussed in subsection 2.1.5. While
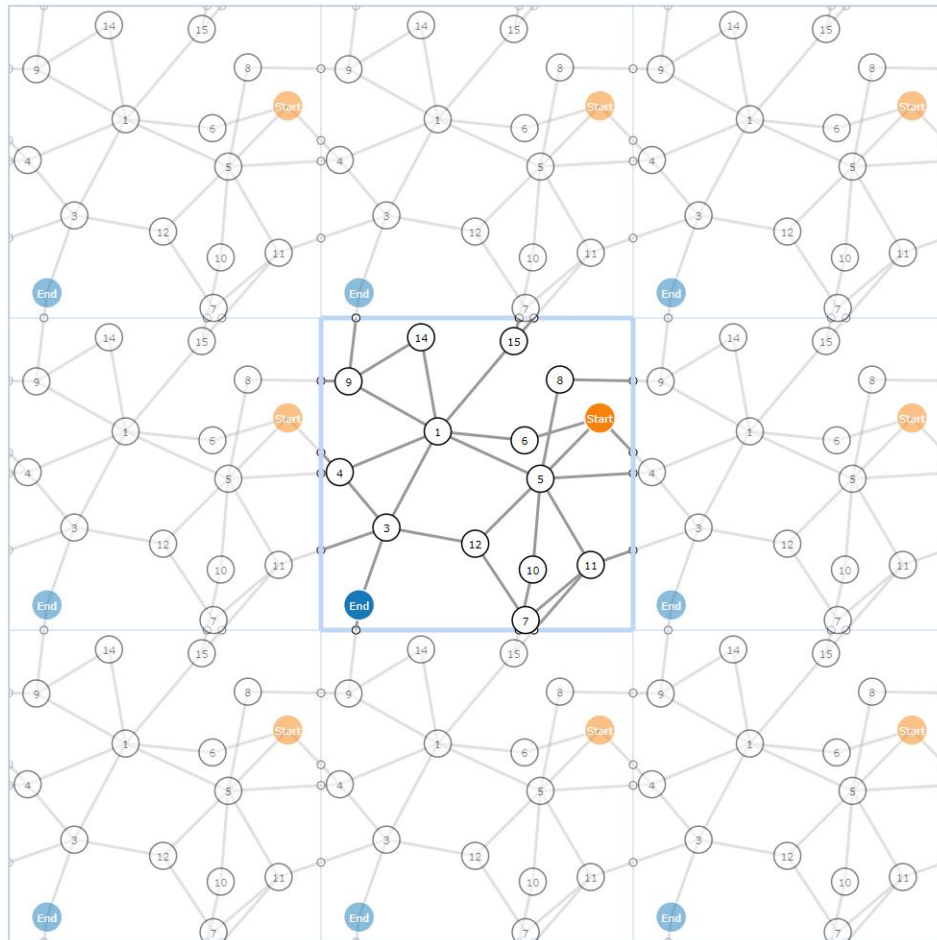
Fig. 3.22.: Example of tile repetition in a torus embedding of a network, laid out using our layout algorithms described in chapter 6.

wrapping some 2D visualisations, such as charts, heatmaps, matrices, or horizon graphs is straightforward, wrapping a network layout on a torus needs to optimise their node positions in a torus topology to better show the patterns or relationships within the data.

Figure 3.21 shows a larger and denser network example with community structures (defined in section 2.1) laid out in node-link diagrams. In (a) it shows that a traditional non-wrapped node-link representation of the network appears more tangled with many links crossing the diagram, making it difficult to inspect the data's underlying structure (investigated in chapter 6. In (b) the same network is laid out in a 2D side-projected torus using our torus wrapping algorithm detailed in chapter 6. It shows that torus layouts make clusters appear more distinct. However, some clusters may be split across the edges of the diagram. In (c) it shows such torus-based network layout is freely pannable in both horizontal and vertical directions without any panning constraint. When a cluster is panned off on the left side of the diagram, it reappears on the horizontally right position, and vice versa. When a node or link is panned off on the top boundary of the diagram, it reappears on the vertically bottom position, and vice versa. This allows a viewer to interactively wrap the network to bring any region of interest to the centre (e.g., to see full clusters).

A torus-based network can also be illustrated with a tile display, with replicated network layouts shown on the edges of the central highlighted diagram to show the link continuity wrapped across the boundaries. For example, Figure 3.22 shows tile repetition of a network which shows link connectivity wrapped across the left-and-right and top-and-bottom sides of the original layout.

In chapter 5 and chapter 6, we explore layout algorithms that can produce the drawings of above examples, and investigate whether such tiled display (chapter 5) and the aforementioned interactive wrapping (chapter 5 and chapter 6) improve the usability of torus-based network layouts.

3.4.7 Self-organising maps

An illustrative example of self-organising maps is shown in Figure 3.23 where tiled repetition is shown that preserves the connectivity of the graphical elements split across the left-and-right, or top-and-bottom boundaries by Ultsch [Ult03]. SOM is also used for preserving the topological structure of multidimensional data in lower dimensions by Kohonen [Koh82].
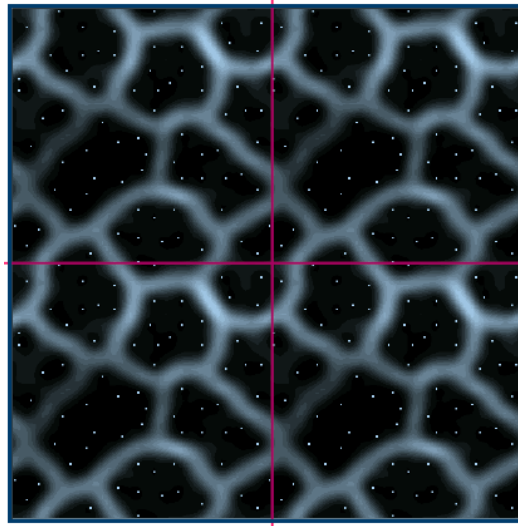
3.4.8 Multidimensional scaling

A MDS plot can be laid-out on a torus topology. This example (Figure 3.24) shows a MDS plot based on the hepta data set. The 3-dimension attributes can be converted into a 2D similarity matrix between all pairs of nodes. The diagrams in the left column show an unwrapped 2D MDS plot of 50-70 points, 7 classifications using a standard force-directed layout method. The diagrams in the right column show the same plot laid-out on a torus, using the same algorithm we developed in chapter 6. Figure 3.24 shows a multidimensional scaling (MDS) calculated on a torus topology. Figure 3.24(b) and Figure 3.24(c) show our torus layouts in two different panning configurations (free vertical and horizontal panning, no panning constraints).

3.5 Web Tools for Creating Wrappable Visualisations

To create the wrappable visualisations described and figured in previous sections, we implemented a small web tool[4] that can present arbitrary 2D visualisations with one- or two-dimensional wrapped panning and allows for the selection of panning constraints in accord with the topology (cylinder or torus). The tool, called `WrappingChart` is a small JavaScript library which allows web developers to easily create interactive wrap-around visualisations like those depicted in this chapter from their own static images of charts or other visuals. The user can prepare separate

---

[4]`https://github.com/Kun-Ting/WrappingChart`
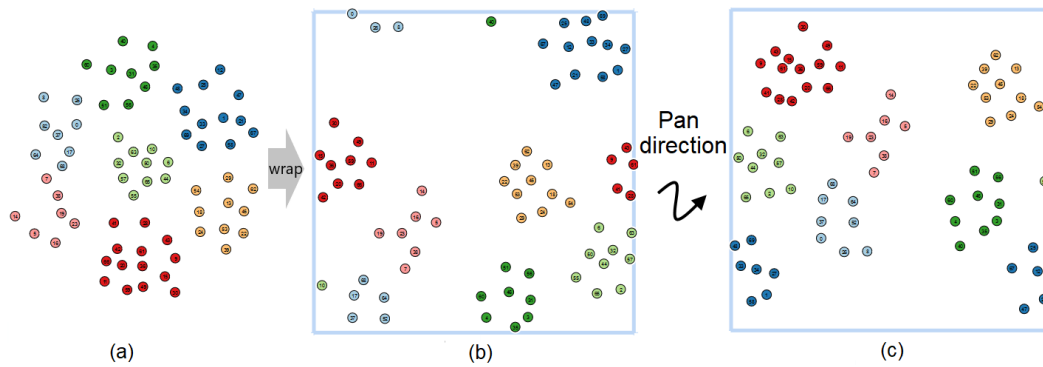
Fig. 3.24.: Examples of torus panning on an MDS using the Hepta data set [TU20] with already known clustering information: (a) traditional (unwrapped) MDS plot on a 2D-plane; (b, c) two configurations wrapped MDS plot laid-out on a 2D torus, using the algorithm presented in chapter 6. Datasets was adapted from Benchmark data set from Fundamental Cluster Problem Suite (FCPS).

```
<script>
    wrapChart("simpleHorizontalWrap2",
        "images/traffic_accidents_2017_chart.png",
        null,
        "images/traffic_accidents_2017_horizontal_axis.png",
        "images/traffic_accidents_2017_vertical_axis.png",
        null,
        "horizontal")
</script>
```

Fig. 3.25.: Interactive tool library for one-dimension (cylindrical) wrapping or two-dimensional toroidal) wrapping of visualisations

images for the visualisation and optional axes-labels that remain fixed at the sides while the visualisation is being panned. Example is shown Figure 3.25.

The API and parameters are described below. The user can upload separate images for the visualisation and optional axes-labels that remain fixed at the sides while the visualisation is being panned. The tool was developed using Typescript.

- HTML element that hosts the interactive visualisation (`targetElementSelector`)

- chart body (`bodyImageURL`)

- image URL of x-axis (e.g., tick marks, bars, labels) at the top boundary (`xAxisTopImageURL`)

- image URL of x-axis at the bottom boundary (`xAxisBottomImageURL`)

- image URL of y-axis at the left boundary (`yAxisLeftImageURL`)

- image URL of y-axis at the right boundary (`yAxisRightImageURL`)

Fig. 3.26.: Cylindrical wrapping of Hobo-Dyer projection

Fig. 3.27.: Interactive cylindrical wrapping configuration of cyclical time series data using our tool

- constrained panning: `panConstraint` (possible value: horizontal, vertical, diagonal, antidiagonal)

Geographic visualisation For example, in Mercator projection of the world map (Figure 3.2-top), to have poles remain at the north and south and to keep adjacencies of continents on the earth's surface correct, only horizontal panning makes sense. We call our *wrapChart* library (with a "horizontal" pan constraint to limit the panning (Figure 3.26).

Hourly traffic flow over 24-hour period

To create wrappable visualisations as shown in Figure 3.16, we can upload the images of the bar charts and x-axis, y-axis images of hourly traffic flows over months or years. We call wrapChart with a "horizontal" pan constraint to limit the panning (Figure 3.27).

Our small library supports interactive panning of data visualisations. While we develop separate tools for creating tile-display and interactive rotation for our cylindrical and toroidal design space, in future, we consider integrating them as options: "rotate", "wrapandrotate", "sphericalrotate" ("sphericalrotate" needs to specify the projection type as wrapping differs accordingly). We will also add a

new boolean variable to turn on/off tile-display. The tile repetition is automatically computed based on Constraint.

For cylindrical topology, we also implement a similarly novel 'rotate' gesture for polar charts, which allows the user to bring any bar to the top or centre bars for more aligned comparison.

3.6 Conclusion

In this chapter, we present a design space exploration for a novel class of interactive wrapped visualisations. The outcomes of this exploration characterise data types whose visualisations can potentially benefit from being wrapped around 2D surfaces of 3D shapes for projection back to a 2D interactive wrap-around display. They may benefit from being understood as 'cylindrical', 'toroidal', or 'spherical', addressing **RG1** (subsection 1.2.1). This design space exploration leads us to develop novel interaction techniques for not only geographic data, but also cyclical time series, multidimensional and network structured data. The design space also contributes to a number of caveats and design guidelines for future applications.

We showed that the design space can be a powerful tool to: *(a)* understand relations between visualisations (e.g., linear and polar bar charts), *(b)* apply interactive wrapping to a range of visualisations, and *(c)* to create new visualisations by mapping existing visualisations onto sphere, cylinder and torus visualisations.

We presented a simple interactive web tool that is capable of wrapping arbitrary data visualisations around 2D projected cylindrical and toroidal topologies with support for constrained panning, addressing **RG2** (subsection 1.2.2). The JavaScript source code for our library API is available to web developers to easily create their own wrap-around data visualisations.

In the following chapters, we present design, development and evaluation of cyclical, toroidal, spherical wrapping for cyclical time series, networks, geographic maps, showing the benefits of interactive wrapped visualisations compared with standard unwrapped representations.

Cylindrical Wrapping of Cyclical Data

4

> 99 *The interactive charts were significantly easier than the standard charts. Being able to move the chart and line up the areas instead of having them go off screen was a great advantage. By far, the hardest was the standard circle chart. I found myself having to really turn my neck to get a good look at some of them.*
>
> — **Anonymised user study participant, P2**
> Trend identification tasks

Inspired by the observations in our design space (chapter 3), this chapter demonstrates the perceptual benefits of interactive (pannable) wrapped visualisation over standard visualisations. We start with cylindrical topology that is suitable for data that is cyclic in one temporal or spatial dimension. Examples include average rainfall over 12 months, hourly traffic accidents rate, or average wind strength over different compass directions.

This chapter investigates the trade-off between the continuity provided by polar visualisations versus the ease of height comparison on linear charts, and—most noteworthy—the use of novel (cylindrical) interactions for polar and linear chart representations of cyclical data to work around their respective limitations.

Specifically, we address the following open questions (linked to RG3 in subsection 1.2.3):

- **[RQ3.1]**: Does adding interactive wrapping to linear bar charts improve their effectiveness?
- **[RQ3.2]**: How effective is interactive wrapping for linear bar charts compared with static polar bar charts?
- **[RQ3.3]**: Does adding interactive rotation to polar bar charts improve their effectiveness?

To investigate these questions, we report on a controlled user study with 72 participants comparing four different visualisations of cyclic data: (a) linear bar chart (without interactive wrapping); (b) linear bar chart with interactive wrapping; (c) polar bar chart (without interactive rotation); and (d) polar bar chart with interactive rotation.

4.1 Empirical Evidence

Several studies have investigated error performance with bar charts, finding that people are more accurate at comparing *adjacent* bars than bars that are further apart by Cleveland et al. [CM84], Heer et al. [HB10], and the reproduced version of Cleveland et al.'s in four experiments by Talbot et al. [TSA14]. Other studies of comparison of cyclical data, in particular, temporal data (hourly, weekly, or monthly) with static linear or polar bar charts have been discussed in section 2.3.

To cope with the complexity of temporal data, interaction has been investigated. For example, pan and zoom along a linear timeline by Schwab et al. [Sch+19], mouse hover and showing hints on a linear chart or circular chart by Adnan et al. [AJB16], and a study of interaction and single-scale selection techniques on a linear timeline by Schwab et al. [SPB21] have been presented. However, we are not aware of wrapped panning (as defined in Sec. 7.1), being evaluated with cyclical time-dependent data.

In our work, we investigate interactive panning as one possibility to wrap a linear visualisation. This notion is different from other notions of wrapping in visualisations, such as *"DuBois Wrapping"* by Karduni et al. [Kar+20], which wraps oversized bars in bar charts into zig-zag lines.

In summary, while visualisation tasks described in this chapter have been previously tested in literature (described in section 2.3), such as trend detection [AJB16; ZT99], pairwise group comparison [AJB16], and pairwise single value comparison [Wal+19], no study has particularly focused on bars far apart but which could be brought closer through wrapping. Our contribution to this corpus of knowledge is the particular study of interaction for wrapping in linear bar charts and rotation in radial charts. This task suggests looking at intervals across both ends of a linear bar chart as well as bars that are 'far apart' in both chart types that could be brought 'closer together' in interactive linear bar charts.

Fig. 4.1.: Bar chart and polar chart representation of hourly traffic flow over a 24-hour period in NYC in 2016 [Tro22]. To inspect whether the average hourly traffic accidents continue to increase or decrease from midnight till morning (upper-left), a user may need to mentally wrap the bars from 11pm at the right end across the page to 1am at the left end. Polar chart (lower-left) arranges the sequence of bars continuously, but certain sequences of bars (12pm to 5pm) are not aligned at the centre and thus a user may need to spin their head to make comparisons. Interactive wrapped and rotatable visualisations proposed in this thesis (upper-right, lower-right) allow a user to centre any group of bars while the visual cut is elsewhere in other hours (upper-right) and to have the same vertical baseline (lower-right) for comparing heights of bars.

4.2 User Study 1: Wrapped Time Series Readability

This section reports on a controlled user study with 72 participants investigating the performance of linear and interactive wrapped charts to understand time series data. The study aims to answer our questions Q1 to Q3 stated earlier. Apart from the interactive wrapping, our study differs from past work by focusing on tasks that require considering multiple bars: pair comparisons and aggregation across intervals (trend identification and average value estimation), as we expect these to be the tasks most affected by wrapping in bar charts and rotational centring in polar charts.

The full study material, illustrative examples of interactive wrapping or rotation are available in the Open Science Foundation: `https://osf.io/r8cw4/`.

4.2.1 Techniques

In our study, we compare four techniques for the visualisation of cyclical data: *static bar*, *interactive bar*, *static polar* and *interactive polar*.

- STATICBAR represents a traditional static bar chart, as in Figure 3.9a.
- STATICPOLAR represents a traditional static polar bar chart, arranging bars in a radial fashion, as in Figure 3.9b.
- INTERACTIVEBAR is a linear bar chart (Figure 3.9a) with interactive cylindrical wrapping, using mouse or touch drags, as described in subsection 3.3.1.
- INTERACTIVEPOLAR is a polar bar chart like STATICPOLAR (Figure 3.9b) with interactive cylindrical rotation, using mouse or touch drags, as described in subsection 3.3.1.

4.2.2 Hypotheses

We group our hypotheses by our research questions RQ4.1-RQ4.3. Hypotheses have been pre-registered with the Open Science Foundation: `https://osf.io/9k5bm`.

[RQ3.1] Does adding interactive wrapping to linear bar charts improve their effectiveness?

- H3.1-error: INTERACTIVEBAR has less ERROR than STATICBAR. This is observed by the existing study results that people perform more accurately when comparing adjacent bars than bars that are far apart by Talbot et al. [TSA14]. With our interactive technique, it allows them to bring bars that are far apart closer.
- H3.1-time: STATICBAR has less TIME than INTERACTIVEBAR as the latter requires panning.
- H3.2-error: INTERACTIVEBAR has less ERROR than INTERACTIVEPOLAR.
- H3.2-time: INTERACTIVEBAR has less TIME than INTERACTIVEPOLAR.

[RQ3.2] How effective is interactive wrapping for linear bar charts compared with static polar bar charts?

- H3.3-error: INTERACTIVEBAR has less ERROR than STATICPOLAR.
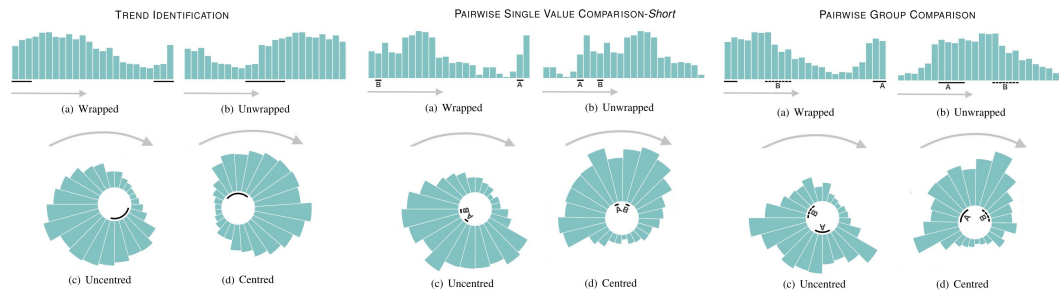
Fig. 4.2.: Examples of data sets and visual stimuli for all three tasks used in the study: TREND IDENTIFICATION(left), PAIRWISE GROUP COMPARISON(centre) and PAIRWISE SINGLE VALUE COMPARISON(right). Sub figures (a)-(d) show the individual conditions for wrapped and unwrapped as well as centred and uncentred across both visualisation techniques.

[RQ3.3] Does adding interactive rotation to polar bar charts improve their effectiveness?

- H3.4-error: INTERACTIVEPOLAR has less ERROR than STATICPOLAR since participants are able to rotate the chart to best solve the task.
- H3.4-time: STATICPOLAR has less TIME than INTERACTIVEPOLAR since the latter potentially involves panning.

Our last hypothesis is about user preference:

- P4.1: user prefers INTERACTIVEBAR to STATICBAR, STATICPOLAR, or INTERACTIVEPOLAR since INTERACTIVEBAR facilitates comparisons across the cut while allowing to best compare bar heights.

4.2.3 Tasks

We designed three tasks to compare our four techniques. The tasks are motivated by existing time series visualisation research [Wal+19; Bre+18; AJB16; Fuc+13; Cor+12; ACG14], and graphical perception task typology by Brehmer and Munzner [BM13]. However, they did not look at the effect of interactive panning nor did they look at a group of bars that span across the cuts. We focus on the particular effect of analysing data across the cuts in linear static and interactive bar charts. To that end, we created two levels of difficulty for each task: *unwrapped* (for bar charts) or *centred* (for polar charts) and *wrapped* (for bar charts) or *uncentred* (for polar bar charts). In the *wrapped* condition, the user is required to analyse the data across the cuts, compared to the *unwrapped, centred, and uncentred* conditions. Examples for all tasks and conditions are shown in Figure 4.2.

- **TREND IDENTIFICATION:** *"What describes the highlighted sequence of bars best, e.g., continuously increasing/continuously decreasing/neither?"* (Figure 4.2-TREND IDENTIFICATION) The participants were asked to identify the trend of a highlighted set of adjacent bars. The timeout for this task was 5 seconds, inspired by our pilot studies. We recorded participants' responses with multiple-choice questions with 4 options with *continuous decreasing, continuous increasing, neither*, or *unsure*. We created 10 trials for this task per technique, 3 of them were monotonically increasing, 3 were monotonically decreasing intervals, 4 were neither. We fixed the length of all intervals to be 6 bars.

- **PAIRWISE GROUP COMPARISON:** *"Which group of bars, A or B, has the higher total values?"* (Figure 4.2-PAIRWISE GROUP COMPARISON) The timeout for this task was 10 seconds. We gave this task more time, as it required mental aggregation of a group of bar height. We recorded participants' responses with multiple-choice questions with 3 options with *A, B*, and *unsure*. We created 10 trials for this task per technique, balancing the answers of *A* and *B*.

- **PAIRWISE SINGLE VALUE COMPARISON:** *Which bar, A or B, has the higher value?* (Figure 4.2-PAIRWISE SINGLE VALUE COMPARISON) The timeout for this task was 5 seconds. We created 12 trials for this task per technique. We controlled for the distance between targeted bars, i.e., 6 trials per technique for *Short* (2 bars apart) and 6 trials per technique for *Long* (6 bars apart). We balanced the answers of *A* and *B*. We recorded participants' responses with multiple-choice questions with 3 options with *A, B*, and *unsure*.

4.2.4 Visual configuration

The width of the bar chart was fixed to 628 pixels, which was the same as the median grid line of the circumference of a polar chart with 200-pixels radius, as seen in Figure 3.9-middle and Figure 3.9-right. The y-axis scale of the bar chart was the same as the y-axis scale in the polar chart, i.e., 200 pixels. This setting was similar to prior visualisation studies comparing polar and bar charts by Waldner et al. [Wal+19]. For the study, we removed all tick marks, grid lines and labels, such that visual comparison is based purely on bar height. We used a monochrome colour for the bars, as Adnan et al. [AJB16] found colour visual encoding makes the effect of linear or polar chart negligible for time series visualisation. For the polar chart, we used an inner circle with a radius of 50 pixels for showing a curve. This marks the bars in each task, as shown in Figure 4.2. The corresponding visual encoding was an underline for bar charts. To support TREND IDENTIFICATION, each chart was accompanied by a grey arrow indicating the direction of time inside the visualisation.

For the linear bar charts, the arrow was shown from the left to the right on the bottom of the chart. We implemented each technique using D3.js [Bos22].

4.2.5 Data

Our stimuli were generated from New York hourly traffic accident datasets between 2013 and 2016 [Tro22]. We selected Brooklyn, Bronx, Manhattan, Queens, and Staten Island borough data. These data show a non-smooth distribution of the aggregated number of traffic accidents over 24 hours, which we felt would provide a suitable and ecologically-valid data set for our study. The same data has been used in prior time series visualisation studies comparing linear and polar charts by Waldner et al. [Wal+19]. To generate cyclical temporal data for the study tasks, we first aggregated the hourly traffic accidents on a monthly basis for each borough. We then obtained average hourly traffic accident datasets in 24 hours (i.e., 24 data points) for a given borough, month and year to obtain hundreds of candidate sample data sets. We selected from these candidates for our task stimuli, controlling for difficulty based on pilot testing, as follows:

- TREND IDENTIFICATION: Across all samples we restricted the candidates to instances with a monotonically increasing or decreasing set of six adjacent bars, with a minimum height difference of 1-2% (i.e., small but apparent on the displays tested). For neither monotonically increasing nor decreasing instances, there are no other monotonic sequences of more than two bars. Furthermore, we added one quality control (obvious) trial per technique with a much larger minimum height difference of greater than 20%.
- PAIRWISE GROUP COMPARISON: The difference of total values between two groups was 5-15% of bar chart height. For the quality control trial, the difference is set to be greater than or equal to 20% of bar chart height.
- PAIRWISE SINGLE VALUE COMPARISON: The difference of heights between the two bars was 1-1.5% of bar chart height. For the quality control trial, the difference is set to be greater than or equal to 20% of the bar chart height.

4.2.6 Experimental design

Our experimental design follows the practical guideline for experimental human-computer interaction by Purchase [Pur12]. Similar to many existing time series studies that crowdsourced (explained in section 2.4) their studies [Sch+19; Wal+19;

Bre+18], and to evaluate the applicability of our interactive wrapped visualisation based on how a time series visualisation would be accessed on the Internet today, we opt for a crowdsourced study.

We decided on a within-subject design (explained in section 2.4), where each participant performed all of the 4 techniques × 3 tasks. We used 10 repetitions for TREND IDENTIFICATION, PAIRWISE GROUP COMPARISON, and 6 repetitions for PAIRWISE SINGLE VALUE COMPARISON-*Short* and 6 repetitions for PAIRWISE SINGLE VALUE COMPARISON-*Long*. Following existing unsupervised time series studies by Brehmer et al. [Bre+18], we inserted a quality control trial for each technique per task to check participants' attention. We used 4 practice trials for each technique per task. This gives a total of 47 trials × 4 techniques = 188 trials per participant. We used a full-factorial design to counterbalance the learning effect of four techniques (which results in a total of 24 different orderings of techniques). Each recorded trail has a timer associated with the task type, as described in subsection 4.2.3. We expected the study to complete within 40 minutes. The order of *wrapped*, *unwrapped*, *centred, and uncentred* questions for each task was randomised across tasks per technique. Each trial used a randomly selected dataset from 5 borough data that satisfies the constraints as described in subsection 4.2.5. Therefore, none of the same graphics appear twice throughout the study.

We used equal numbers of trials for each of these four groups. We recorded *task-completion time* (TIME), *task-error* (ERROR), and *subjective user preference* (PREF) as dependent variables across all the tasks.

4.2.7 Participants and procedure

We crowdsourced the study via both convenience sampling and the Prolific platform [PS18]. The participants on the Prolific platform have been reported producing data quality comparable to the dominant Amazon Mechanical Turk by Peer et al. [Pee+17]. Many time series visualisation studies were in fact crowdsourced [Sch+19; Wal+19; Bre+18; Di +20; Sat+21; ACG14; Cor+12], with increasing number of visualisation studies deployed over the Prolific platform, e.g., by Satriadi et al. [Sat+21].

We hosted the study on our web server application. We set a pre-screening criterion on performance, i.e., minimum approval rate of 95%, and minimum number of previous submissions of 10. We also limited our study to desktop users with larger screens. We provided a payment of £5 (i.e. a rate of £7.5/h) to Prolific participants.

This is considered good payment according to the Prolific platform. We recorded 72 participants who passed quality control tasks and completed the study. This comprised 3 full counterbalanced blocks of participants. There were 48 participants from the Prolific group and 24 participants from local sampling (recruited via email). Out of all participants, there were 23 identified as females, 47 identified as males and 2 participants preferred not to disclose their gender. The age of participants was between 18 and 50. Each participant went through all of the 4 techniques with the order assigned by the software with the following procedure. First, they completed a tutorial explaining the technique and task. For TREND IDENTIFICATION, participants were instructed to read the trend from a set of adjacent bars highlighted by either an underline (for bar charts) or a curve (for polar charts). Examples of monotonically increasing, monotonically decreasing, and neither were given during the training as well as in practice trials. An example of neither is shown in Figure 4.2-TREND IDENTIFICATION(a-d). During the training, when the participant's answer was incorrect they would be shown the same practice trial again until a correct answer was given.

For INTERACTIVEBAR and INTERACTIVEPOLAR, an animated image demonstrating the wrapping or rotation interaction was shown for the task. Participants were encouraged to try the interaction themselves with the same example as the one in the animation. Following that, participants were required to successfully complete 4 practice trials before proceeding to the recorded trials. For recorded trials, each trial was first loaded on a participant's browser before the software started the timer. Each participant went through the same task order, i.e., TREND IDENTIFICATION→PAIRWISE GROUP COMPARISON→PAIRWISE SINGLE VALUE COMPARISON-*Short*→ PAIRWISE SINGLE VALUE COMPARISON-*Long*.

4.2.8 Results

We report on the results of 72 participants from both the Prolific group and the convenience sampling group. All of these participants passed the attention check trials, completed the training and recorded trials. This resulted in 9,216 trials. We excluded attention check trials in the analysis, as they used a much looser constraint than recorded trials, as described in subsection 4.2.5. Therefore, we have equal numbers of *wrapped*, *unwrapped* trials for bar charts, and equal numbers of *centred* and *uncentred* trials for polar charts for each task in the analysis.

Since the distribution of ERROR and PREF of each technique did *not* follow a normal distribution, to compare error rates between different techniques we used standard
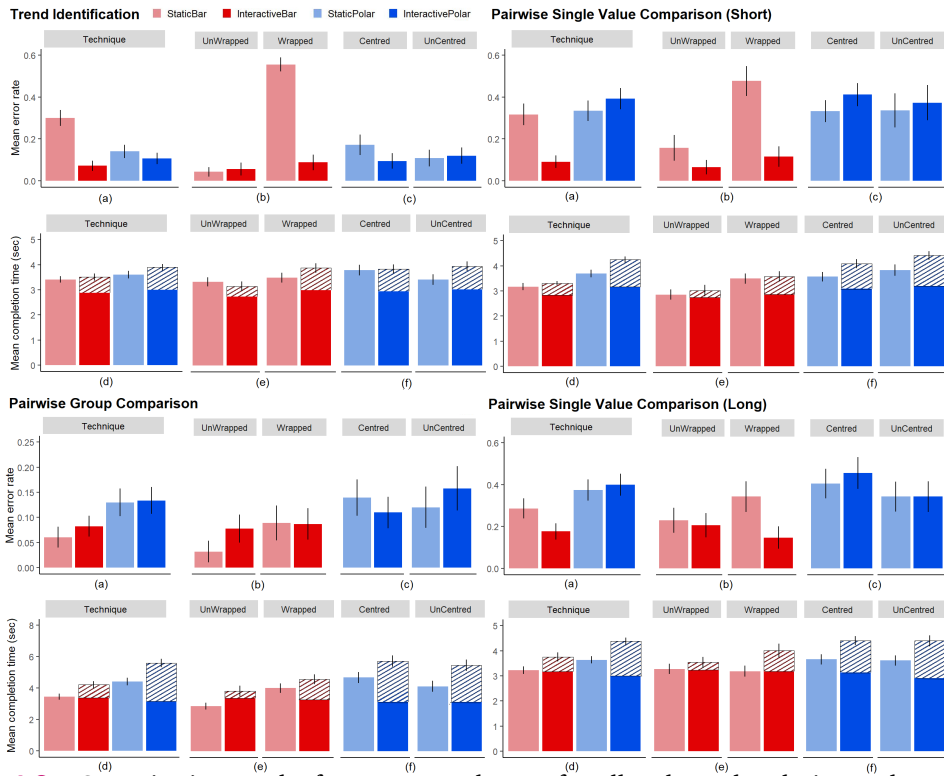
Fig. 4.3.: Quantitative results for ERROR and TIME for all tasks and techniques: bar charts (red) and polar charts (blue). Static techniques are indicated in a lighter colour. Textured parts of some bars indicate the fraction of time used for panning (INTERACTIVEBAR) and rotating (INTERACTIVEPOLAR). The bars show the results of average performance in terms of error rate and task completion time. Error bars indicate 95% confidence interval.
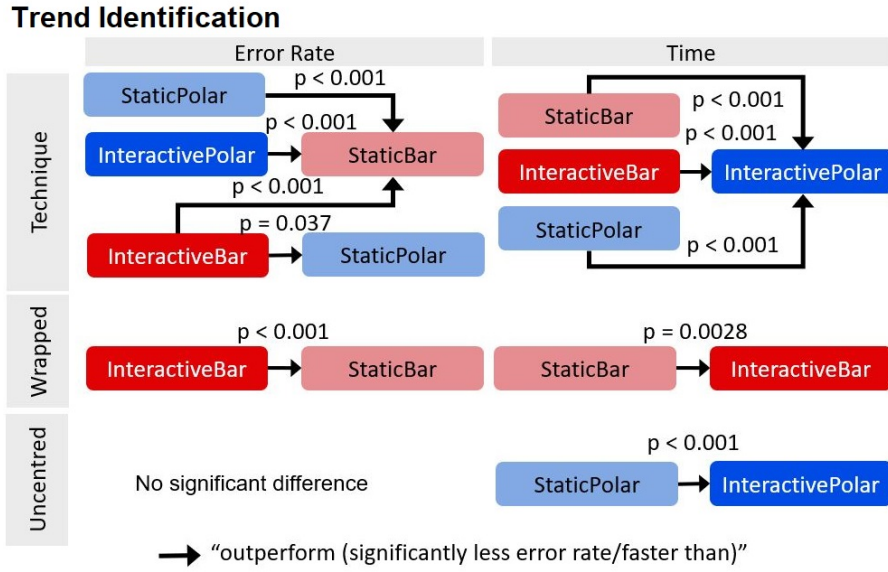
Fig. 4.4.: Statistically significant results of TREND IDENTIFICATION

non-parametric statistics described by Field et al. [FMF12]. To compare multiple (four) conditions, we used Friedman's non-parametric test and Tukey's post hoc pairwise comparison with Bonferroni correction [FMF12] to identify significant differences between STATICBAR, INTERACTIVEBAR, STATICPOLAR, and INTERACTIVE-POLAR. The confidence interval is 95%. Since TIME of each technique was normally distributed, tested with Shapiro-Wilk's normality test [SW65] and visually checked by Q-Q plot, we used standard parametric statistics methods, described by Field et al. [FMF12]. To compare multiple (four) techniques, we used analysis of variance (ANOVA) repeated measures (i.e. hypothesis testing for within-subject analysis) and Tukey's post hoc pairwise comparison with Bonferroni correction [FMF12] to test significance. The results of error bar graphs are shown in Figure 4.3 and Figure 4.8. We report on the most significant findings by tasks as follows.

Trend identification

Friedman's test revealed a statistically significant effect of techniques on ERROR ($X^2(3) = 142.6; p < .001$). Post-hoc tests showed that INTERACTIVEBAR ($.07, SD = .19$), STATICBAR ($.14, SD = .26$) and INTERACTIVEPOLAR ($.1, SD = .22$) resulted in significantly less errors ($p < .001$) than STATICBAR ($.3, SD = .3$), as seen in Figure 4.3-TREND IDENTIFICATION(a) and Figure 4.4-ERROR. We also found that with interactive wrapping, INTERACTIVEBAR resulted in significantly less errors than STATICPOLAR ($p = .037$), as seen in Figure 4.3-TREND IDENTIFICATION(a) and Fig-

Pairwise Single Value Comparison (Short)
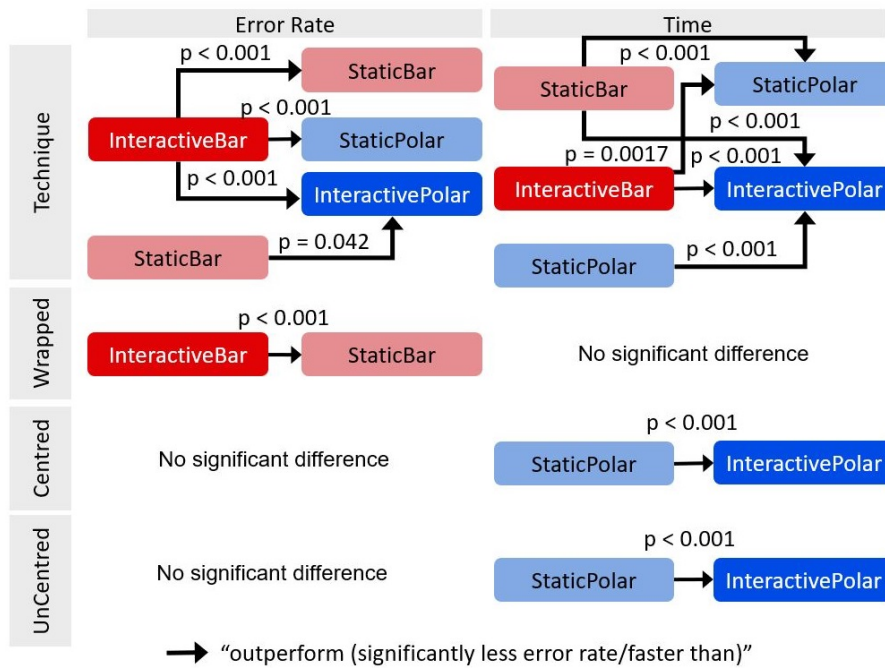
Fig. 4.5.: Statistically significant results of PAIRWISE SINGLE VALUE COMPARISON (Short distance - 2 bars apart)

ure 4.4-ERROR. For wrapped trials, INTERACTIVEBAR (.08, $SD = .21$) significantly outperformed STATICBAR (.56, $SD = .19$) in terms of error ($p < .001$), as seen in Figure 4.3-TREND IDENTIFICATION(b) and Figure 4.4-ERROR. The results of repeated measure ANOVA showed a statistically significant effect of techniques on TIME ($F = 8.54, p < .001$). Post-hoc tests showed that INTERACTIVEBAR (3.95s, $SD = 1.08$) was significantly ($p < .001$) slower than STATICBAR (3.45s, $SD = 1.09$), STATICPOLAR (3.65s, $SD = 1.24$) and INTERACTIVEBAR (3.57s, $SD = 1.17$), as seen in Figure 4.3-TREND IDENTIFICATION(d) and Figure 4.4-TIME. INTERACTIVEBAR (3.93s, $SD = 1.07$) was significantly ($p = .0028$) slower than STATICBAR (3.54s, $SD = 1.13$) for wrapped trials, as seen in Figure 4.3-TREND IDENTIFICATION(e) and Figure 4.4-TIME. For Uncentred trials, INTERACTIVEPOLAR (4s, $SD = 1.11$) was significantly ($p < .001$) slower than STATICPOLAR (3.46s, $SD = 1.22$), as shown in Figure 4.3-TREND IDENTIFICATION(f) and Figure 4.4-TIME.

Pairwise single value comparison-Short

Friedman's test revealed a statistically significant effect of techniques on ERROR ($X^2(3) = 118.88; p < .001$). Post-hoc tests showed that INTERACTIVEBAR (.09, $SD =$

.24) resulted in significantly ($p < .001$) less error than STATICBAR (.32, $SD = .41$), STATICPOLAR (.33, $SD = .39$), and INTERACTIVEPOLAR (.39, $SD = .41$), as seen in Figure 4.3-PAIRWISE SINGLE VALUE COMPARISON-Short(a) and Figure 4.5-ERROR. STATICBAR significantly ($p = .042$) outperformed INTERACTIVEPOLAR in terms of error. For wrapped trials, INTERACTIVEBAR (.11, $SD = .28$) resulted in significantly ($p < .001$) less errors than STATICBAR (.48, $SD = .41$), as shown in Figure 4.3-PAIRWISE SINGLE VALUE COMPARISON-Short(b) and Figure 4.5-ERROR. The results of repeated measure ANOVA showed a statistically significant effect of techniques on time ($F = 38.2, p < .001$). Post-hoc tests showed that INTERACTIVEPOLAR ($4.31s, SD = 1.03$) was significantly slower than STATICBAR ($3.21s, SD = 1.23$), STATICPOLAR ($3.74s, SD = 1.22$) and INTERACTIVEBAR ($3.34s, SD = 1.29$), as shown in Figure 4.3-PAIRWISE SINGLE VALUE COMPARISON-Short(d) and Figure 4.5-TIME. STATICPOLAR was significantly ($p < .001$) slower than STATICBAR and INTERACTIVEBAR. There was no significant difference between INTERACTIVEBAR and STATICBAR. For centred trials, INTERACTIVEPOLAR ($.4.15s, SD = 1.04$) was significantly ($p < .001$) slower than STATICPOLAR ($3.61s, SD = 1.11$). For uncentred trials, INTERACTIVEPOLAR ($4.46s, SD = 1$) was significantly ($p < .001$) slower than STATICPOLAR ($3.87s, SD = 1.31$).

Pairwise single value comparison-Long

The Friedman's test revealed a statistically significant effect of techniques on ERROR ($X^2(3) = 61.3; p < .001$). Post-hoc tests showed that INTERACTIVEBAR (.17, $SD = .32$) resulted in significantly less errors ($p < .001$) than STATICPOLAR (.37, $SD = .41$) and INTERACTIVEPOLAR (.4, $SD = .43$). INTERACTIVEBAR resulted in significantly less errors ($p = .02$) than STATICBAR (.29, $SD = .38$), as seen in Figure 4.3-PAIRWISE SINGLE VALUE COMPARISON-Long(a) and Figure 4.6-ERROR. For wrapped trials, INTERACTIVEBAR (.14, $SD = .3$) significantly outperformed STATICBAR (.34, $SD = .42$) in terms of error ($p = .0015$), as seen in Figure 4.3-PAIRWISE SINGLE VALUE COMPARISON-Long(b) and Figure 4.6-ERROR. The results of repeated measure ANOVA showed a statistically significant effect of techniques on time ($F = 33.03, p < .001$). Post-hoc tests showed that INTERACTIVEPOLAR ($4.45s, SD = 1.15$) was significantly ($p < .001$) slower than STATICBAR ($3.27s, SD = 1.21$), STATICPOLAR ($3.68s, SD = 1.16$) and INTERACTIVEBAR ($3.83s, SD = 1.48$), as seen in Figure 4.3-PAIRWISE SINGLE VALUE COMPARISON-Long(d) and Figure 4.6-TIME. STATICPOLAR was significantly ($p < .001$) slower than STATICBAR, as seen in Figure 4.3-PAIRWISE SINGLE VALUE COMPARISON-Long(d) and Figure 4.6-TIME. INTERACTIVEBAR was significantly ($p < .001$) slower than
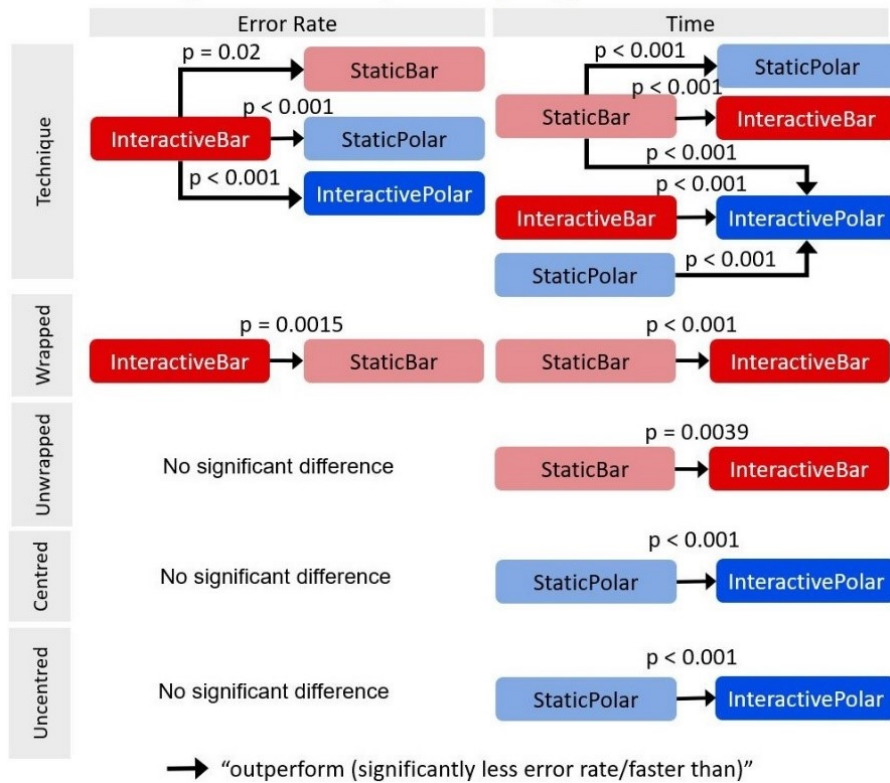
Fig. 4.6.: Statistically significant results of Pairwise Single Value Comparison (Long distance - 6 bars apart)
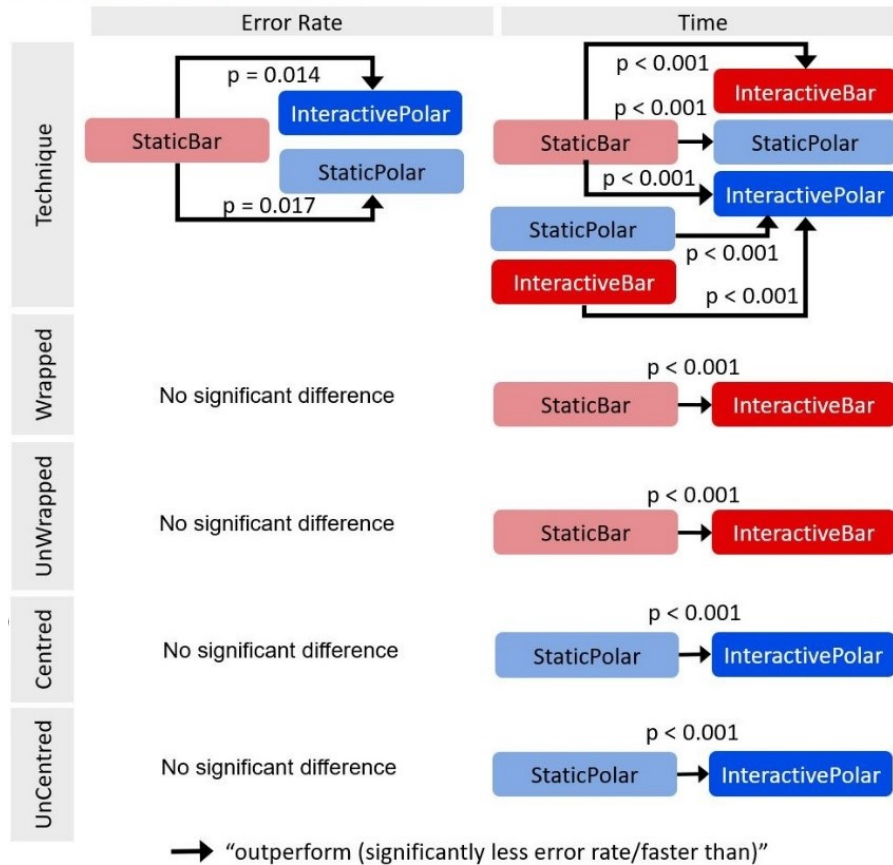
Fig. 4.7.: Statistically significant results of PAIRWISE GROUP COMPARISON

STATICBAR. We also found that INTERACTIVEBAR $(4.08, SD = 1.6)$ was significantly slower than STATICBAR $(3.23, SD = 1.23)$ for wrapped $(p < .001)$ and unwrapped $(p = .0039)$ trials, as seen in Figure 4.3- PAIRWISE SINGLE VALUE COMPARISON-Long(e) and Figure 4.6-TIME. For centred/uncentred trials, INTERACTIVE-POLAR $(4s, SD = 1.11)$ was significantly $(p < .001)$ slower than STATICPOLAR $(3.46s, SD = 1.22)$, as shown in Figure 4.3-PAIRWISE SINGLE VALUE COMPARISON-Long(f) and Figure 4.6-TIME.

Pairwise group comparison

Friedman's test revealed a statistically significant effect of techniques on ERROR $(X^2(3) = 32.9; p < .001)$. Post-hoc tests showed that STATICBAR $(.06, SD = .16)$ resulted in significantly less errors $(p = .017)$ than STATICPOLAR $(.131, SD = .22)$. STATICBAR $(.06, SD = .16)$ resulted in significantly less errors $(p = .014)$ than
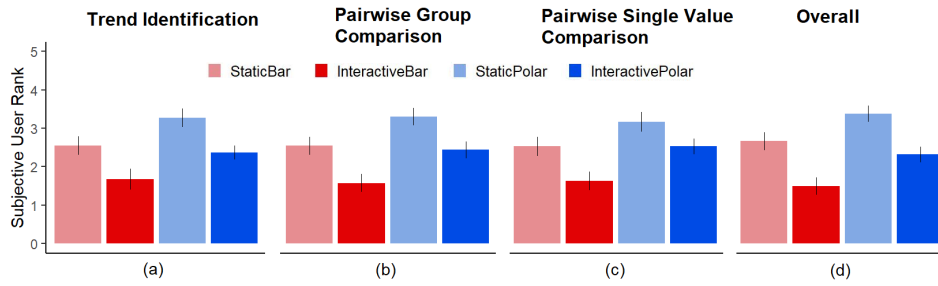
Fig. 4.8.: Subjective User ranking for all tasks and techniques (lower is better).

INTERACTIVEPOLAR $(.135, SD = .22)$, as seen in Figure 4.3-PAIRWISE GROUP COMPARISON(a) and Figure 4.7-ERROR. The reults of repeated measure ANOVA showed a statistically significant effect of techniques on TIME $(F = 33.03, p < .001)$. Post-hoc tests showed that INTERACTIVEPOLAR $(4.45s, SD = 1.15)$ was significantly $(p < .001)$ slower than STATICBAR $(3.27s, SD = 1.21)$, STATICPOLAR $(3.68s, SD = 1.16)$ and INTERACTIVEBAR $(3.83s, SD = 1.48)$, as seen in Figure 4.3-PAIRWISE GROUP COMPARISON(d) and Figure 4.7-TIME. STATICPOLAR was significantly $(p < .001)$ slower than STATICBAR, as seen in Figure 4.3- PAIRWISE GROUP COMPARISON(d) and Figure 4.7-TIME. INTERACTIVEBAR was significantly $(p < .001)$ slower than STATICBAR, as seen in Figure 4.3-PAIRWISE GROUP COMPARISON(d) and Figure 4.7-TIME. For either wrapped or unwrapped trials, INTERACTIVEBAR was significantly $(p < .001)$ slower than STATICBAR, as seen in Figure 4.3- PAIRWISE GROUP COMPARISON(e) and Figure 4.7-TIME.

User preference

For INTERACTIVEBAR: Friedman's test revealed a statistically significant effect of techniques on PREF $(X^2(3) = 57.7; p < .001)$. Post-hoc tests showed that INTERACTIVEBAR was significantly preferred over STATICBAR $(p < .001)$, STATICPOLAR $(p < .001)$ and INTERACTIVEPOLAR $(p = .0054)$. STATICBAR was significantly preferred over STATICPOLAR $(p = .0034)$. INTERACTIVEPOLAR was significantly preferred over STATICPOLAR $(p < .001)$.

For PAIRWISE SINGLE VALUE COMPARISON: Friedman's test revealed a statistically significant effect of techniques on PREF $(X^2(3) = 53.1; p < .001)$. Post-hoc tests showed that INTERACTIVEBAR was significantly preferred over STATICBAR $(p < .001)$, STATICPOLAR $(p < .001)$ and INTERACTIVEPOLAR $(p < .001)$. STATICBAR was significantly preferred over STATICPOLAR $(p = .015)$. INTERACTIVEPOLAR was significantly preferred over STATICPOLAR $(p = .015)$.

For PAIRWISE GROUP COMPARISON: Friedman's test revealed a statistically significant effect of techniques on PREF ($X^2(3) = 66.5; p < .001$). Post-hoc tests showed that IN-TERACTIVEBAR was significantly preferred over STATICBAR ($p < .001$), STATICPOLAR ($p < .001$) and INTERACTIVEPOLAR ($p < .001$). STATICBAR was significantly preferred over STATICPOLAR ($p = .0021$). INTERACTIVEPOLAR was significantly preferred over STATICPOLAR ($p < .001$).

Overall: Friedman's test revealed a statistically significant effect of techniques on PREF ($X^2(3) = 79.7; p < .001$). Post-hoc tests showed that INTERACTIVEBAR was significantly preferred over STATICBAR ($p < .001$), STATICPOLAR ($p < .001$) and IN-TERACTIVEPOLAR ($p < .001$). STATICBAR was significantly preferred over STATICPO-LAR ($p = .005$). INTERACTIVEPOLAR was significantly preferred over STATICPOLAR ($p < .001$).

4.2.9 Qualitative user feedback

The majority of participants reported more confidence in using INTERACTIVEBAR across all tasks. Some participants mentioned that panning the chart brings bars closer to one another, making it significantly easier to inspect bars and come to a decision. For example, *"Interactive bar chart was the easiest because I had a clear vision of whether the bars were getting smaller or larger."* (P4, TREND IDENTIFICATION), *"with the interactive bar chart, it was possible to change the position of the bars to either not have an interruption in a group of bars or compare the groups/single bars better."* (P33, TREND IDENTIFICATION), *"The interactive bar chart makes it easier by allowing me to more closely align the 2 bars to work out the difference between them."* (P22, PAIRWISE SINGLE VALUE COMPARISON).

Some participants reported more confidence in STATICBAR than STATICPOLAR, citing that the STATICPOLAR looked more confusing and was harder to read. This confirms existing studies that polar charts are generally less effective than linear bar charts [Wal+19; AJB16; Bre+18]. For example, participants mentioned they had to turn their head or neck to spot the answer and static polar charts caused confusion about comparing the bars, and therefore they ranked it the worst, e.g., *"Bar charts are easier in [PAIRWISE GROUP COMPARISON] than circular charts. For both [STATIC and INTERACTIVE] circular charts, I found myself turning my head to the side to get a better look. The standard bar chart, although it went off the side was still easier than both circular."* (P2, PAIRWISE GROUP COMPARISON), and *"It's easier to see it in the bar charts because all the bars are at the same level unlike in circular charts"* (P40, PAIRWISE SINGLE VALUE COMPARISON).

Furthermore, some participants mentioned the ability to rotate allows them to see in different angles and make them feel more confident in the analysis, e.g., *"It is easier for me to compare the lengths when the bars are close together (even if they are not straight lines). It is also easier when the bars are growing upwards (they do not point downwards)."* (P37, PAIRWISE SINGLE VALUE COMPARISON), and *"Overall the interactive circular chart is more helpful in working out the patterns and difference in height of the bars because you can move it so the highlighted bars are at the top and facing clockwise plus the style of the bars makes it easier to notice differences."* (P22, Overall). Some other participants mentioned the polar charts provided a more panoramic view. STATICBAR made it hard to focus when seeing broken ranges or bars that are located far apart.

4.3   Discussion and future work

Overall, the results of our study indicate that interactive cylindrical wrapping leads to significant improvements in error over static representations for TREND IDENTIFICATION and PAIRWISE SINGLE VALUE COMPARISON, despite the additional time required to pan.

Returning to our questions from the Introduction, to answer **[RQ3.1]** *"Does adding interactive wrapping to linear bar charts improve their effectiveness?"*, we begin by noting that contrary to findings from past studies [Wal+19; Bre+18; AJB16] (that static linear bar charts outperform static polar bar charts), we found that for TREND IDENTIFICATION across the 'cut' in linear bar charts, the polar representation is actually significantly better in terms of error rate. However, introducing interaction to linear bar charts reverses this result. We can conclude that the INTERACTIVEBAR clearly outperforme STATICBAR for TREND IDENTIFICATION and PAIRWISE SINGLE VALUE COMPARISON tasks in terms of error, especially for trials where the bars being compared are separated by the 'cut'. These results allow us to reject the null-hypothesis for H3.1-error for some tasks. We found that INTERACTIVEBAR results in less errors than INTERACTIVEPOLAR for some tasks, rejecting the null-hypothesis for H3.2-error and sometimes is significantly faster, rejecting the null-hypothesis for H3.2-time.

While we found that STATICBAR was faster than INTERACTIVEBAR across all tasks except PAIRWISE SINGLE VALUE COMPARISON-Short (rejecting the null-hypothesis for H3.1-time for some tasks), the time spent actually moving the charts was a notable fraction of the trial time. We see from the textured parts of bars in Figure 4.3-TREND

IDENTIFICATION and Figure 4.3-PAIRWISE SINGLE VALUE COMPARISON-*Short* that for INTERACTIVEBAR the interaction time in TREND IDENTIFICATION and PAIRWISE SINGLE VALUE COMPARISON-short tasks were greater than the time difference compared to STATICBAR. We observed that most people use the interaction once, moving INTERACTIVEBAR or STATICPOLAR so that they could comfortably solve the task. While it is tempting to assume the time difference is due to the extra time spent performing the interaction, it is impossible to know whether people are able to actively reason about the visualisation during interaction, suggesting an interesting direction for future study.

We also found that INTERACTIVEBAR significantly outperformed STATICPOLAR in terms of error for TREND IDENTIFICATION and PAIRWISE SINGLE VALUE COMPARISON **[RQ3.2]**. We can thus reject the null-hypothesis for H2-error for these tasks. We conclude that adding interactive wrapping to bar charts—while incurring a cost in terms of the time spent interacting—decreases errors compared to static polar versions which technically do not require rotation to avoid the cut-problem. This trend is similar across all tasks but significant only for two.

Eventually, while we did not find any significant result that interaction reduces errors in polar charts **[RQ3.3]**, we found STATICPOLAR was significantly faster than INTER-ACTIVEPOLAR across all tasks, we therefore accept the null-hypothesis for H3.4-error and reject the null-hypothesis for H3.4-time. We also found that INTERACTIVEBAR was significantly preferred to STATICBAR, STATICPOLAR, and INTERACTIVEPOLAR. We therefore reject the null-hypothesis for P4.1.

In our study, we exclusively focus on evaluating cyclic temporal data in one spatial dimension to keep the crowdsourced study from becoming overly complicated. Based on the study results, we might hypothesise that, for a more complicated case of 2D temporal data exploration in a 2D torus topology, interactive wrapping with two spatial (or temporal) dimensions may still outperform the rotational and wraparound polar chart. In future, we intend to further investigate this hypothesis, which we discussed further in subsection 9.4.3. Another direction is to investigate if the performance benefits afforded by interactive wrapping across the boundaries also applies to other non-temporal cyclic data types, such as geographic maps or flow diagrams with cycles. We would also like to further investigate the aspects of the wrapping method that drive better performance (subsection 9.4.4). For example, we cannot say whether the interactive panning we provided is better than a passive animation of wrapping affording different views. Our feeling, however, is that the interaction gives users a better understanding of the paradigm as well as a sense of control.

In summary, we take these results as strongly encouraging interactive wrapping for linear bar charts, especially since the upfront cost of adding an interactive wrapping technique is small and the interaction is *not* required to read the bar charts.

4.4 Conclusion

We have presented a study comparing the effect of interactive wrapping of bar charts and rotation of polar charts on the readability of real-world cyclical data. Our study confirms the benefits of one-dimensional interactive wrapping on a 2D projected cylindrical topology for cyclical time series, as described in our design space (section 3.3). This is the first study demonstrating that a pannable wrapped visualisation offers significant benefits in terms of error rate over the equivalent static visualisation for reading ranges and comparing values that are split across the edges of the chart, which addresses **RG3** (subsection 1.2.3). Specifically, our study indicates that standard bar charts with interactive wrap panning offers significant benefits over standard unwrapped bar charts in accuracy for reading ranges or value comparison tasks and overall user preference. However, an interactive bar chart is significantly slower than standard bar charts. For polar charts, interactive panning makes them significantly slower and less accurate than static polar charts or bar charts, except for tasks of reading monotonic ranges. In the latter, the standard bar chart is least accurate due to the need to mentally connect discontinued ranges across two ends of the charts.

In the next chapter, we extend the concept of wrapping to two-dimensional and explore a more complex data type – network, with toroidal wrapping.

Interactive Torus Wrapping for Network Visualisations - Smaller Networks

<div style="text-align: right;">

5

</div>

Clutter and confusion are failures of design, not attributes of information.

— **Edward Tufte**
(American statistician and emeritus professor of political science, statistics, and computer science at Yale University.)

Some types of visualisations are fairly straightforward to either wrap or rotate. Examples include bar charts, polar charts, or horizon graphs (described in chapter 3). However, arranging a network on the surface of a torus such that it can be projected to a 2D panel that wraps around horizontally or vertically requires optimisation of node positions on that surface to better show the patterns or relationships within the data.

In this chapter, we investigate visualisations of networks on a two-dimensional torus topology, like an opened-up and flattened doughnut (Figure 2.4). That is, the network is drawn on a rectangular area such that certain links may be "wrapped" across the boundary, allowing for additional spreading of node positions to reduce visual clutter. Apart from crossing reduction as discussed in subsection 2.1.5, toroidal layouts may afford other benefits: e.g., greater angular resolution between links connected to the same node and more uniform link (edge) lengths. We find it interesting that torus drawings of graphs, like those in Figure 2.4(c), have not (to our knowledge), been seriously considered as a practical means of visualising networks.

A well known example for torus wrapping is the game Asteroids (Figure 2.5) where players intuitively understand the torus topology. We consider whether this intuitive understanding is also applicable to network visualisation. However, as we described in Figure 2.4(c), one possible disadvantage of such toroidal drawing is that it requires users to follow wrapped links from one side of the display to the other. We are

inspired by illustrations of tile-display of networks and SOMs (subsubsection 2.1.5) that show connectivity of graphic elements wrapped across the boundary, and the significant improvement of introducing interactive cylindrical wrapping of bar charts in understanding cyclical data (chapter 4). Therefore, this chapter investigates if these wrapping approaches aid comprehension of torus wrapping. Three fundamental research issues that link to **RG4** (subsection 1.2.4) must be addressed if torus layout is to become more widely used.

- **[RQ4.1]**: How to develop layout algorithms that take advantage of the extra flexibility of graph layout on the torus?
- **[RQ4.2]**: How can we best visualise the layout of a node-link diagram on the surface of a torus on a piece of paper or 2D computer monitor?
- **[RQ4.3]**: What, if any, are the perceptual benefits graph layout on a torus has over standard layout on a 2D plane?

To answer these questions, we first present a modified version of a stress minimisation algorithm that supports interactive human-guided layout (subsection 5.1.1) of graphs on the torus (**RQ4.1**). In subsection 5.2.1, we describe graph aesthetic metrics [Pur02], that have been correlated with user preference and performance in readability tasks [War+02]. Number of link-link crossings, the first metric that we can demonstrate can be improved by toroidal layout, is certainly known to affect readability. Another metric that may benefit from a toroidal topology is the minimum angle between links incident on nodes.

In two controlled experiments with 48 participants, we explore if these improvements in metrics permitted by toroidal embeddings lead to an improvement in readability greater than any detrimental effect of, for example, the challenge of following links that wrap around the boundaries of the drawing, and whether providing partial, full context, or interactive panning aide understanding of toroidal graphs (**RQ4.2**), compared with their standard non-wrapped representations (**RQ4.3**).

5.1 ALL-PAIRS Stress Minimising Torus Layout

As described in subsection 2.1.1, for very dense, scale-free, or small-world graphs, standard force-directed methods generate node-link layouts that are difficult to read.

Force-directed methods are the most commonly-used layout algorithms for general purpose graph visualisation. These methods find a layout by minimising an objective
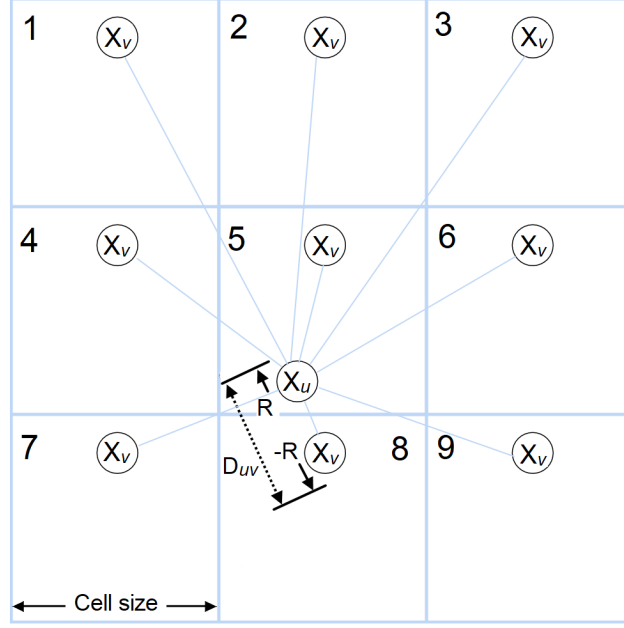
function, such as the standard *stress* function, based on differences between ideal and the actual distances in a low-dimensional coordinate system [BP08; Dwy09; ZPG18; Dev+19]. We adapt the stress function for 2D toroidally-wrapped topology. We consider the positions of each pair of nodes in a 3×3 repeated tiling, with equal square *cell size* as shown in Figure 5.1, which is the length of each square tile. Each node is considered to have nine positions, with the same offset position within every cell. Like ALL-PAIRS, for each pair of nodes, we compute the gradient information across the nine possible ways to consider their adjacency. Thus, we have the following definitions for stress for conventional unwrapped graphs NOTORUS, and then for toroidal wrapped graphs TORUS. Given a graph G with nodes V, we define:

$$
stress = \begin{cases} \sum_{(u,v)\in V \times V, u \neq v} \frac{(L \times D_{uv} - d_{uv})^2}{(L \times D_{uv})^2} & \text{NOTORUS} \\ \sum_{(u,v)\in V \times V, u \neq v} min_{w \in W} \frac{(L \times D_{uv} - d_{uvw})^2}{(L \times D_{uv})^2} & \text{TORUS} \end{cases} \tag{5.1}
$$

For a pair of nodes (u, v), d_{uv} is the Euclidean distance between u and v, D_{uv} is the shortest graph-theoretic path length between them. In TORUS, $w \in \{1 \dots 9\}$ selects one of the nine possible adjacencies which informs the wrapping as described previously, such that each d_{uvw} is the actual (Euclidean) distance of (u, v) between

the centre cell and adjacent cell w. For both NOTORUS and TORUS, the term $\frac{1}{L \times D_{uv}^2}$ is used to penalise long-range attraction. The constant L indicates the desired link length in the resulting layout.

Various methods for minimising *stress* are used, detailed in subsection 2.1.8. We follow Dwyer et al. [DMW08] in using a gradient descent approach. Our approach is implemented as a modification of the *WebCoLa* [15] implementation of the algorithm from Dwyer et al. [DMW08]. Note, that this technique moves all pairs of nodes at a time by taking the partial derivatives across all pairs of nodes, summed to give a gradient vector across all node positions, and thus the placement of nodes occurs for all pairs of nodes.

In this thesis, we name this approach ALL-PAIRS[1]. Meanwhile, the ALL-PAIRS approach using *WebCoLa* also allows us to add constraints to avoid overlapping node labels.

We take a straightforward approach to adapting the goal function (1) to the torus. At each iteration, we compute gradient contributions of each node, not just with respect to each other node, but with respect to the positions of each other node in the eight cells adjacent to the central cell in a 3×3 tiling corresponding to the possible adjacencies in a torus topology, as seen in Figure 5.1. The positions of each node within each cell are computed by simple offset, i.e. in Figure 5.1 the position of node X_v in the top-left cell is simply X_v's x, y position in the central cell, minus the cell size, and so on for the other cells. This results in nine different sets of gradient contributions.

From these nine choices of gradient components for each node, we compute the one that results in the largest reduction of stress, and take this to be the entry in the gradient vector for that node. The resultant descent vector (computed from the gradient and a step size from the corresponding second derivative information, as per [DMW08]) has horizontal and vertical components for each node with contributions from all other nodes e.g. for node X_u in Figure 5.1. If the descent vector takes a node beyond the cell boundaries, then the position for that node wrap around, as per asteroids rules.

---

[1]We name our simple interactive layout approach ALL-PAIRS, given that it minimises the stress function by moving all pairs of nodes at a time using gradient descent (an optimisation method for finding a local minimum of a differentiable function), in contrast to a different approach, named PAIRWISE, which we explore in chapter 6 that minimises the stress function by moving a single pair of nodes at a time using gradient descent.

5.1.1 Interactive layout

As with all gradient descent approaches to optimisation of non-convex functions, it can happen that the layout can converge to a configuration corresponding to a poor local minimum of the stress function. In future, new approaches for initialising the layout to a better state may autonomously give a high-quality trade-off between aesthetic measures. For the layouts used in our study stimuli, however, we achieve this with an interactive, human-guided approach, made possible by our iterative stress minimisation approach. A user can drag nodes while layout is proceeding which can be useful to guide the layout away from poor local minima, but also to explore different configurations of torus layout and the different symmetries that they reveal, as seen in Figure 5.3. This interactivity was useful in preparing stimuli for our controlled study (section 5.2), which is not intended as an evaluation of the automatic layout algorithm, but rather an evaluation of the torus drawing paradigm in general as per **[RQ4.3]**, and for which we needed to control for the various aesthetic measures.

In summary, a human-guided stress minimising approach to generate a torus layout is as follows[2].

1. Automatically obtain an initial layout, starting with all nodes at the centre of the centre cell in a 3×3 grid.
2. Of the nine possible wrappings for each pair of nodes, choose the wrapping such that the torus distance is closest to the ideal (graph-theoretic related) distance, Figure 5.1.
3. For each pair of nodes, compute the partial derivative of the stress function for the chosen wrapping for that pair.
4. As per [DMW08], we sum the partial derivatives across all pairs to give a gradient vector across all node positions.
5. Also per [DMW08] we use second derivative information to compute an optimal descent vector in the gradient direction.
6. After moving all nodes according to the descent vector some may have moved outside of the centre cell. These are translated back to the centre cell (e.g., by moving X_v in Figure 5.1 to its corresponding position in cell 5).
7. The above steps are repeated until convergence to a local minimum, i.e. movement falls below a predefined threshold.
8. User drags a node to a desired position.
9. The layout algorithm automatically adjusts the positions of all other nodes using aforementioned steps 2-7.

---

[2]The detailed pseudocode to our method is available from `https://github.com/Kun-Ting/WebCola`

5.2 Comparing Torus and non-Torus Drawings

In this section, we compare different ways to render torus drawings of graphs with standard non-toroidal node-link diagram representations. We do this using both *i)* established metrics for assessing network layout aesthetic quality, and *ii)* some new metrics specific to torus drawings, as described below. We then prepared a corpus of 72 graphs using standard graph generation techniques that simulate real-world graphs as described in Section *Graph Corpus*. These were then laid out to balance the aesthetic measures using our interactive layout for both toroidal and non-toroidal conditions. Finally, we prepared three different styles of renderings of these graph layouts for use as stimuli in our two controlled studies.

5.2.1 Standard layout aesthetics measures

The diagrams in our graph corpus were arranged to find a balance of a number of aesthetic measures. For a graph with node set V and edge (link) set E we follow past work in considering the following graph layout aesthetics:

Link Length Variance—Graphs with more uniform link length have been found to be preferred by readers [Dwy+09]. We compute link length variance by first scaling the graph layout such that the average link length is 1. Thereafter, we take the Link Length Variance as the average squared deviation from 1 across all links E.

$$\frac{1}{|E|} \sum_{e \in E} (1 - l_e)^2 \tag{5.2}$$

where l_e is the length of link e in the scaled drawing.

Minimum Angle—For a node v we take the ideal angle between adjacent links incident to v to be $\theta_v = \frac{360°}{deg(v)}$, where $deg(v)$ is the number of incident links on v. Then, our Minimum Angle metric, is the average difference between this ideal angle of incidence and the actual minimum angle of incidence for links incident to v in the drawing:

$$\frac{1}{|V|} \sum_{v \in V} \frac{|\theta_v - min\theta_v|}{\theta_v} \tag{5.3}$$

Link-link Crossings—The adverse effect of link-link crossings on readability of graphs is well studied [HEH09]. Since we consider only straight-line drawings link-link crossings is a straightforward count of the number of times in each diagram that the line segments for pairs of links intersect. In all of the diagrams in our corpus

we generally tried to keep crossing counts low by steering the stress minimisation algorithm, but truly crossing minimal drawings with straight-line links are difficult; first, to identify, especially in larger graphs, and second—especially for the non-torus condition—it is difficult to nudge the stress layout into a state that leads to poor link length variance. In general, we did not want to overly prioritise link-link crossing minimisation over the other metrics, as studies such as [Kie+15] indicate a balance of metrics is required.

Past work has found node-node overlaps and link-node crossings significantly worsen graph readability [Ben+07]. As per [DKM06], we include constraints in late iterations of the stress minimisation algorithm to prevent node overlaps. There are automatic processes for removing link-node crossings [Sim+11; DMW08]. For the layouts in our diagram corpus, we remove node-link crossings through a simple manual nudging, taking care not to overly disturb the other metrics.

5.2.2 Torus layout aesthetics measures

We define the following two measures for torus layouts, specifically.

Left-right and Top-bottom Wrapping—Straight-line links lead the readers' eye in an undemanding way from one node to its neighbours. However, where the links wrap around the links of a torus drawing, it seems reasonable to assume that finding the continuation of the link on the other side of the diagram is a more demanding task. We report left-right and top-bottom wrapping separately, as we have controlled the stimuli in our study such that their numbers are similar overall.

Corner Wrapping—When diagonal links wrap near the corners of a torus diagram the link must be split into three segments, as per the example to the right (Figure 5.2). In pilots for our first study we quickly realised that such corner-wrapping is especially confusing to readers. As described in Study 1, we therefore introduced additional training material to prepare participants for this case and controlled for corner-wrapping while generating our stimuli.

5.3 Graph Corpus

In this section we describe the test corpus of graphs, generated as described below, that we used in our studies. The layouts were generated using the human-guided
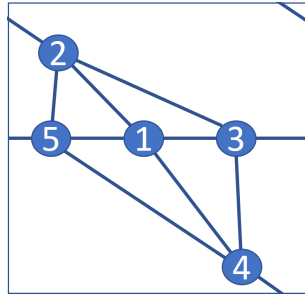
Fig. 5.2.: Corner wrapping

a   b   c   d
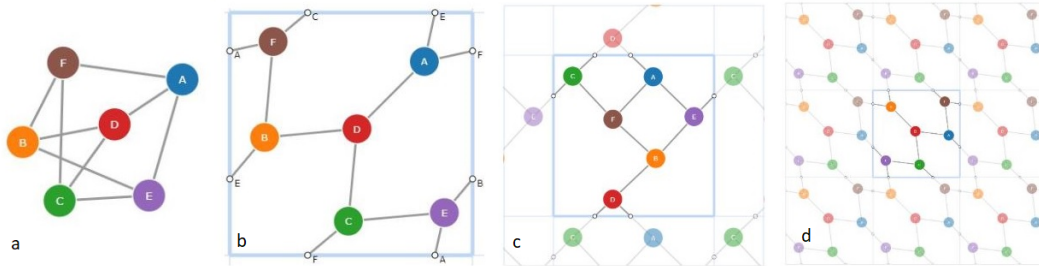
Fig. 5.3.: The complete bi-partite k3,3 graph drawn using the four different techniques considered in this chapter: (a) NOTORUS; (b) NOCONTEXT; (c) PARTIALCONTEXT; (d) FULLCONTEXT

stress minimising approach as described above and *WebCoLa* [15] layout for non-torus graphs, also with some manual adjustment. In general we tried to find layouts that balanced the crossings, minimum angle, and link length variance statistics. However, there is a tradeoff. For example, in some cases fewer crossings may be possible (as in Figure 5.3(a)), but it would come at a great cost to the other measures. As can be seen from the table, the torus embedding permitted significantly better layouts in terms of all three of these measures. However, it came at the new cost of wrapping. Table 5.4 provides summary statistics for these aesthetic measures.

Layout	Number of Graphs	Average number of crossings	Average link length variance	Average incidence angle	Average number of wrappings	Average number of corner wrappings	Average number of answers on wrappings	Average answers on corner wrappings
Small-NoTorus	8	5.875	0.041	0.662	0	0	0	0
Small-Torus	24	0	0.023	0.376	2.729	0.25	5.625	0.25
Medium-NoTorus	6	13	0.05	0.65	0	0	0	0
Medium-Torus	18	0.778	0.035	0.429	3.917	0.833	3.5	0.5
Large-NoTorus	4	19.75	0.054	0.642	0	0	0	0
Large-Torus	12	4	0.049	0.479	4.75	1	3	0

Fig. 5.4.: Average number of crossings, link length variance, and incidence angle. Total number of torus wrappings and number of answers requiring a wrapping.
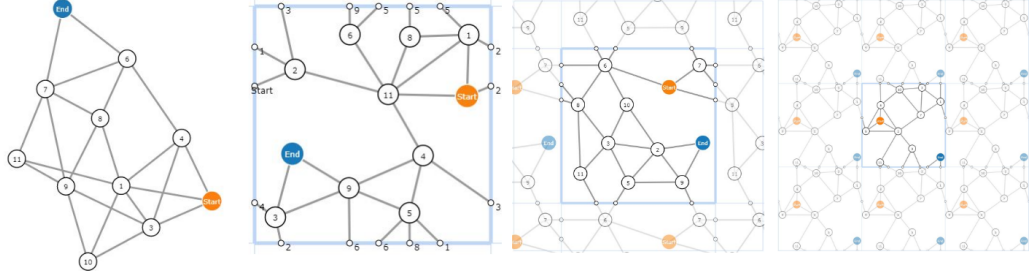
Fig. 5.5.: Example graph for our studies for MEDIUM. Techniques left-to-right are NOTORUS, NOCONTEXT, PARTIALCONTEXT and FULLCONTEXT.
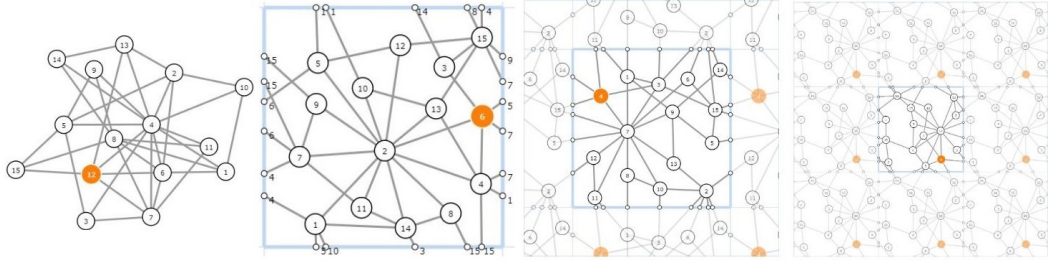
Fig. 5.6.: Example graph for our studies for LARGE. Techniques left-to-right are NOTORUS, NOCONTEXT, PARTIALCONTEXT and FULLCONTEXT.

We generated a sample corpus of 18 graphs rendered using the four different techniques for a total of 72 torus and non-torus drawings. The graphs were generated using algorithms designed to simulate real-world social networks and biological networks, using generators from NetworkX [Pac19]. The majority of the graphs we used are Watts-Strogatz's small-world and Barabasi-Albert's scale-free graphs. We also used Erdos-Renyi's binomial network model to generate networks that balance the total number of wrappings. Therefore, we used one scale-free, small-world, and binomial model generator for creating graph instances for SMALL class. For MEDIUM class, we used two scale-free generators and one small-world generator , an example is shown in Figure 5.5. For LARGE class, we used two scale-free generators and one small-world generator, an example is shown in Figure 5.6.

Each class (SMALL, MEDIUM, LARGE) had a constant number of nodes (8, 11, and 15, respectively) and number of links in a tight range (12–18, 18–28, and 26–36).

5.4 Torus Rendering Techniques

In this section we comparatively evaluate four different visual representations for node-link diagrams. An example of each representation applied to the complete bi-partite graph known as *k3,3* is shown in Figure 5.3.

- NOTORUS is a conventional non-toroidal node-link diagram as laid out by a force-directed technique (WebCoLa [15]) with manual refinements to balance metrics, Fig. 5.3(a).
- NOCONTEXT is a torus drawing without any contextual tiling, as described below, Fig. 5.3(b). Note that the torus drawings used in our studies include labels at the boundaries indicating to which nodes wrapped links are connected. The need for such labelling in torus drawings without additional context became clear from the pilots for our first study.
- PARTIALCONTEXT shows only partial context, that is, we see part of a repeated 3×3 tiling to show some torus adjacencies, Fig. 5.3(c).
- FULLCONTEXT shows the full 3×3 layout tiles (context) of a torus drawing, Fig. 5.3(d).

Note that the diagrams including context (PARTIALCONTEXT and FULLCONTEXT) are scaled to the same size in the figures in this paper, but in our studies they were shown to participants so that the central cell is the same size as the NOCONTEXT drawing; i.e., the area of FULLCONTEXT drawings shown to study participants were nine times that of NOCONTEXT drawings.

In Study 1 we evaluate static versions of these four representations. In Study 2 we introduce a simple panning interaction for the torus rendering techniques, giving us three more conditions: NOCONTEXT-PAN, PARTIALCONTEXT-PAN, and FULLCONTEXT-PAN.

5.5 User Study 2: Static Torus Drawing Readability

In this study, we investigate if our torus drawings are effective to solve low-level network analysis tasks and if they are more efficient than baseline drawings of node-link diagrams without torus link wrapping. Thus the techniques in our study were exactly the same as described in the previous section. We were also interested in subjective user feedback and if the amount of visual *context* shown has an impact on effectiveness and efficiency of torus drawings.

5.5.1 Tasks

There are many tasks that users typically perform while analysing network visualisations [Lee+06]. For our study, we selected representative link and path following tasks for network analysis [Lee+06] whose performance we believed would be influenced by torus link wrapping.

- SHORTESTPATH: **What is the shortest path in terms of number of links that need to be traversed between the nodes labelled *Start* and *End*?** Participants had to count the number of links between the marked nodes. We recorded participants' responses with multiple-choice questions with 8 options with answers of similar length but varying length.
- NEIGHBORS: **Identify all the friends (neighbouring nodes) of the orange node.** We recorded participants' responses with multiple-choice questions with 8 options with similar ordered lists of nodes of various length.
- NODECOUNT: **Identify the total number of people (nodes) in the network**. To answer, we provided a slider with a range of 1 to 20. Error rate was obtained by calculating the absolute difference between participant's answers and actual answers divided by the actual answers.
- LINKCOUNT: **Identify the total number of relationships (links) of the network**. To answer, we provided a slider with a range of 1 to 20. Error rate was calculated the same way as for NODECOUNT.

5.5.2 Data sets

For all tasks, we chose graphs from one of the 18 graphs generated as described in Section *Graph Corpus*. Out of 18 trials in each condition in our study, there were 3 SMALL, 3 MEDIUM, 2 LARGE graphs (difficulty levels as defined above) used in SHORTESTPATH tasks, 3 SMALL, 3 MEDIUM, 2 LARGE in NEIGHBORS tasks, 1 SMALL graph in NODECOUNT, and 1 SMALL graph in LINKCOUNT. While the graph size (up to 15 nodes, 36 links) is relatively small, as pointed out in [Dwy+09; Dwy+08; Kie+15], such small graphs already present a significant challenge to readability and present a suitable level of difficulty for path-following tasks in a study.

5.5.3 Hypotheses

Our predictions were pre-registered with the Open Science Foundation: `https://osf.io/2e6bm`.

Effect of layout

- **L4.1**: FULLCONTEXT has better task effectiveness (in terms of participant speed and error) than NOCONTEXT, and PARTIALCONTEXT across all task difficulties ([RQ4.2]).
- **L4.2**: FULLCONTEXT has better task effectiveness than NOTORUS (involves too many link crossings), NOCONTEXT (requires too much mental wrapping), and PARTIALCONTEXT (requires certain mental wrapping) for difficult tasks ([RQ4.2][RQ4.3]).
- **L4.3**: PARTIALCONTEXT has better task effectiveness than NOCONTEXT across all task difficulties ([RQ4.2]).
- **L4.4**: NOTORUS has better (participant reported) task effectiveness than FULLCONTEXT for SMALL and MEDIUM graphs (**RQ4.3**).

Effect of tasks

- **T4.1**: Participants will perform better (in terms of participant speed and error) using FULLCONTEXT than NOCONTEXT and PARTIALCONTEXT on SHORTESTPATH and NEIGHBORS (**RQ4.2**).

Effect of size of graphs

- **D4.1**: Participants will perform better using FULLCONTEXT across all task difficulties than NOCONTEXT and PARTIALCONTEXT (**RQ4.2**).

Participant Preference

- **P4.1**: Participants will prefer FULLCONTEXT over either NOCONTEXT or PARTIALCONTEXT ([RQ4.2]).
- **P4.2**: Participants will prefer PARTIALCONTEXT over NOCONTEXT ([RQ4.2]).
- **P4.3**: Participants will report more confidence in using FULLCONTEXT than NOCONTEXT and PARTIALCONTEXT ([RQ4.2]).
- **P4.4**: Participants will report more confidence in using NOTORUS than FULLCONTEXT ([RQ4.3]).

5.5.4 Participants

We recruited 24 participants through email and snowball from our institute. 6 people were female while 18 were male. There were 2 participants aged below 20, 16 between 20-30, 5 between 30-40, and 1 greater than 50. While 5 participants never see social network diagrams, the rest 19 responded they either seldom (15) or often (4) see network diagrams from a course, textbook, or Internet. In pilot studies,

we found participants were confused about where the data points and link connect when they were split across the boundaries, especially for low-level path following tasks. Therefore, we opt for a controlled lab-based study where the experimenter went through the tutorials with participants and supervised the study.

5.5.5 Design and procedure

We opted for a within-subject design study with 3 factors: 4 techniques × 3 difficulty levels × 4 tasks. We used a full-factorial design to balance for the 4 techniques (24 orderings). We used 18 trials with the number of graphs in each difficulty level (Table Figure 5.4) + 10 training trials = 28 trials in each technique. For each technique, participants progressed through the same order of graph sizes from SMALL, MEDIUM to LARGE. Participants sat in front of a Dell 22-inch LCD screen. To collect visual focus of each representation, we equipped the Tobii-pro X3-120 eye-tracking system to record visual focus of participants doing trials. We used a laptop with Intel I5 8350U (1.7GHz), Intel UHD Graphics 620 to run the website and control experiment.

5.5.6 Results

All of the 24 participants completed 72 trials in the real (non-training) question set. They answered all the questions in tutorial and training sections correctly before entering the real question set. Therefore, we recorded the performance of 1,728 trials. There were 29 significant differences in Study 2 as summarised below, based on 95% confidence levels. Since the distribution of the logarithm of completion time of each condition followed normal distribution, we used ANOVA repeated measures and Tukey's post-hoc pairwise comparison to test significance. Since the distribution of error rate and subjective rank of each condition did *not* follow normal distribution, we used Friedman's non-parametric test to test null-hypothesis and Nemenyi's post-hoc pairwise comparison to test significant difference. Significant differences are summarised in Figure 5.7(error and time), and 5.8(participant preferences). Graphics with detailed results are found in Figure 5.9.

For SHORTESTPATH and LINKCOUNT, our most significant results showed the following:

- NOTORUS significantly outperformed NOCONTEXT in error and time independent of size of graphs (as seen in Figure 5.7(a), Figure 5.7(c), box plots in Figure 5.9(a),
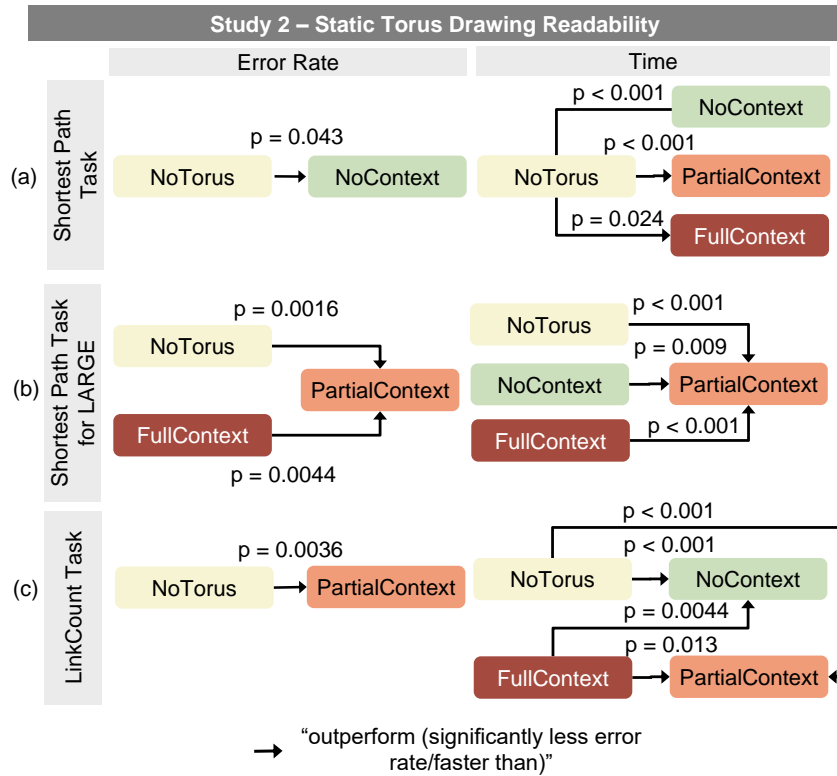
Fig. 5.7.: Study 2: Statistical results of performance comparisons between NoTorus and Torus drawings under 95% confidence level.

Fig. 5.8.: Statistical results of preference differences of Study 2 (95% confidence level).

Fig. 5.9.: Study 2: Results for error, time, and subjective user rank by task. Higher rank indicates stronger preference. Dotted lines indicate significant differences for $p < .05$. The best significant results are highlighted in the border of bars.

Figure 5.9(e), Figure 5.9(g)) and outperformed PARTIALCONTEXT (as seen in Figure 5.7(a), Figure 5.7(c), with box plots in Figure 5.9(c), Figure 5.9(e), Figure 5.9(g));

- FULLCONTEXT significantly outperformed PARTIALCONTEXT in time independent of size of graphs (as seen in Figure 5.7(c), with its box plot in Figure 5.9(g));
- For SHORTESTPATH by LARGE graphs, NOTORUS and FULLCONTEXT both significantly outperformed PARTIALCONTEXT in error and time (as seen in Figure 5.7(b), with box plots in Figure 5.9(d), Figure 5.9(h));
- Compared to NOCONTEXT, participants found it easier and reported greater confidence using both NOTORUS and FULLCONTEXT (as seen in Figure 5.8(a), Figure 5.8(b), with box plots in Figure 5.9(i), Figure 5.9(l)). NOTORUS was significantly preferred over PARTIALCONTEXT (as seen in Figure 5.8(d) with its box plot in Figure 5.9(k));

For **NEIGHBORS**, we found

- NOCONTEXT and FULLCONTEXT were significantly preferred over NOTORUS (Figure 5.8(c), 5.9(j)).
- This correlates with a weak trend for NOCONTEXT to outperform NOTORUS in error rate ($p = 0.23$) independent of size of graph, as seen in Figure 5.9(b). The results of performance by graph size are omitted for space limitations and can be found in the study material web link.

Based on participants' qualitative feedback, they used NOCONTEXT with labelled display to quickly identify neighbours of a chosen node. NOCONTEXT drawing was (surprisingly to us) the most preferred torus display.

For **NODECOUNT**, results did *not* show any significant difference between techniques and most participants could correctly identify the number of nodes in all conditions, which suggests that most participants were not confused by the torus wrapping. The result is omitted and can be found in the study material web link.

Based on these results, we *rejected* the following hypotheses for [RQ4.2]: L4.3, T4.1 (for NEIGHBORS), P4.2, P4.4, and for [RQ4.3]: L4.2, L4.4 (for SMALL), P4.4. We accepted hypotheses for [RQ4.2]: L4.1, L4.2, T4.1 (for SHORTESTPATH), D4.1, P4.1, P4.3 and for [RQ4.3]: L4.4 (for MEDIUM).

5.5.7 Qualitative user feedback

The majority of participants mentioned that their preferences were dominated by the technique, independent from task and graph size. Participants favoured

FULLCONTEXT for SHORTESTPATH over NOCONTEXT because it helped understanding link wrapping and provided a great overview over the network. At the same time, they liked the cleanness of NOCONTEXT which has no repetition and extra information, which allowed them to concentrate on the task.

5.5.8 Summary

In our first study, we generally found an improvement to torus drawing by adding full context to torus drawing. Our results can be summarised as follows. For **[RQ4.3]** (What, if any, are the perceptual benefits graph layout on a torus has over standard layout on a 2D plane?), FULLCONTEXT is as good as NOTORUS for SHORTESTPATH-LARGE, NODECOUNT and LINKCOUNT. Static NOCONTEXT or PARTIALCONTEXT torus is clearly worse than NOTORUS. For **[RQ4.2]** (How can we best visualise the layout of a node-link diagram on the surface of a torus on a piece of paper or 2D computer monitor?), FULLCONTEXT is the best static torus layout. Static NOCONTEXT or PARTIALCONTEXT is the worst torus layout. NOCONTEXT had a tendency (confidence intervals $\leq 77\%$) to perform better in accuracy than NOTORUS, independently of graph size for NEIGHBORS. For NEIGHBORS, NOCONTEXT and FULLCONTEXT were both significantly preferred over NOTORUS in subjective user ranking (**[RQ4.3]**). Participants indicated that the NOCONTEXT technique appeared cleaner and would be better suited to show larger graphs.

The role of context is crucial yet not entirely conclusive from our results. Result 3 above, regarding preference for NOCONTEXT over torus drawings with context, implies that the redundant information disturbs users, even as it assists them (as per Result 1 and 2). Moreover, context requires screen space. We therefore designed a 2nd study (section 5.6) to investigate the effect of interactive panning across the three torus representations.

5.6 User Study 3: Torus drawings + Panning

For our second study, we repeat the evaluation of the three torus conditions from Study 2 (NOCONTEXT, PARTIALCONTEXT, FULLCONTEXT): the visual representation in each of these techniques was exactly the same (e.g. Figure 5.5 and Figure 5.6). The sole interaction was panning, using the mouse or touch drags. Since link wrapping is not applicable to the node link representation we do not repeat trials for NOTORUS, but instead perform a between-groups analysis of the Study 2 results

for this condition, with those in Study 3. Tasks and graphs were also the same as in Study 2.

5.6.1 Hypotheses

Our predictions were pre-registered with the Open Science Foundation: `https://osf.io/v3756`.

- **I4.1**: NoContext-Pan has better task effectiveness than NoContext (**RQ4.2**).
- **I4.2**: PartialContext-Pan has better task effectiveness than PartialContext (**RQ4.2**).
- **P4.5**: Participants will prefer NoContext-Pan to FullContext-Pan (**RQ4.2**).

5.6.2 Participants, design, and procedure

We recruited a new set of 24 participants from a local institute through email and snowball, none of which had participated in Study 2. One person was aged below 20, 17 between 20-30, 4 between 30-40, and 2 greater than 40. All of the participants never or seldom see social network diagrams. Similar to Study 2, we used a within-subject design study with 3 factors: 3 techniques, 18 graphs, and 4 tasks. We performed a between-subject study comparing Interactive panning as in this study to the results of no-panning in Study 2. We use the same trials as previous user study and add interactive panning to the trials, using the mouse.

5.6.3 Results

All 24 participants successfully went through 54 trials in the real question set. They answered all the questions in the tutorial and training sections correctly before entering the real question set. Therefore we recorded a performance of 1,296 trials.

To test for significance, of panning on performance between panning and no-panning conditions, we used paired t-test. We used Wilcoxon's signed rank test to test significance difference in error rate between 2 groups. To test differences in performance between NoTorus in Study 3 and torus with panning, we opted for ANOVA independent measures and Tukey's post-hoc pairwise comparison, as the logarithm
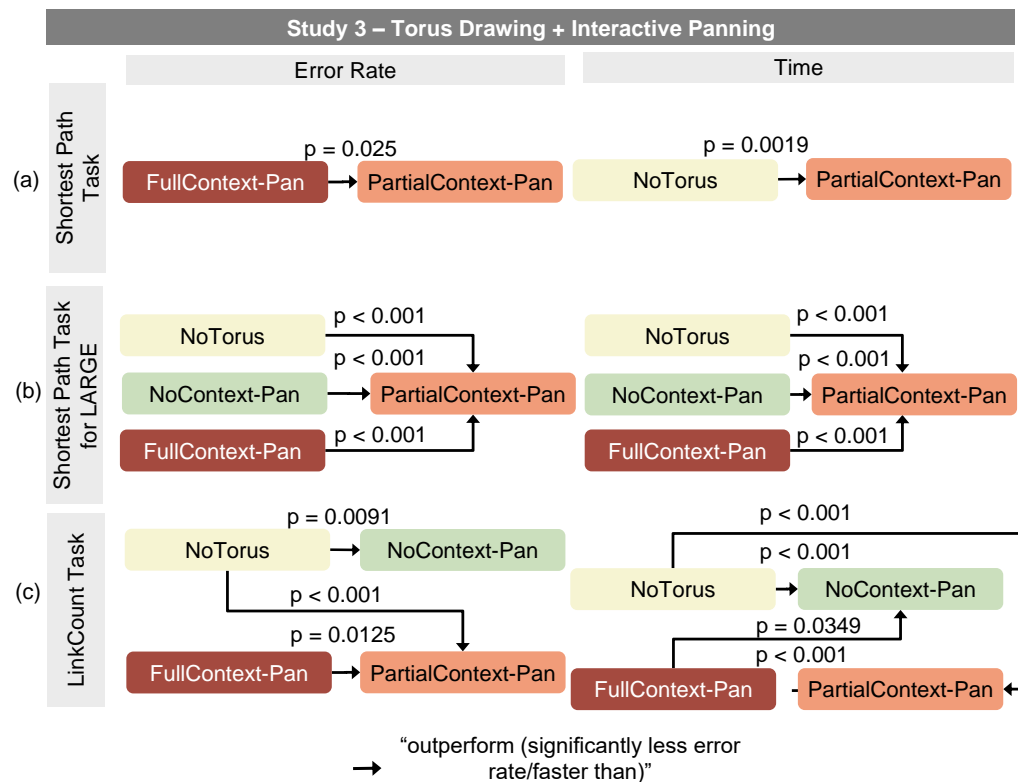
Fig. 5.10.: Study 3: Statistical results for performance of NoTorus and the torus+panning conditions, with 95% confidence level.
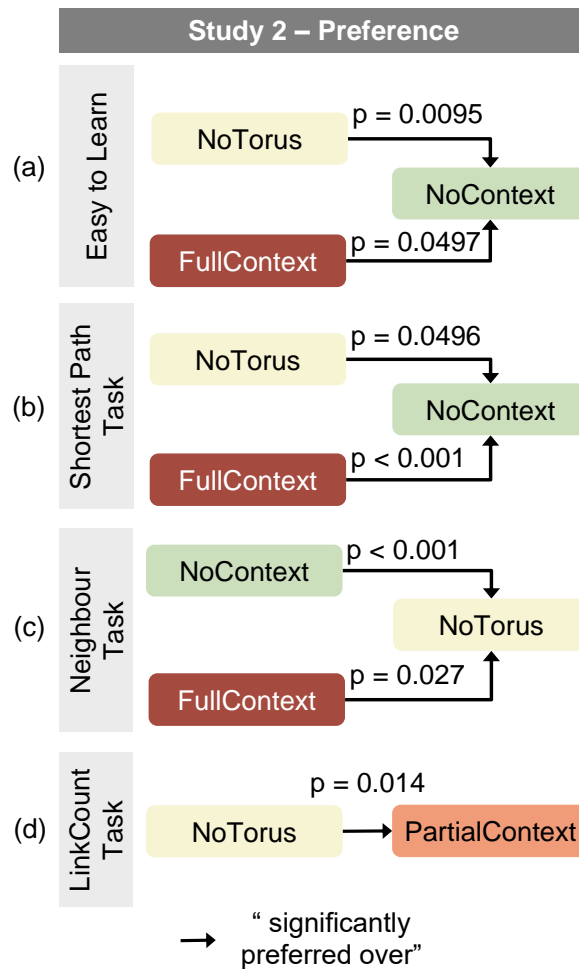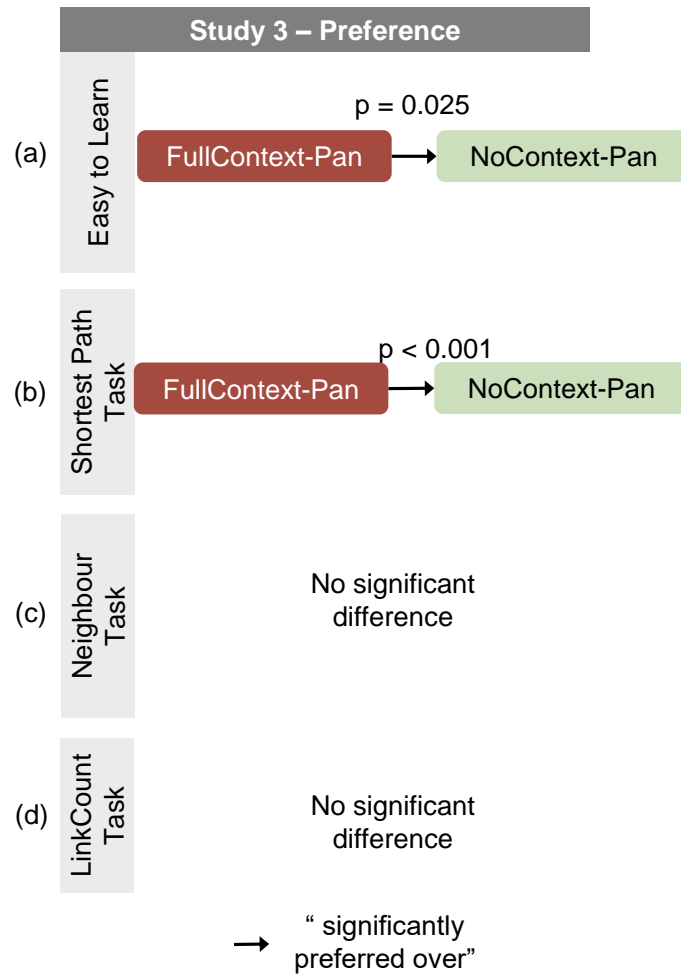
Fig. 5.11.: Statistical results of preference differences of Study 3 (95% confidence level).
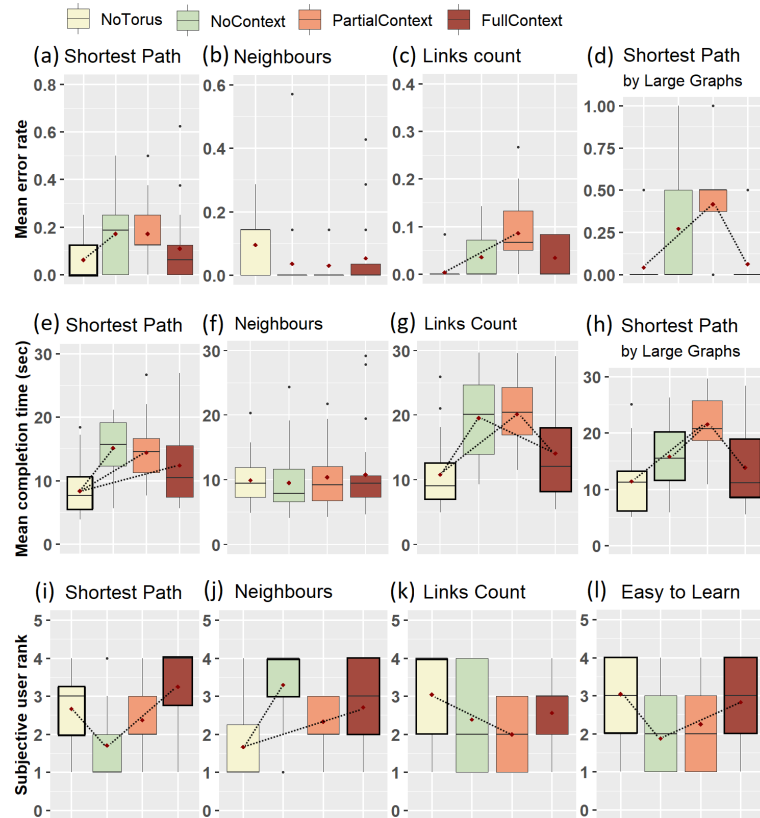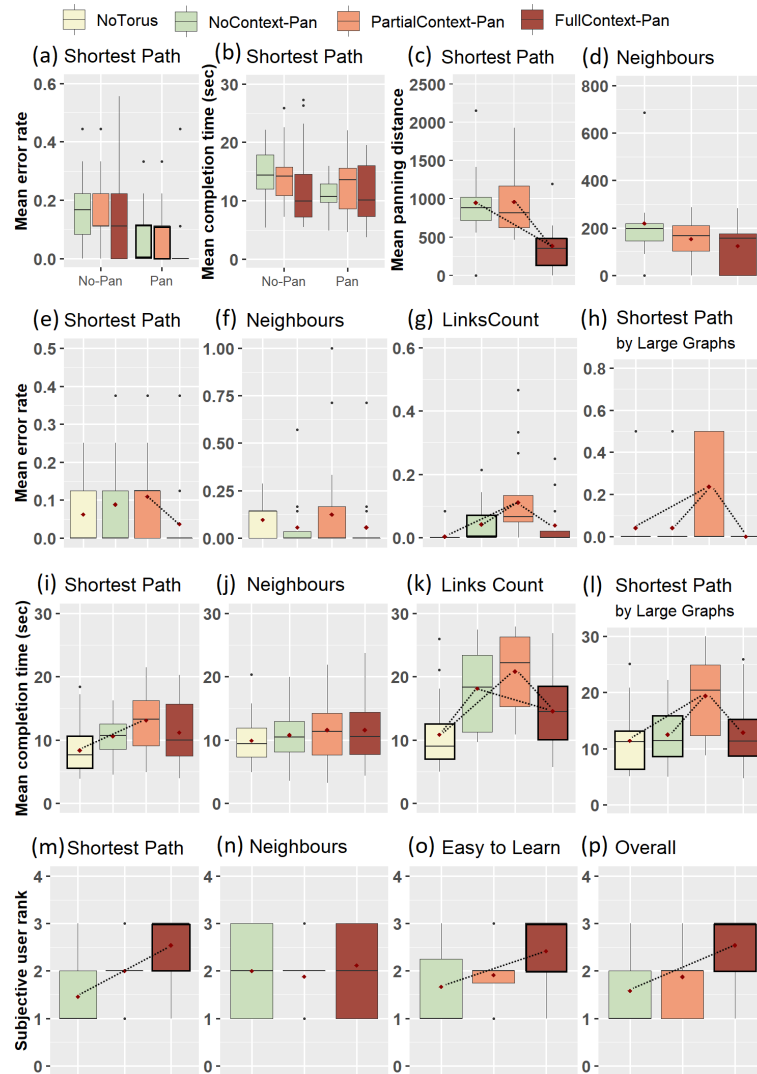
Fig. 5.12.: Study 3: Results for error, time, panning distance, and subjective user rank by task. Higher rank indicates stronger preference. Dotted lines indicate significant differences for p < 0.05. The best significant results are highlighted in the border of bars.
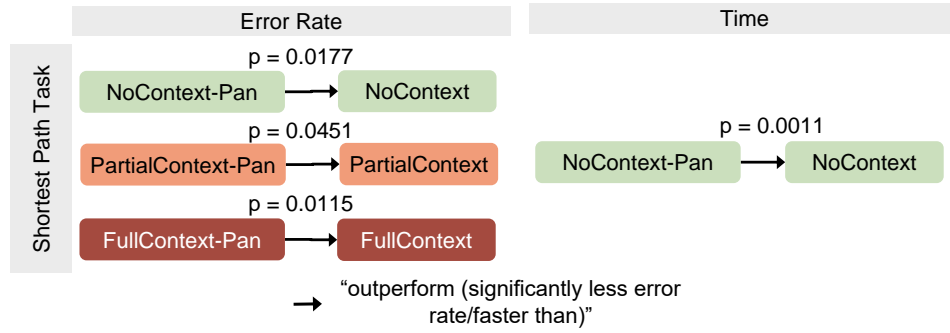
Fig. 5.13.: Statistical results of performance comparisons between groups of static torus and interactive panning, under 95% confidence level.

of completion time followed normal distribution. We used Kruskal-Wallis's non-parametric test and Wilcoxon multiple pairwise comparisons to test error. To test panning distance between layout conditions we used ANOVA RM and Tukey's post-hoc pairwise comparison. Significant differences are shown in Figure 5.13 (between groups of static and interactive panning), Figure 5.10(for indiviual techniques) and Figure 5.11 (subjective participant rankings). Graphics with detailed perfomance results are found in Figure 5.12.

In contrast to Study 2, where there were significant differences between no-context and full-context torus or standard layout representations, with panning, we did not find any significant differences between no-context and full-context or standard layout, independent of graph size for SHORTESTPATH task. This suggests that panning makes no-context torus techniques equally performant to full-context or standard layout, as participants pan more in the conditions where less context is given.

We found significant differences in the panning conditions for **SHORTESTPATH**:

- Independently of graph size, all torus panning significantly outperformed their non-panning counterpart in error (as shown in Figure 5.13, Figure 5.12(a)). NOCONTEXT-PAN significantly outperformed NOCONTEXT without panning in time (as shown in Figure 5.13, Figure 5.12(b));
- NOCONTEXT-PAN and PARTIALCONTEXT-PAN were panned significantly more than FULLCONTEXT-PAN, as seen in Figure 5.12(c);
- NOTORUS significantly outperformed PARTIALCONTEXT-PAN in time (as shown in Figure 5.10(a), Figure 5.12(i)), and FULLCONTEXT-PAN significantly outperformed PARTIALCONTEXT-PAN in error (as shown in Figure 5.10(a), Figure 5.12(e)) independently of size of graphs.

- With panning, NOCONTEXT-PAN, PARTIALCONTEXT-PAN and FULLCONTEXT-PAN all significantly outperformed PARTIALCONTEXT-PAN in error and time by LARGE (as seen in Figure 5.10(c), Figure 5.12(h), Figure 5.12(l));
- Compared to NOCONTEXT-PAN, participants found it easier, and reported greater confidence in using FULLCONTEXT-PAN (as seen Figure 5.11(a), Figure 5.11(b), Figure 5.12(m), Figure 5.12(o)). Overall, FULLCONTEXT-PAN was significantly ($p = 0.0026$) preferred over NOCONTEXT-PAN, as seen in Figure 5.12(p);

For NEIGHBORS, we did not find significant differences between any of the torus techniques, but for LINKCOUNT, both NOTORUS and FULLCONTEXT-PAN significantly outperformed NOCONTEXT-PAN and PARTIALCONTEXT-PAN in error and time (as shown in Figure 5.10(b), Figure 5.12(g), Figure 5.12(k)), but FULLCONTEXT-PAN outperformed NOCONTEXT-PAN only with significant support in terms of time.

Based on these results, we rejected P4.5 and accepted I4.1 and partially accepted I4.2 (with significant support in terms of Error).

5.7 Discussion and Limitation

With respect to **[RQ4.3]**, our most significant results are 1) full-context or pannable full-context and no-context torus is as good as (not demonstrably different from) standard layout; 2) static or pannable partial-context torus is, despite being a drawing style most commonly used in the literature, e.g. by Kocay et al. [KK16] (Figure 2.6), the worst torus layout and clearly worse than standard non-wrapped layout. For **[RQ4.2]**, 1) full-context is the best static torus layout; 2) no-context + pan is as good as (not demonstrably different from) full-context.

Overall, we found that while torus topology allowed for significant improvements in the standard graph layout aesthetic measures (as per Figure 5.4), the wrapping of links at the sides of the diagram imposes a significant cost in terms of speed and accuracy in tasks requiring users to follow links (as demonstrated in section 5.5). However, this cost is mitigated by providing either more wrapped context in static diagrams (section 5.5), or by allowing interactive panning (section 5.6). Of the three torus representations shown in section 5.6, the NOCONTEXT-PAN representation seems to have shown the greatest benefit from the introduction of panning, and indeed panning was used more in the NOCONTEXT-PAN condition than FULLCONTEXT-PAN. It is interesting that different levels of replicated context (no, partial, full) have significant differences on torus readability. As a majority of users commented that

they like the overview provided by full-context, it may be interesting to explore gaze distribution in different levels of context in future.

There are also limitations. First, while we focus on precise graph readability and path/link following, the small graphs tested (≤ 15 nodes, ≤ 36 links) are not really representative of real-world networks of interest in many domains, such as social and biological networks. We recognise that testing other types of tasks (such as cluster identification) on realistically larger graphs than those in our studies is an important next step - but it is beyond the scope of this initial study which focuses on precise graph readability and path/link following. In the next chapter, we investigate if torus drawing works well for graphs larger than those in our studies. Based on the improved graph aesthetics such as crossing reduction, such torus topology with link wrapping has a potential to better untangle a network than non-wrapped layouts. We expect that the reduced clutter torus diagrams may work well for tasks particularly important to large networks, and those with complex structures that can benefit from a relaxation of the structure and less link crossings - an important network exploration task - visual cluster identification, which is explored in the next chapter.

Another limitation is that we only tested one static force-directed layout algorithm. There are many different layout algorithms for 2D node-link visualisations, such as orthogonal link layout by Batini et al. [BNT86; Kie+15], layered layout by Sugiyama et al. [STT81], network layout with data aggregation by Yoghourdjian et al. [Yog+18], as described in section 2.1. We leave detailed discussion to section 9.2.

Second, the automatic layout algorithm based on ALL-PAIRS stress-minimisation approach is known to easily become stuck in local minima of the stress function (as described in subsection 5.1.1), corresponding to a poor choice of link wrapping across the torus surface. Therefore, the layouts tested in both studies involved human-guidance of the algorithm. The interactive nature of the algorithm means that a user can guide it to a quite reasonable layout, certainly layouts that were good enough for our study. However, such an interaction would not scale to larger, real-world networks. In the next chapter, we explore a different algorithm that is more robust and completely autonomous. We further evaluate the new algorithm with realistically larger networks that addresses the limitations stated in this section.

5.8 Conclusion and Future Work

We have investigated a technique using a force-directed approach for mapping arbitrary network structures onto a two-dimensional toroidal topology with links wrapping around the boundaries for relaxing node positions to reduce visual clutter.

Our studies indicate that torus-based network layout could be a practical technique (not worse than the standard force-directed graph visualisation technique tested) for link-following tasks (identify neighbours/find the shortest path) and feature identification tasks (estimate the number of nodes or links), but that either redundant context or interactive panning are necessary. The graphs tested were automatically generated using algorithms designed to simulate naturally occurring graphs, but we would like to further evaluate the torus representations in a real-world application and see if it is usable by domain experts. This chapter addressed **RG4** (subsection 1.2.4). The study results confirmed the feasibility of two wrapping approaches: tile-display and interactive wrapping described in section 3.4 that aid understanding of torus wrapping.

Technically speaking, our ALL-PAIRS stress-minimising torus layout method is the first method we are aware of that is able to layout all graphs (not just graphs limited to a particular genus) on a torus topology. While we can force it to converge we cannot make strong guarantees that it converges to a local optimum. However, we are interested to see if the combinatorial techniques for layout of restricted classes of graphs on the torus, can be adapted to help us find good starting conditions for our stress-minimising torus layout in order to create a robust and completely autonomous high-quality torus layout.

Interactive Torus Wrapping for Network Visualisations - Larger Networks

6

> *It is not how much empty space there is, but rather how it is used. It is not how much information there is, but rather how effectively it is arranged.*
>
> — **Edward Tufte**
> (American statistician and emeritus professor of political science, statistics, and computer science at Yale University.)

Building upon the work presented in chapter 5, this chapter addresses **RG4** (subsection 1.2.4). We introduce and evaluate two new algorithms for improving toroidal wrapped layouts. We address limitations of our previous investigation of toroidal network layouts (described in section 5.7) by considering larger networks, and investigating whether torus-based layout might also better display high-level network structure like clusters.

First, we present a new algorithm for computing toroidal node-link diagram layout that is completely autonomous and considerably more robust than the previous ALL-PAIRS method described in section 5.1, and consistently producing high-quality layouts (section 6.3). Like ALL-PAIRS stress minimising torus layout method, this algorithm is a general-purpose toroidal layout algorithm based on a variant of a force-directed placement, and therefore it is capable of laying out arbitrary networks. We extend the standard non-wrapped pairwise gradient descent algorithm of Zheng et al. [ZPG18] to handle the more complex case of layout on a torus.

Second, to better discriminate clusters, we explore a completely novel issue identified through our testing (section 6.4) as well as our previous study results (section 5.7) which indicate that link wrapping across the plane's edges impedes network understanding tasks when neither tile-display nor interactive panning (section 5.7) is

|(a) Unwrapped|(b) Wrapped|(c) Best pan|
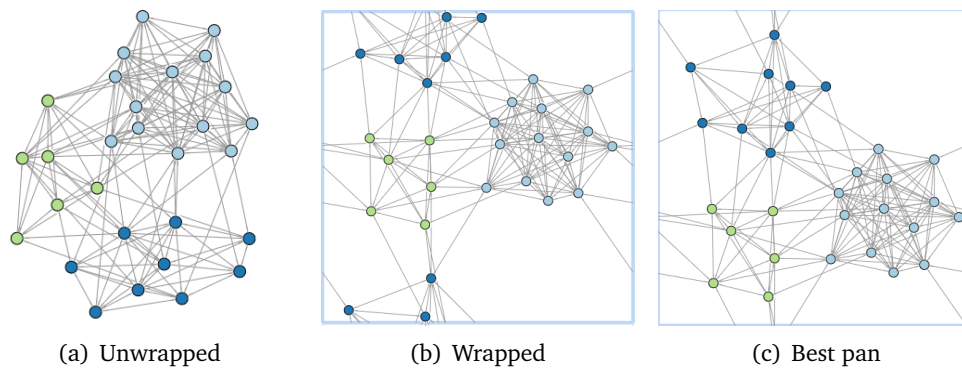
Fig. 6.1.: Example of unwrapped and wrapped networks; Colours are used to illustrate already known clustering information: (a) standard force-directed layout; (b) torus layout wraps the links around the boundaries vertically top-to-bottom, or horizontally left-to-right using our toroidal layout algorithm (section 6.1); (c) wrapping links are minimised by our automatic pan algorithm (section 6.2)

introduced. We present a new algorithm for computing how best to "cut" a toroidal network layout to the surface of a torus, i.e., to automatically pan the viewport to reduce the number of links wrapped. To the best of our knowledge this has not been previously considered. As shown in Figure 6.1(c) and Figure 6.2(d), toroidal layouts with automatic panning better reveal the structure of the networks such as clusters and makes a considerable difference to the quality of the final layout.

Third, we compare ALL-PAIRS and our new automatic toroidal layout method with non-wrapped flat layout algorithms, using a large corpus of 200 networks with community structures (defined in section 6.3). We explore whether our improved torus-based layout algorithm is able to find node positions affording better graph aesthetics.

Fourth, we conduct a new study (section 6.4) with 32 participants, evaluating (a) a new and optimised layout algorithm with (b) automatic wrapping, using (c) larger networks (≤ 134 nodes, ≤ 2590 links), and (d) focus on cluster discrimination tasks. We show that torus layout significantly outperformed non-torus for cluster identification tasks in terms of improvement in error by 62.7% and time by 32.3%.

6.1 Improved Algorithm for Torus Layouts

We introduce a toroidal layout algorithm that solves a key limitation of the ALL-PAIRS gradient descent algorithm proposed in section 5.1 which could become stuck in local minima (as shown in Figure 6.5(a)) and thus which often requires manual

| (a) No pan | (b) Best horizontal pan | (c) Best vertical pan | (d) Best pan in both horizontal and vertical directions |
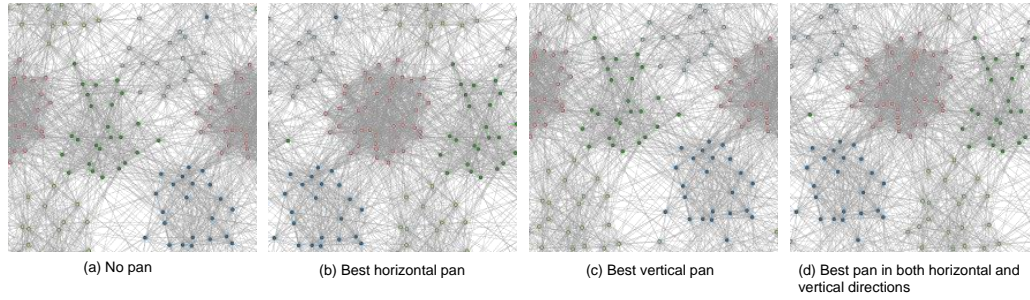
Fig. 6.2.: Example of automatic panning of toroidal layouts of a network with 126 nodes and 2496 links from SMALL+HARD (described in subsection 6.3.1 and Figure 6.7): (a) Original toroidal layout without auto pan requires a user to pan to navigate the network; (b-c) Either horizontal or vertical pan reduces link wrappings across the edges of the display; (d) Best pan in both horizontal and vertical directions minimises the number of link wrappings and better reveals 5 main clusters.

intervention to guide the algorithm to find a reasonable layout. We use this algorithm to find layouts that consistently yield better graph aesthetics: fewer crossings, less stress, greater incidence angle, and greater cluster distance (section 6.3).

Following ALL-PAIRS gradient descent (section 5.1), our layout algorithm finds node positions of each pair of nodes in a 3×3 repeated tiling. We develop an iterative algorithm which seeks to minimise *stress* of the layout across the surface of the torus. However, while in ALL-PAIRS gradient descent the layout algorithm moves all pairs of vertices at each iteration and also needs a good initialisation to find a suitable torus wrapping, our new approach randomly moves a single pair of vertices at a time. In our empirical testing (see subsection 6.3.2), we find that this stochastic approach avoids the algorithm getting stuck prematurely in local minima of the stress function and leads autonomously to a high-quality torus layout without the need of a good initial state.

We name this new approach PAIRWISE, given that it minimises the stress function by moving a single pair of nodes at a time using gradient descent, as opposed to our previous layout approach, ALL-PAIRS. The idea to randomly select and move a single pair of nodes at a time is inspired by an approach suggested for general (non-toroidal) stress-minimising graph layout by Zheng et al. [ZPG18].

The key component to adapting a gradient-descent layout approach (such as stress minimisation) to a 2D torus layout, is to consider the nine different possible choices for wrapping each link, as shown in Figure 5.1. We find that the PAIRWISE approach works particularly well for toroidal layout because at each iteration, as well as optimising stress for a single link, we can choose the optimal link wrapping configuration for that link. In the ALL-PAIRS gradient descent approach using ALL-PAIRS,

selecting the wrapping configuration for all links at once was the source of significant instability.

6.1.1 Pairwise Gradient Descent

Following ALL-PAIRS, our toroidal layout approach is based on minimising the stress function defined in Equation 5.1). We consider a non-wrapped node-link layout, named NOTORUS, and a wrapped layout on a torus topology, named TORUS. For each pair of nodes, we compute the gradient information across the nine possible ways to consider their adjacency.

The constant L is selected proportionally to *cell size* and the graph diameter, i.e., the longest of the shortest-paths of G, as defined in Algorithm algorithm 1. Note that it is possible to change the choice of L to produce a layout that has a desired level of torus wrapping. Unlike ALL-PAIRS where the choice of L was not clearly shown and relied on a good initialisation, the method presented in this section produces deterministic layouts.

A gradient-descent approach to reducing the stress function uses the gradient information for this function in a given graph configuration to choose descent vectors, or directions by which to move the nodes to reduce the overall stress function, as per Dwyer et al. [DMW08]. For TORUS, for a given pair of nodes we choose the adjacency across cells which contributes to the greatest reduction in stress as the descent vector by which the nodes will be moved. In ALL-PAIRS, such descent vectors were computed simultaneously across all pairs of nodes, before moving all nodes according to the computed vectors. In this paper, however, we move just a pair of nodes at a time and follow an annealing schedule to enforce convergence.

A summary of our annealing schedule is shown in Equation 6.1, where $\eta(t)$ is a time-dependent scale factor applied to descent vectors before moving nodes accordingly.

$$\eta(t) = \begin{cases} min\left(1, \frac{D^2_{max}}{D^2_{uv}}e^{-\lambda t}\right) & \text{for} \quad t \leq \tau \\ min\left(1, \frac{D^2_{min}}{D^2_{uv}}\frac{1}{1+\lambda t}\right) & \text{for} \quad \tau < t \leq \tau_{max} \end{cases} \tag{6.1}$$

We follow Zheng et al. [ZPG18] in choosing an exponential decay schedule for a fixed number of iterations τ. Starting from the next iteration after τ, the algorithm uses a $\frac{1}{t}$ schedule to converge to a stable configuration. The annealing schedule begins with a maximum step size, 1 at first iteration $t = 0$. This avoids local minima, as reported by Zheng et al. [ZPG18]. λ is a decay constant determined by a given

parameter ϵ such that the schedule's step size at iteration τ is constrained from above by ϵ, i.e., $D_{max}^2 e^{-\lambda\tau} = D_{min}^2 \epsilon$.

Data: *graph $G = (V, E)$, wrapping $W = 3 \times 3$ tiles*
Result: *Graph embeddings on a 2D plane*
$L \leftarrow \frac{Cell\ Size}{min(Graph\ Diameter\ ,\ 2)+\ 1}$;
$X \leftarrow |V| \times |V| \times 2$ node position matrix;
$D \leftarrow ShortestPathsMatrix(G)$;
for *η in each annealing schedule in Equation* 6.1 **do**
    for *each $u, v \in V$ in random order* **do**
        if *wrapped* **then**
            $X'_{uv} \leftarrow$ set of 9 possible vectors from u to v across the cells of W;
            $d_{uv} \leftarrow$ euclidean lengths of each X'_{uv};
            $R \leftarrow (min_{w \in W} \frac{(L \times D_{uv} - (d_{uvw})^2}{2}) \frac{\overrightarrow{X'_{uvw}}}{d_{uvw}}$;
        else
            $R \leftarrow \frac{(L \times D_{uv} - d_{uv})^2}{2} \frac{\overrightarrow{X_{uv}}}{d_{uv}}$;
        end
        $X_u \leftarrow X_u - \eta R$;
        $X_v \leftarrow X_v + \eta R$;
        if *wrapped* **then** *translate X_u, X_v back to centre cell*;
    end
end

Algorithm 1: PAIRWISE gradient descent layout algorithm, which provides a TORUS layout when *wrapped* is true, or a NOTORUS layout when *wrapped* is false.

The parameter that determines how well a toroidal layout spreads out clusters of a network is (1) τ which controls when to switch from exponential schedule to the $\frac{1}{t}$ schedule (Equation 6.1), (2) ϵ which determines the constraint of $\eta(t)$. We experimented with a variety of parameter τ and ϵ for the step size schedule $\eta(t)$ as shown in Figure 6.3 and Figure 6.4. For τ, we fixed $\epsilon = 0.1$ while varying τ in a range of 20 and 120. We used the graphs from SMALL-HARD graph corpus described in subsection 6.3.1 and Figure 6.7. We find increasing τ led to better separation of clusters (defined more formally in subsection 6.3.4) and less stress, as shown in Figure 6.3(a, c). For ϵ, we then fixed $\tau = 80$ while varying ϵ from 0.05 to 0.3. The results indicate that τ greater than 0.2 is a poor choice and thus it has worse cluster separation and stress, as shown in Figure 6.3(b, d). We therefore set $\tau = 80$ and $\epsilon = 0.1$ for the exponential schedule. For $\frac{1}{t}$ schedule we set $\epsilon = 0.001$. This gives a smaller step size and thus the layout algorithm always converges.

The stopping criterion is set to a maximum pairwise movement $\delta < 0.03$ or a maximum of $\tau_{max} = 200$ iterations , whichever comes first. Algorithm 1 gives
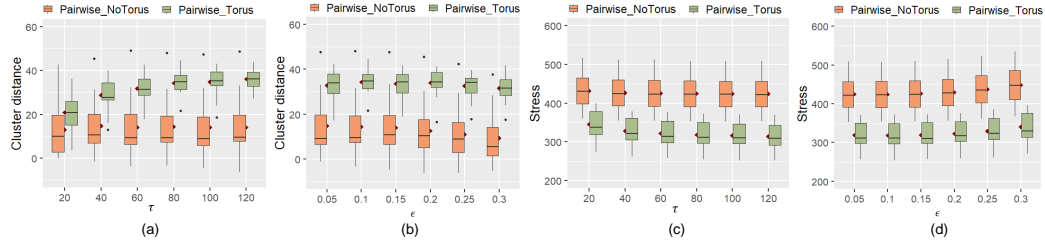
Fig. 6.3.: The cluster distance and stress of networks from SMALL+HARD (subsection 6.3.1) with 20 runs when varying parameters τ and ϵ on Equation 6.1. A larger value of τ gave greater cluster distance (a) and less stress (c) with $\epsilon = 0.1$. There was not much improvement for either $\tau > 80$ (a,c) or $\epsilon > 0.1$ (b,d). Therefore we chose $\tau = 80$ and $\epsilon = 0.1$ for our PAIRWISE layout algorithm.
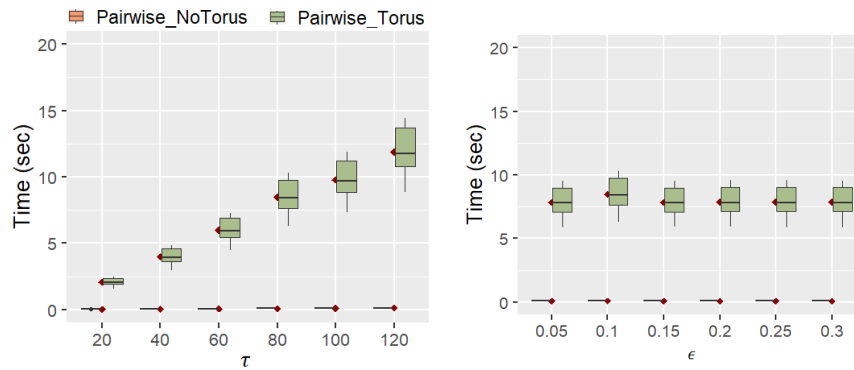

Fig. 6.4.: The running time of layout algorithms from SMALL+HARD (subsection 6.3.1) with 20 runs when varying parameters τ and ϵ on Equation 6.1.

pseudocode for this process [1]. Figure 6.5 shows layout examples from Large+Hard graphs of All-Pairs and Pairwise for NoTorus and Torus at convergence. Similarly, examples of Large+Easy graphs using these layout algorithms are shown in Figure 6.6.

6.1.2 Computational Complexity

The time complexity of each iteration of Pairwise and All-Pairs stress minimisation is $O(|V|^2)$ where V is a set of vertices. There is a constant factor W due to torus adjacencies which does make our reported run-times uniformly slower than non-torus layout. The space complexity of all the methods are the same, i.e., $O(|V|^2)$ Pairwise-Torus's asymptotic computation time complexity is the same as traditional force-directed layout so we would expect similar scalability (subject to the constant factor W). Further improvement may be possible using multilevel spatial decomposition methods, e.g. by Quigley and Eades [QE00] and Walshaw [Wal00].

6.2 Automatic Panning Algorithm

As investigated in section 5.5, additional context and interactive panning have been shown to improve understanding of wrapped visualisations. The qualitative feedback in that study revealed that repetition of networks in adjacent cells is not practical for displaying large graphs. Rather, interactive panning was found to be highly-beneficial in helping study participants to pan, such that links of interest are not wrapped. However, we can greatly support this process by automatically panning to minimise the number of links that are split across viewport boundaries.

To perform automatic panning of a given layout (e.g. as determined by the algorithm given in subsection 6.1.1, we take a sweepline approach by Shamos and Hoey [SH76] to search for pan positions which lead to the least number and severity of split links, horizontally and vertically. A given layout provides a fixed set of node positions relative to a viewport. A "pan" of a Torus layout, involves translating all node positions uniformly, except where a node would move outside the viewport, it is wrapped to the other side of the viewport, top-to-bottom or left-to-right. For a given layout of a graph with $|V|$ nodes, the node positions left-to-right define an ordering over the nodes, and the positions top-to-bottom provide a second ordering. The set

---

[1]The detailed pseudocode to our method is available from `https://github.com/Kun-Ting/its-a-wrap`

of links that are wrapped across viewport boundaries $E_{wr} \subseteq E$ is constant under translation of the nodes, until the translation is sufficiently large that a node must be wrapped around. Thus, in each axis (horizontal and vertical) there are precisely $|V|$ distinct translations that must be considered in order to examine all sets of possible wrapped links for a given layout. We can examine each of these sets to determine which induces the lowest *wrapping cost*, defined as Equation 6.2, where d_{uv} is the Euclidean distance between u and v connected by a link e:

$$wrapcost(E_{wr}) = \sum_{e \in E_{wr}} \frac{1}{d_{uv}} \tag{6.2}$$

Algorithm 2 details the steps of this procedure. The runtime complexity of Automatic Panning is $O(|V| \log |V| + |E|)$ due to the need to sort node positions and examine all links.

> **Data:** Node position vector X with position $(x, y) \in X$ for each node, for a torus layout of a graph G from Algorithm algorithm 1. Figure 6.2(a) shows an example.
> **Result:** Torus layout with minimum link wrappings on the boundary and less disconnected clusters centred within the viewport
> **Horizontal Sweep (result shown in Figure 6.2(b)):**
> (1) Sort nodes by increasing $(x, \_) \in X$ and initialise sweep-line positions S with the mid-points s_i of all adjacent $x_i, x_{(i+1)}$;
> (2) For each sweep line position $s_i \in S$, maintain a set of open links. Assuming we sweep left-to-right, at each s_i we add to E_{wr} the links outgoing from the right side of node i and remove any links incoming to the left-side of i. If $wrapcost(E_{wr})$ (Equation 6.2) of open links is smaller than $minCost$, take this as the new $minCost$ and set $minX$ to the current sweep line position s_i;
> **Vertical Sweep (result shown in Figure 6.2(c)):**
> (3) Repeat step 1 and 2 for all y-positions $(\_, y) \in X$. Therefore, $minY$ is the sweep line position with minimum cost.
> **Apply optimal pan and centre within viewport**
> **(result shown in Figure 6.2(d)):**
> (4) Translate all node positions based on $minX, minY$.
> (5) Centre the layout such that the centre x-position of the left-most and the right-most nodes is at the centre of the cell; and similar to centre vertically.
> **Algorithm 2:** Auto Panning Procedure

6.3 Algorithm Evaluation

In this section, we compare two layout conditions: torus (TORUS) and traditional 2D planes (NOTORUS) for both our proposed algorithm PAIRWISE (Algorithm 1) and the ALL-PAIRS algorithm (section 5.1) against a large corpus of 200 graphs. We

show that PAIRWISE has convergence and run-time performance benefits over its predecessor ALL-PAIRS. We assess layout quality using established graph aesthetics measures and a novel cluster readability metric, *cluster distance*, measuring visual distance between boundaries of clusters in a given layout, finding that PAIRWISE toroidal layout algorithm outperforms either PAIRWISE non-torus or ALL-PAIRS layout algorithms.

In order to evaluate how well a layout method separates nodes and clusters, we look at graphs with community structures [Mis+07; SSK14]. We control for *modularity*, a metric for graph theoretic community structure, defined by Newman [New06]. A graph with high modularity indicates that the cluster structure is more distinct, as there are more links within each cluster than between clusters [New06; For10].

We compared means of runtime, stress, crossings, incidence angle, and cluster distance of four different layout methods, i.e., PAIRWISE-TORUS, PAIRWISE-NOTORUS, ALL-PAIRS-TORUS and ALL-PAIRS-NOTORUS, using Friedman's non-parametric test and Nemenyi's post-hoc pairwise comparison, as they did *not* follow normal distribution. We report significant differences under 95% confidence.

6.3.1 Graph Corpus

The graphs in our sample corpus were generated using algorithms designed to simulate real-world community structures in graphs [BGW03; For10], using generators from NetworkX [Pac19]. We generated 200 graphs, grouped by two variables: graph modularity [New06] (5 levels from low to high: **0.25, 0.3, 0.35, 0.4, 0.45**) and graph size (2 levels: SMALL: 68-80 nodes, 710-925 links, 3-8 clusters, and LARGE: 126-134 nodes, 2310-2590 links, 3-8 clusters). This gives us 10 classes. For all classes, the graph density is calculated by the ratio of the number links of the graph to the maximum number of possible links ($\frac{2 \times |E|}{|V| \times (|V|-1)}$). This density is fixed at a range of 0.3±0.01.

We use a standard *Random Partition Network* model by Fortunato [For10] and *Gaussian Random Partition* model [BGW03] to generate our graph corpus. This gives us graphs with clustering information based on the desired range of modularity, density and size. We exclude graphs whose minimum modularity of an individual cluster is ≤ 0.23, which we found was the minimum to provide visible community structure [For10].
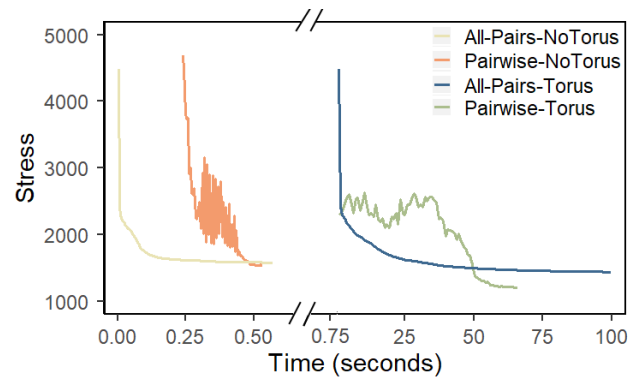
(a) Stress Against Time
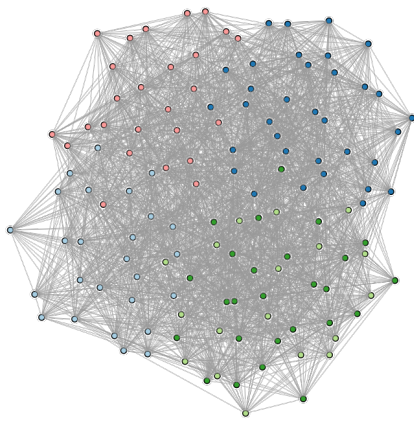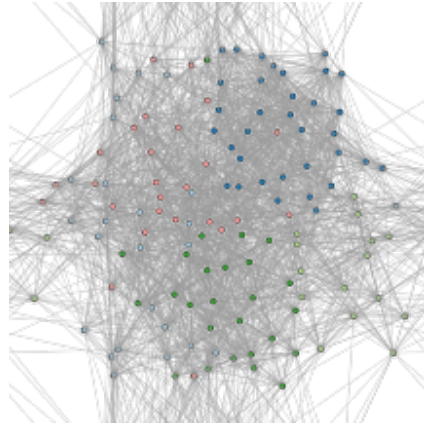
(b) ALL-PAIRS NOTORUS
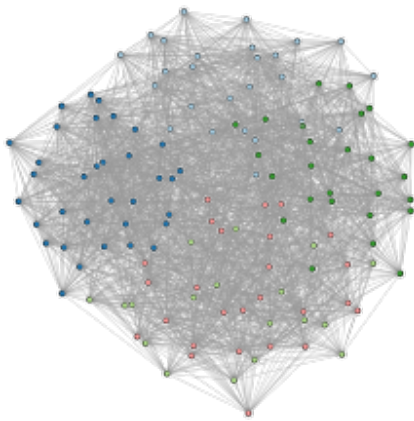
(c) ALL-PAIRS TORUS

(d) PAIRWISE NOTORUS
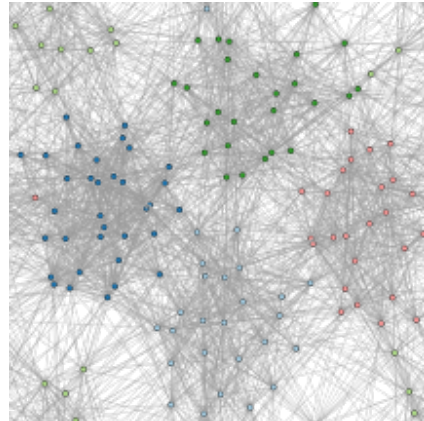
(e) PAIRWISE TORUS

Fig. 6.5.: (a) Mean stress over time of 20 runs of torus and standard node-link for both PAIRWISE and ALL-PAIRS algorithms for a network with 130 nodes, 2504 links from LARGE+HARD (subsection 6.3.1). (b-e) show the network layouts at convergence. PAIRWISE-TORUS reached a lower stress level, was faster to converge, and better revealed network clusters than ALL-PAIRS-TORUS.
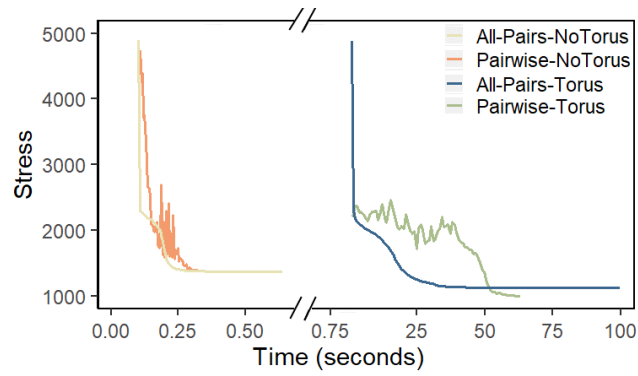
(a) Stress Against Time
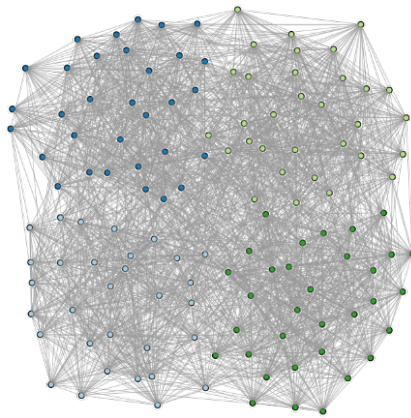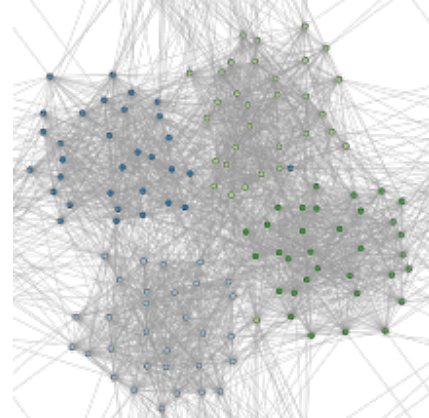
(b) ALL-PAIRS NOTORUS
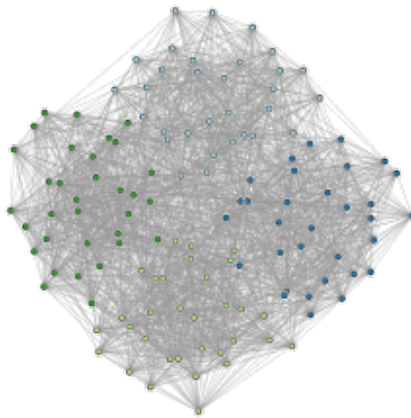
(c) ALL-PAIRS TORUS

(d) PAIRWISE NOTORUS
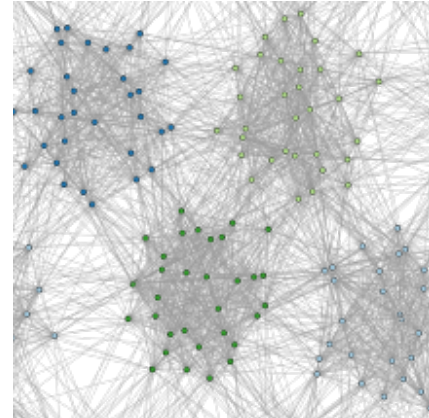
(e) PAIRWISE TORUS

Fig. 6.6.: (a) Mean stress over time of 20 runs of torus and standard node-link for both PAIRWISE and ALL-PAIRS algorithms for a network with 130 nodes, 2504 links from LARGE+EASY (subsection 6.3.1). (b-e) show the network layouts at convergence. PAIRWISE-TORUS reached a lower stress level, was faster to converge, and better revealed network clusters than ALL-PAIRS-TORUS.

| Graph Set	Number Graphs	Nodes	Links	Number Clusters	Graph Density	Topology	Layout Method	Cluster Distance	Stress	Time (sec)	Crossings	Incidence Angle
Clustered Small Modularity=0.4 (SMALL-EASY)	20	72.1	785.1	4.6	0.3	No Torus	All Pairs	37.7	399	0.16	24702	0.957
						No Torus	Pairwise	42.9	393.9	0.08	24376	0.958
						Torus	All Pairs	-10.7	386	22.4	20635	0.951
						Torus	Pairwise	56.1	274.3	8.5	14075	0.946
Clustered Large Modularity=0.4 (LARGE-EASY)	20	129.1	2470.3	4.4	0.3	No Torus	All Pairs	29	1412.2	0.5	277356	0.978
						No Torus	Pairwise	33.7	1408.9	0.3	276040	0.979
						Torus	All Pairs	-24.4	1328.5	85.3	209443	0.975
						Torus	Pairwise	47	1047.1	64.5	158202	0.975
Clustered Small Modularity=0.3 (SMALL-HARD)	20	71.5	774.6	4.7	0.3	No Torus	All Pairs	11.1	427.9	0.16	29555	0.959
						No Torus	Pairwise	14.3	424.6	0.08	29343	0.959
						Torus	All Pairs	-43	422.9	23.2	23984	0.952
						Torus	Pairwise	34.3	318.1	8.4	14751	0.944
Clustered Large Modularity=0.3 (LARGE-HARD)	20	129.1	2535.4	5.3	0.3	No Torus	All Pairs	3.1	1545.4	0.5	363685	0.979
						No Torus	Pairwise	3.3	1543.4	0.3	361353	0.98
						Torus	All Pairs	-67.4	1529.1	86.4	268336	0.975
						Torus	Pairwise	28.6	1220.7	65.6	179452	0.973

Fig. 6.7.: Average properties of 80 random partition networks and graph aesthetics results of 20 runs of each network rendered using TORUS, NOTORUS for both PAIRWISE and ALL-PAIRS algorithms when varying modularity and size.

Fig. 6.8.: Graphics showing statistically significant results of running time of SMALL graphs: 68-80 nodes, 710-925 links, and LARGE graphs: 126-134 nodes, 2310-2590 links of 200 networks laid out using TORUS and NOTORUS for both PAIRWISE and ALL-PAIRS algorithms when varying graph modularity between 0.25 and 0.45 for SMALL and LARGE networks.

Fig. 6.9.: Box plots of running time of 200 networks laid out using TORUS and NOTORUS for both PAIRWISE and ALL-PAIRS algorithms when varying graph modularity between 0.25 and 0.45 for SMALL and LARGE networks.
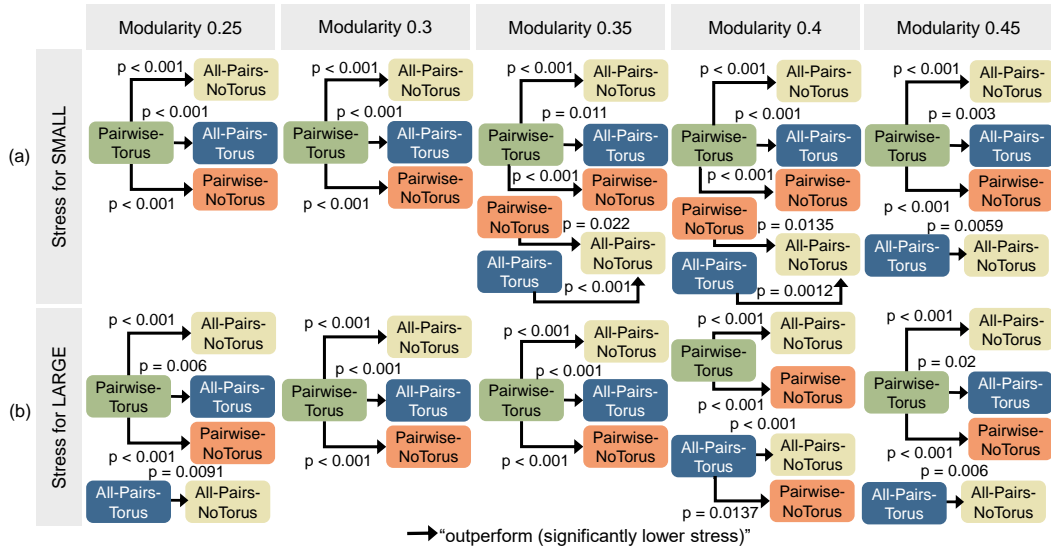
Fig. 6.10.: Graphics showing statistically significant results of stress of SMALL graphs: 68-80 nodes, 710-925 links, and LARGE graphs: 126-134 nodes, 2310-2590 links of 200 networks laid out using TORUS and NOTORUS for both PAIRWISE and ALL-PAIRS algorithms when varying graph modularity between 0.25 and 0.45 for SMALL and LARGE networks.
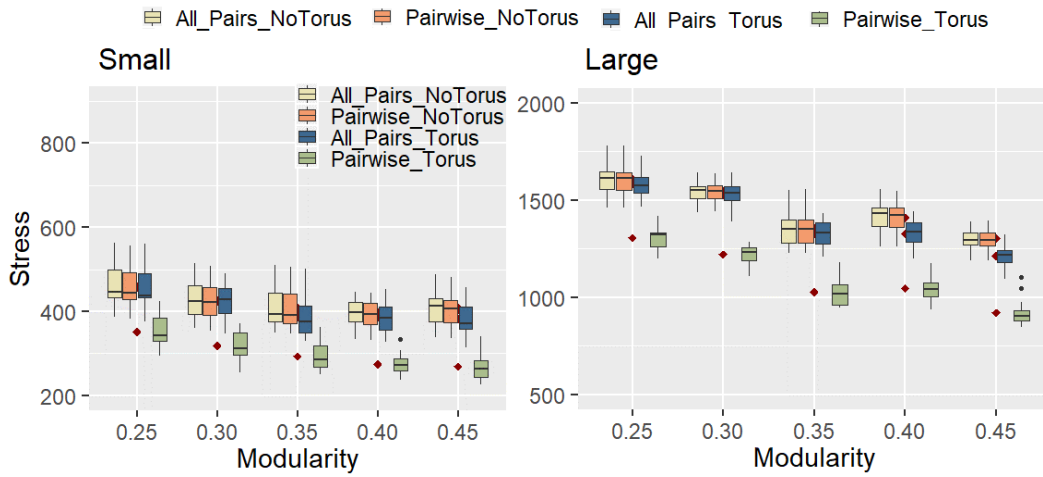
Fig. 6.11.: Box plots of stress of 200 networks laid out using TORUS and NOTORUS for both PAIRWISE and ALL-PAIRS algorithms when varying graph modularity between 0.25 and 0.45 for SMALL and LARGE networks.
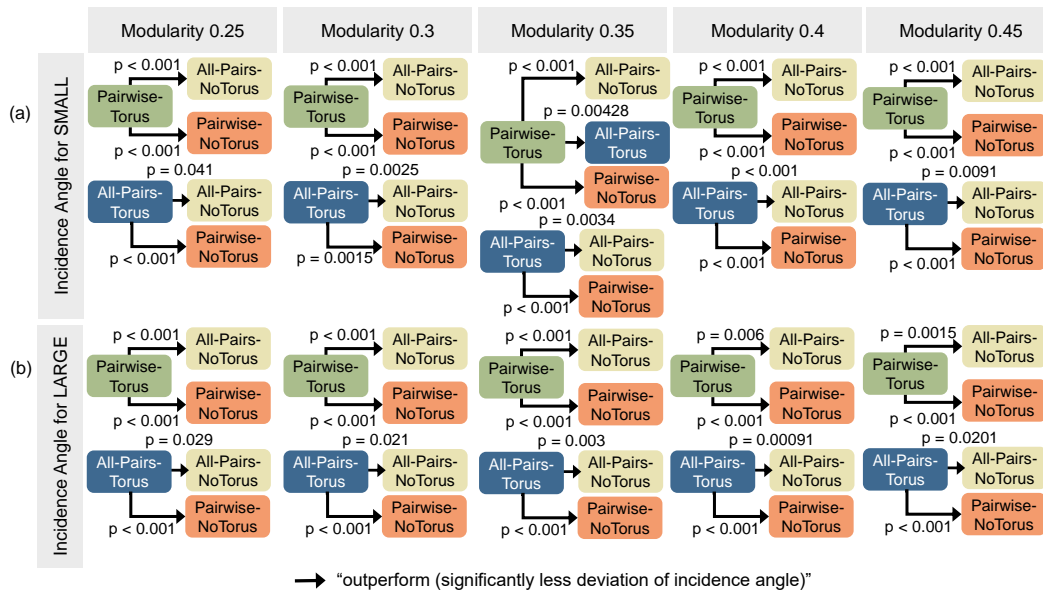
Fig. 6.12.: Graphics showing statistically significant results of incidence angle of SMALL graphs: 68-80 nodes, 710-925 links, and LARGE graphs: 126-134 nodes, 2310-2590 links of 200 networks laid out using TORUS and NOTORUS for both PAIRWISE and ALL-PAIRS algorithms when varying graph modularity between 0.25 and 0.45 for SMALL and LARGE networks.
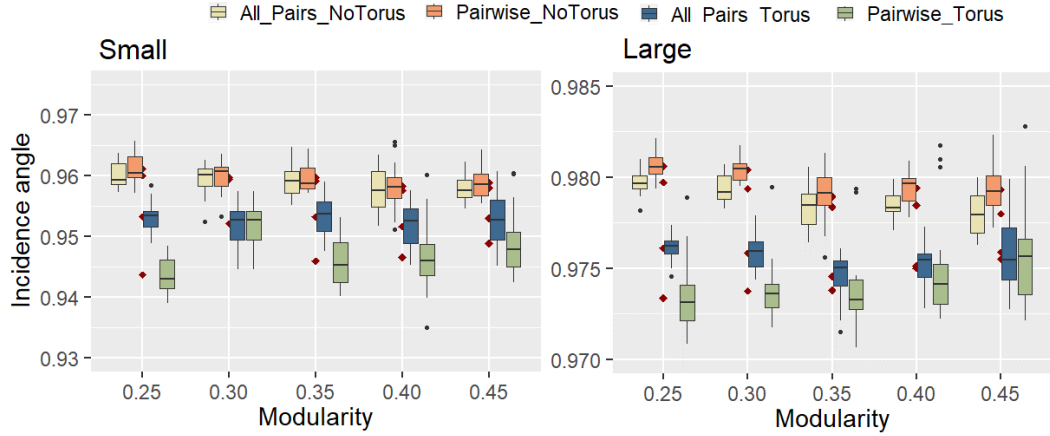
Fig. 6.13.: Box plots of incidence angle of 200 networks laid out using Torus and NoTorus for both Pairwise and All-Pairs algorithms when varying graph modularity between 0.25 and 0.45 for Small and Large networks.
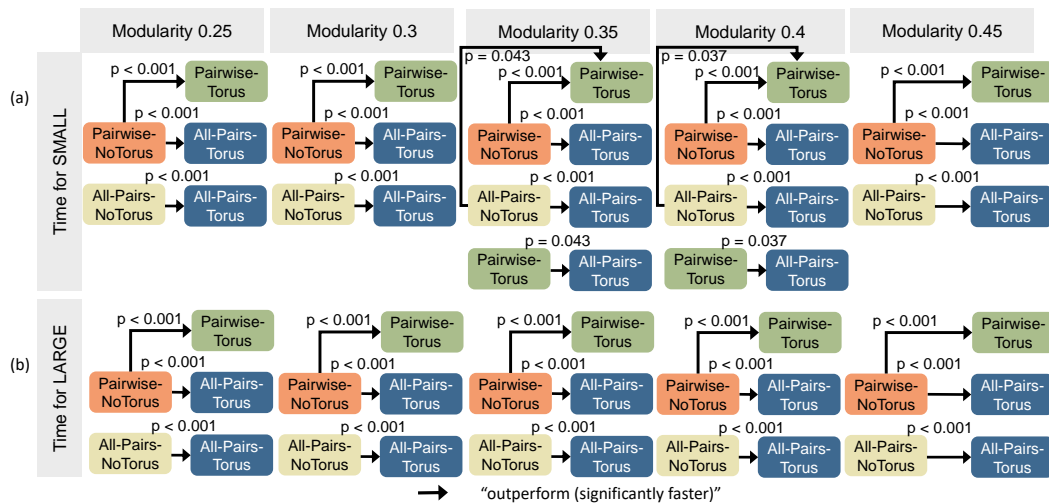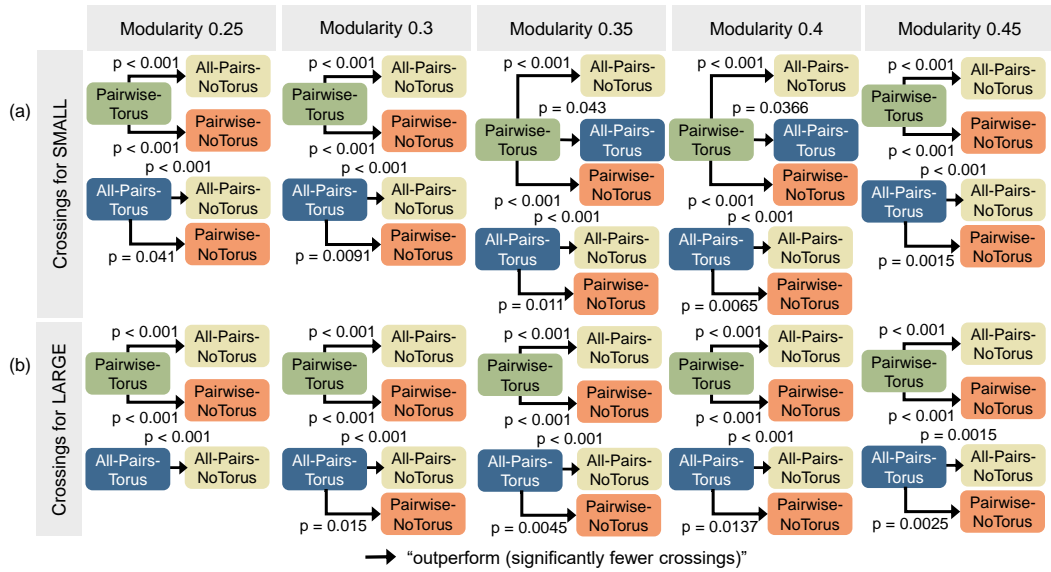

Fig. 6.14.: Graphics showing statistically significant results of link crossings of Small graphs: 68-80 nodes, 710-925 links, and Large graphs: 126-134 nodes, 2310-2590 links of 200 networks laid out using Torus and NoTorus for both Pairwise and All-Pairs algorithms when varying graph modularity between 0.25 and 0.45 for Small and Large networks.
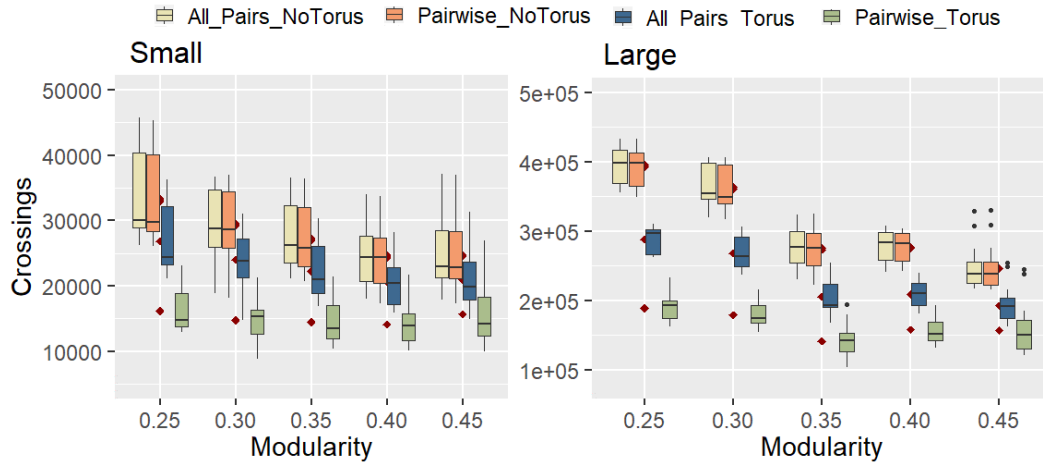
Fig. 6.15.: Box plots of link crossings of 200 networks laid out using Torus and NoTorus for both Pairwise and All-Pairs algorithms when varying graph modularity between 0.25 and 0.45 for Small and Large networks.
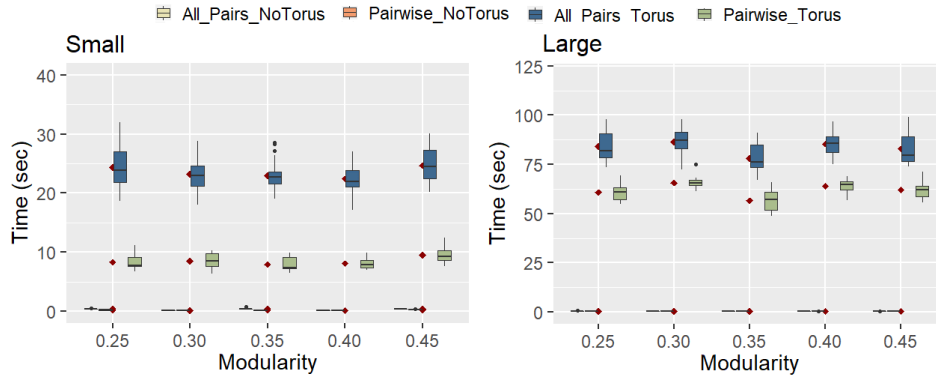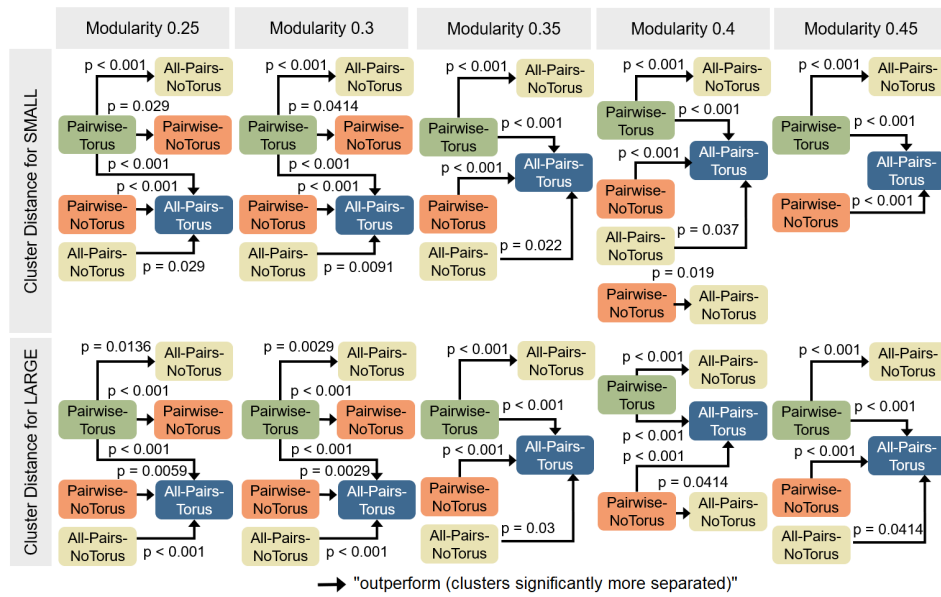

Fig. 6.16.: Statistical results of cluster distance of 200 networks laid out using Torus and NoTorus for both Pairwise and All-Pairs algorithms when varying graph modularity between 0.25 and 0.45 for Small and Large networks.
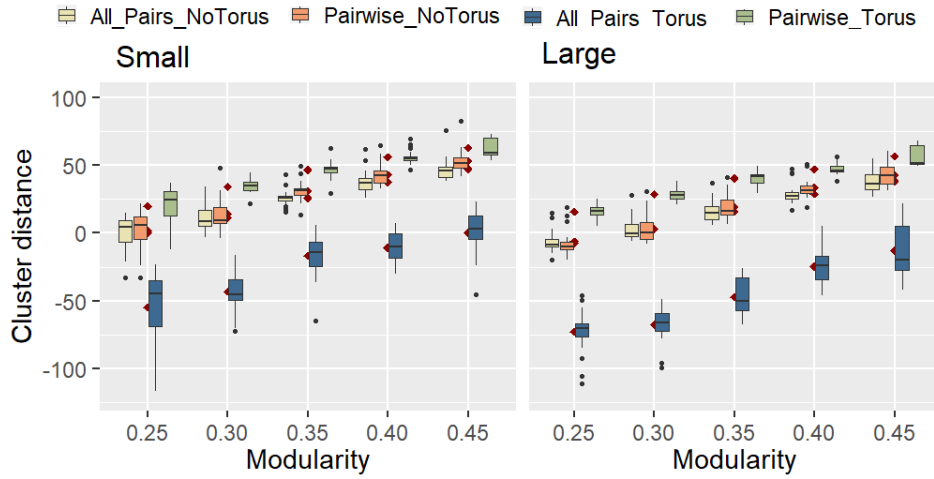
Fig. 6.17.: Box plots of cluster distance of 200 networks laid out using TORUS and NOTORUS for both PAIRWISE and ALL-PAIRS algorithms when varying graph modularity between 0.25 and 0.45 for SMALL and LARGE networks.

6.3.2 Runtime comparison

We compare runtime performance of the PAIRWISE algorithm with ALL-PAIRS. We implemented the PAIRWISE algorithm in JavaScript and D3 [Bos22], and implemented ALL-PAIRS as per section 5.1 in *WebCola*[15]. We ran both algorithms with networks from our graph corpus as described in subsection 6.3.1. For each graph, we generated network layouts with 20 random initial node positions controlled by seed within a 1×1 square at the centre for each graph. The configuration of unit link length, stopping criteria, and maximum number of iterations are the same for each technique and described in section 6.1. We record the time of each technique using Google Chrome browser (version 80), running on an Intel i7-7800X (3.5GHz) CPU and 32GB of RAM. Overall, Figure 6.7 indicates that PAIRWISE-TORUSis significantly faster (p=0.037) to converge than ALL-PAIRS-TORUS by 62% (differences between means) for SMALL+EASY. The significant results are also found for SMALL graphs with modularity at 0.35 and 0.4 as shown in Figure 6.8, which shows graphics of statistically significant results of running time using our graph corpus (subsection 6.3.1). The corresponding results in box plots are shown in Figure 6.9 .

Figure 6.5(a) shows mean stress over time of an example graph for LARGE+HARD with 20 runs. An example graph for LARGE+EASY is illustrated in Figure 6.6. The result shows that PAIRWISE avoids the algorithm getting stuck in local minima of the stress function as opposed to ALL-PAIRS. Furthermore, PAIRWISE-TORUS reaches lower stress levels than ALL-PAIRS-TORUS at convergence. While the convergence time is affected by stopping conditions, in our evaluation, we used the same con-

vergence threshold and maximum number of iterations for all methods. However, step size attenuation also affects convergence time. It is computed differently in ALL-PAIRS torus, which is based on gradient contribution of all pairs nodes as in Dwyer et al. [DMW08] and PAIRWISE, which we chose 80 iterations as the threshold τ (Equation 6.1) between exponential and convergence schedule based on experimental results. Experimentally, we found a larger τ led to a longer time to converge, as seen in Figure 6.4(a), but it did not give much improvement in terms of cluster separation, as shown in Figure 6.3. Figure 6.5(b-e) shows an example for a LARGE+HARD graph. It shows that PAIRWISE-TORUS generates higher-quality layouts than PAIRWISE-NOTORUS, ALL-PAIRS-NOTORUS and ALL-PAIRS-TORUS methods at convergence. A similar result can be observed for the example from the LARGE+EASY graph, shown in Figure 6.6(b-e).

6.3.3 Quality comparison using established graph aesthetics metrics

Following subsection 5.2.1, we first used a set of layout aesthetic quality metrics to compare NOTORUS and TORUS using PAIRWISE and ALL-PAIRS. However, we are more interested in whether the additional spreading afforded by torus-based layout is better at showing clusters and so revealing the network structure. This was not considered in the previous chapter.

The metrics— *stress*, *minimum incidence angle,* and *number of line crossings*—have been shown important for user performance in readability tasks in the past by Purchase [Pur02] and Ware [War+02].

Stress—measures how well the layout captures the structure in the underlying network [Dev+19]. Graphs with lower stress have been found to be more preferred by users [Dwy+09]. We calculate stress as per Equation 5.1. Graphics of statistically significant results are summarised in Figure 6.10 with corresponding box plots in Figure 6.11. As shown in Figure 6.5 and Figure 6.7, PAIRWISE-TORUS achieves significantly lower stress ($p < 0.001$) than PAIRWISE-NOTORUS, ALL-PAIRS-NOTORUS for SMALL+EASY, LARGE+EASY, SMALL+HARD, and LARGE+HARD. PAIRWISE-TORUS achieves significantly lower stress ($p < 0.001$) than ALL-PAIRS-TORUS for SMALL+EASY, SMALL+HARD, and LARGE+HARD. For unwrapped layout, we found PAIRWISE-NOTORUS achieves significantly lower stress ($p = 0.0135$) than ALL-PAIRS-NOTORUS for SMALL+EASY.

Incidence Angle—Maximising the minimum angle of incidence between links entering a node gives better readability of network connectivity. Following the definition in subsection 5.2.1, our metric for incidence angle measures deviation of minimum-incidence angle for each node from the ideal maximum for that node's degree. Smaller deviation from this ideal is better. Graphics of statistically significant results are shown in Figure 6.12, with corresponding box plots in Figure 6.13. Figure 6.7 shows that PAIRWISE-TORUS achieves significantly smaller deviation (p<0.001) than PAIRWISE-NOTORUS for SMALL+EASY, LARGE+EASY, SMALL+HARD, and LARGE+HARD. PAIRWISE-TORUS achieves significantly smaller deviation than ALL-PAIRS-NOTORUS for SMALL+EASY (p<0.001), LARGE+EASY (p=0.006), SMALL+HARD (p<0.001) , and LARGE+HARD (p<0.001).

$$\frac{1}{|V|} \sum_{v \in V} \frac{|\theta_v - min\theta_v|}{\theta_v} \tag{6.3}$$

Link Crossings—The negative effect of line crossings on readability of graphs is well studied by Purchase [Pur02] and Huang et al. [HEH09]. Figure 6.14 and Figure 6.7 show that PAIRWISE-TORUS achieves significantly ($p < 0.001$) fewer crossings than PAIRWISE-NOTORUS and ALL-PAIRS-NOTORUS for SMALL+EASY, LARGE+EASY, SMALL+HARD, and LARGE+HARD. Furthermore, PAIRWISE-TORUS significantly ($p = 0.0366$) achieves fewer crossings than ALL-PAIRS-TORUS for SMALL+EASY. As shown in the box plots in Figure 6.15, the difference of crossings is much more obvious for our HARD graphs than the EASY graphs.

In accord with the findings for small networks in chapter 5, our results show that torus-based layout have clear benefits over traditional node link diagrams for larger networks at least for these metrics.

6.3.4 New cluster readability metrics: cluster distance

We use a new metric for cluster readability: *cluster distance* to measure how well a layout algorithm is able to separate clusters. For a given layout, for all pairs of clusters whose convex hulls are not overlapping we compute:

Minimum separation between convex hulls—measures space between non-overlapping clusters. A larger value indicates more distance between the boundaries of clusters in a given layout. For torus, we first identify a convex polygon in a 3 × 3 torus coordinate for each cluster. We then use the Gilbert–Johnson–Keerthi (GJK) algorithm [OG97] to determine the minimum distance between convex polygons.

	WrapCost		Wrapping Links		Time (sec)
Auto Pan	*No Pan*	*Best Pan*	*No Pan*	*Best Pan*	*Best Pan*
Small+Easy	4.64	2.41	522.7	324.7	0.04
Small+Hard	4.28	2.49	514.5	354	0.04

Fig. 6.18.: Automatic panning results: mean wrapCost, number of wrapping links across the boundary and running time of 20 random runs for SMALL graphs at high (0.4) and low (0.3) modularity (with layout pre-computed by Algorithm 1); graph metrics as shown in Figure 6.7.

Then, for all pairs of clusters that are overlapping, we compute:

Minimum penetration depth between convex hulls—To measure the minimum translation vector required to separate the convex hulls of the cluster pair, as a measure of cluster overlap. A larger value indicates more severely overlapping clusters. We use the Expanding Polytope Algorithm (EPA) which is based on Minkowski sum to compute the penetration depth by Van [Van01].

Then, for each pair of clusters we define *cluster distance* as the negative minimum penetration depth if they are overlapping, or the minimum separation between convex hulls if they are not overlapping. Across all pairs we take the average minimum cluster distance as a metric of cluster separateness across the whole graph.

Graphics of statistical results and box plots of cluster distance are summarised in Figure 6.16 and Figure 6.17. Overall, we found:

- PAIRWISE-TORUS significantly outperformed ALL-PAIRS- NOTORUS, ALL-PAIRS-TORUS in cluster distance for both low and high modularity at 0.25, 0.3, 0.35, and 0.4 for both SMALL and LARGE.
- PAIRWISE-TORUS outperformed PAIRWISE-NOTORUS in cluster distance for modularity at 0.25, 0.3 for SMALL and LARGE.
- We also found PAIRWISE-NOTORUS significantly outperformed ALL-PAIRS-NOTORUS for cluster distance under modularity at 0.4 for SMALL and LARGE.
- As PAIRWISE significantly outperformed ALL-PAIRS, we used PAIRWISE to perform user evaluation for NOTORUS and TORUS in the next section.

6.3.5 Automatic panning results

We conducted a small empirical analysis of the Auto Pan Algorithm 2 in terms of number of wrapped links. Results are summarised in Table Figure 6.18. Automatic

panning significantly improves the number of wrapped links, while the *wrapcost* penalty prefers wrapping long links over short links, which tends to keep clusters unwrapped. The difference is visible in Figure 6.2.

6.4 User Study 4: Cluster Visualisation Readability

The previous section demonstrated the improvement provided by our new algorithm PAIRWISE with automatic panning, in terms of graph aesthetics and cluster readability metrics. In this section, we investigate if our torus drawings with automatic panning are more effective for people to use than standard unwrapped representations for high-level network topology analysis tasks.

As mentioned in chapter 1, visual cluster analysis is an important application of network visualisation. The hypothesis that we test in our user study is whether the additional spreading afforded by toroidal layout in terms of the new cluster separation metric (subsection 6.3.4) also leads to better human perception of clusters. We do not use the ALL-PAIRS-Torus algorithm (section 5.1) in the user study because the results of the empirical experiments (Figure 6.7, Figure 6.16) overwhelmingly demonstrate that the new PAIRWISE algorithm is superior.

The particular task that we focus on in this paper is the inspection of community structure, i.e., cluster identification. This study is different from our previous studies in chapter 5 with small graphs (≤ 15 nodes, ≤ 36 links). Our study graphs (subsection 6.4.3) are 4.7 to 8.6 times larger in the number of nodes, 21 to 68 times larger in the number of links than that considered in the previous chapter (section 5.3). In summary, following chapter 5, we are interested in answering the following research question that link to **RG4** (subsection 1.2.4):

[RQ4.4]: Do toroidal layouts with interactive wrapping provide more benefit to perception than a standard unwrapped representation for *cluster identification*?

6.4.1 Techniques & Setup

The techniques in our study are NOTORUS and TORUS generated using our new PAIRWISE algorithm as detailed in section 6.1. In the study, each trial starts with automatic panning for TORUS. We did not show any cluster colours in this study to not hint towards any graph structure. Interactive panning using the mouse is enabled only in the TORUS condition. A user can drag the visualisation such that

when the view is panned off one side of the display, it reappears on the opposite side.

6.4.2 Tasks & Dependent Variables

For our study, we selected two representative network visualisation tasks, inspired by existing task taxonomies by Lee et al. [Lee+06] and Saket et al. [SSK14], and that involve the understanding of clusters. For each task, we record task-completion time (Time), task-error (Error), and subjective user confidence in using for the task in terms of rank (1 or 2) (Confidence). We record pan distance as the overall distance the participant moves the mouse for each trial. We record user ranking for learnability for cluster related tasks in general (Learnability) and overall preference (Overall).

Our tasks are as follows:

- **Cluster Number: Identify the number of clusters (Figure 6.19(a, b))** Participants are required to count the number of clusters they can find in the image. Participants are provided radio buttons to answer 1 to 10. We calculate Error as the absolute difference between the correct answer and the user's response, divided by the correct answer.

- **Node Cluster: Do the two red nodes belong to the same cluster (Figure 6.19(c, d))?** We record participants' responses through multiple-choice questions with the values *yes*, *no*, and *not sure* as options. Error is binary with not-sure counting as error.

6.4.3 Graph Structure

To evaluate effectiveness of toroidal drawings for cluster identification tasks, we used graphs from our graph corpus in subsection 6.3.1, with two levels of difficulty (Easy: modularity=0.4, Hard: modularity=0.3) and two levels of graph size (Small and Large), whose graph metrics are summarised in Figure 6.7. We name these four groups: Small+Easy, Large+Easy, Small+Hard, and Large+Hard. In each group, we randomly selected 5 graphs for trials for each task. The number of clusters for Cluster Number ranged between 4 and 7. For Node Cluster, the number of clusters ranged between 5 and 7, as we found in pilot studies that fewer than five clusters is too easy for Node Cluster. We ran the Pairwise layout algorithm
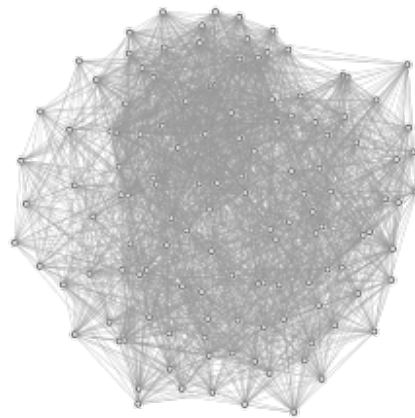
(a) NODE CLUSTER NOTORUS
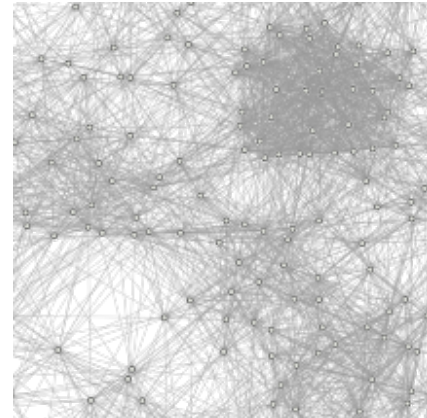
(b) NODE CLUSTER TORUS
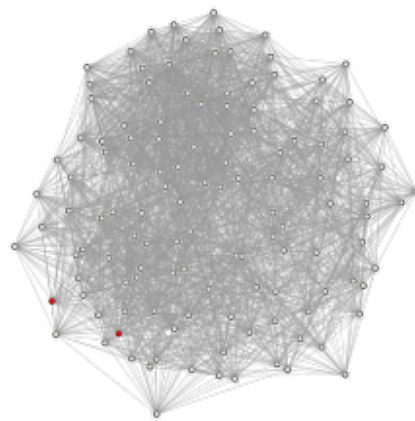
(c) NODE CLUSTER NOTORUS

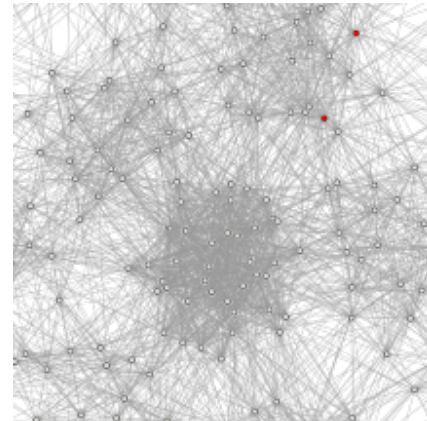(d) NODE CLUSTER TORUS

Fig. 6.19.: Example of CLUSTER NUMBER tasks (a,b) and NODE CLUSTER tasks (c,d) for LARGE+HARD: Modularity 0.3, 130 nodes, 2575 links, 5 clusters (a,b); Modularity 0.3, 126 nodes, 2496 links, 5 clusters (c,d)

20 times for a chosen graph for both NoTorus and Torus. We then selected one layout at random and used it for both NoTorus and Torus for each study trial.

6.4.4 Hypotheses

Our hypotheses were pre-registered with the Open Science Foundation: `https://osf.io/7vbr4`.

For Cluster Number:

- **C4.1**: Torus has better task effectiveness (in terms of time and error) than NoTorus independent of difficulty levels.
- **C4.2**: For Easy graphs, NoTorus has better task effectiveness (in terms of time and error) than Torus (requires certain panning and mental wrapping).
- **C4.3**: For Hard graphs, Torus has better task effectiveness (in terms of time and error) than NoTorus.
- **C4.4**: Participants will report more confidence in Torus than NoTorus.

For Node Cluster:

- **N4.1**: Torus has better task effectiveness (in terms of time and error) than NoTorus independent of difficulty levels.
- **N4.2**: Torus has better task effectiveness (in terms of time and error) than NoTorus for both Easy and Hard tasks.
- **N4.3**: Participants will report more confidence in using Torus than NoTorus.

For participant preference

- **P4.6**: Overall, participants prefer Torus over NoTorus.

6.4.5 Experimental Design

We use a within-subject design (explained in section 2.4) with 2 techniques (Torus, NoTorus) × 2 tasks (Cluster Number, Node Cluster) × 2 level of difficulty (Easy, Hard) × 2 sizes (Small, Large) × 5 recorded repeats. This leaves us with a total of 80 recorded trials per participant. We blocked the study by tasks, i.e., participants would do both techniques with the same task before moving to the second task. For each task block, we counterbalanced the order of the techniques using a full-factorial design. The order of each level of difficulty and size in each technique

was the same: SMALL+EASY→LARGE+EASY→ SMALL+HARD→LARGE+HARD. The order of trials for each technique within each level was randomised.

6.4.6 Participants and Procedures

We recruited 32 participants from local institutes through university's email list and snowballing. 19 were males, 13 were females. The age of participants was between 20 and 50 (mean = 31.5). 22 people reported seldom or never using network diagrams while 10 people often used network diagrams in their work or study.

While run entirely online due to COVID-19 health-concerns, the experimenter supervised each participant through remote video conferencing software to give proper instructions, assure the participants' engagement, and help with eventual questions. The experimental software was loaded in a participant's Google Chrome browser. Each participant shared their screen with the experimenter. The experimenter ensured that each participant used a monitor with resolution no less than 1366×768 pixels. Each study trial used a stimuli with a size of 650×650 pixels. Each trial was correctly loaded in a participant's browser, before the recording started. The experimenter trained each participant to identify a cluster as a graph structure whose links within a cluster are relatively more than the links between clusters. Before each new task and technique, there were 2 training trials. When a participant gave an answer in these training trials, an image with different clusters highlighted in different colours appeared, outlining the cluster. Participants first had to complete all training trials correctly before proceeding to the recorded trials. Each recorded trial had a timeout of 20 seconds to prevent participants from trying to perform precise link counting. For TORUS technique, short animations demonstrating interactive torus wrapping were shown.

6.4.7 Results

All of the participants completed the training and recorded trials. Therefore we recorded performance for 2,560 trials. Since ERROR was not normally distributed, we used Friedman's non-parametric test and Tukey's posthoc multiple pairwise comparison with Bonferroni correction [FMF12] to identify significant differences between NOTORUS and TORUS. The residuals of TIME were normally distributed, visually checked with Q-Q plots and a histogram, supported by a Shapiro-Wilk test [SW65]. The variances of the TIME were equal by Levene's test. We therefore performed a 3-way repeated measures ANOVA to test significant difference in TIME between
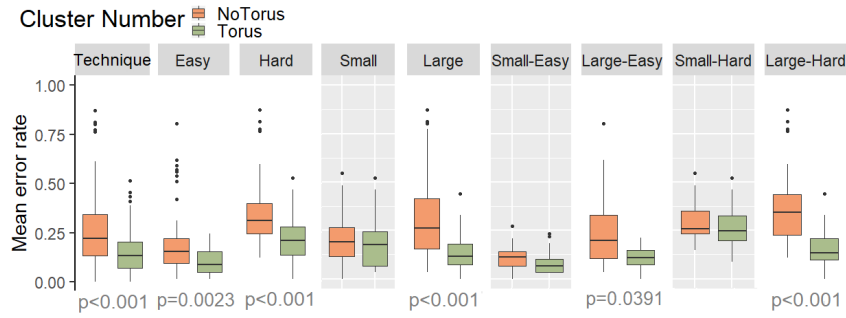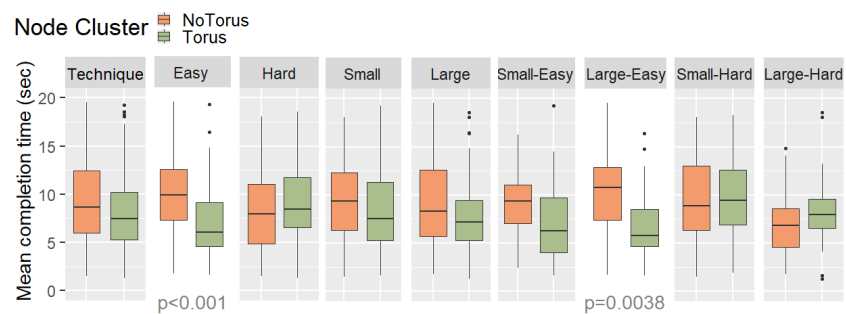
(a) CLUSTER NUMBER error
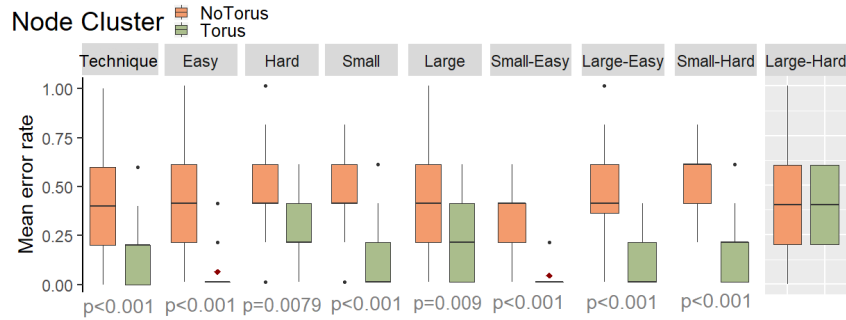
(b) NODE CLUSTER error
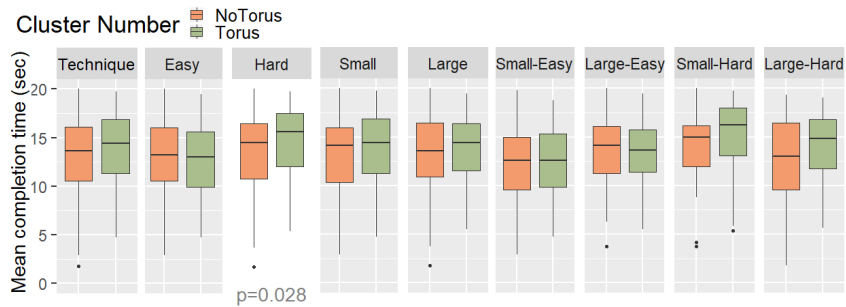
(c) CLUSTER NUMBER time

(d) NODE CLUSTER time

Fig. 6.20.: User evaluation results of mean error and time between NOTORUS and TORUS split by task, difficulty and size. Columns of significant results are shown in white background.
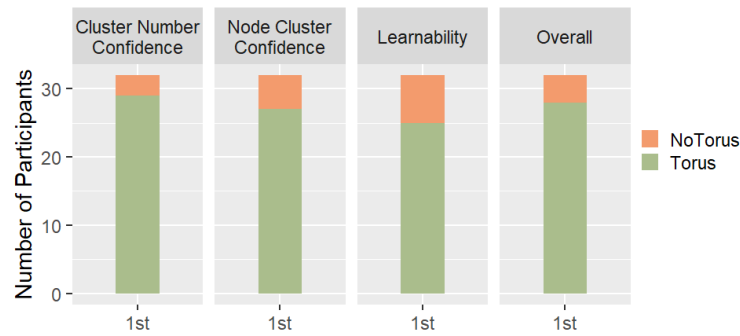
Fig. 6.21.: Subjective user rank of NoTorus and Torus by the number of participants who ranked the condition as their first preferred choice.

NoTorus and Torus. We used paired t-test for posthoc pairwise comparisons with p-value adjustment using Bonferroni correction. Wilcoxon's non-parametric signed rank test was used to analyse paired significance of subjective user rank. Confidence intervals indicate 95% confidence for mean values for all the pairwise comparisons.

Overall, for CLUSTER NUMBER, we found the following significant results. The corresponding box plots can be found in Figure 6.20(a).

- For ERROR, we found TORUS $(.15, SD = .1)$ significantly outperforming NoTorus $(.26, SD = .17)$ for technique $(p < .001)$, as shown in Figure 6.20(a).; TORUS $(.09, SD = .05)$ resulted in significantly less errors than NoTorus $(.18, SD = .16)$ for EASY $(p = .0023)$; TORUS $(.2, SD = .11)$ significantly outperformed NoTorus $(.33, SD = .16)$ for HARD $(p < .001)$; TORUS $(.12, SD = .08)$ has significantly less errors than NoTorus $(.31, SD = .21)$ for LARGE $(p < .001)$; TORUS $(.1, SD = .05)$ significantly outperformed NoTorus $(.25, SD = .2)$ for LARGE+EASY $(p = .0391)$; and TORUS $(.15, SD = .09)$ significantly outperformed NoTorus $(.37, SD = 2)$ for LARGE+HARD $(p < .001)$.

- For TIME, we found NoTorus $(13.7s, SD = 4.65)$ was significantly faster than TORUS $(15.1s, SD = 3.88)$ for only HARD $(p = .028)$, as shown in Figure 6.20(c).

- Participants report significantly more confidence in using TORUS than NoTorus (p<0.001), as seen in Figure 6.21.

For NODE CLUSTER, we found the following significant results. The corresponding box plots can be found in Figure 6.20(b).

- For ERROR, TORUS $(.16, SD = .2)$ was found significantly outperforming NoTorus $(.43, SD = .21)$ for technique $(p < .001)$, as seen in Figure 6.20(b).; TORUS $(.05, SD = .09)$ resulted in significantly less errors than NoTorus $(.39, SD = .22)$

for EASY ($p < .001$); TORUS ($.28, SD = .21$) significantly outperformed NO-TORUS ($.47, SD = .2$) for HARD ($p = .0079$); TORUS ($.09, SD = 14$) had significantly less errors than NOTORUS ($.44, SD = .18$) for SMALL ($p < .001$); TORUS ($.23, SD = .22$) significantly outperformed NOTORUS ($.42, SD = .24$) for LARGE ($p = .009$); TORUS ($.03, SD = .07$) was found significantly outperforming NOTORUS ($.33, SD = .16$) for SMALL+EASY ($p < .001$); TORUS ($.06, SD = .1$) significantly outperformed NOTORUS ($.45, SD = .25$) for LARGE+EASY ($p < .001$); and TORUS ($.16, SD = .16$) significantly outperformed NOTORUS ($.55, SD = .14$) for SMALL+HARD ($p < .001$), as shown in Figure 6.20(b).

- TORUS ($7.22s, SD = 3.92$) was significantly faster than NOTORUS ($9.84s, SD = 3.78$) in time for EASY ($p < .001$); and TORUS ($7.1s, SD = 3.74$) was significantly faster than NOTORUS ($10.5s, SD = 4.09$) for LARGE+EASY ($p = .0038$), as shown in Figure 6.20(d).

- Participants report significantly more confidence in using TORUS than NOTORUS ($p < .001$), as shown in Figure 6.21.

Overall, TORUS improved task effectiveness in terms of error rate by 42.3% on average, compared with NOTORUS for CLUSTER NUMBER task. For NODE CLUSTER task, TORUS improved the task effectiveness in terms of error rate by 62.7% and time by 32.3% on average (i.e., difference between means), as opposed to NOTORUS. Participants reported that TORUS is significantly easier to learn (LEARNABILITY) ($p < .001$) than NOTORUS for our cluster related tasks. TORUS was significantly preferred (OVERALL) ($p < .001$) over NOTORUS in overall rank, as shown in Figure 6.21. Based on these results, for **[RQ4.4]** we rejected hypothesis C4.2 and accepted C4.4, N4.3, P4.6. We accepted C4.1, C4.3, N4.1 for error, and N4.2 for error and time for EASY.

6.4.8 Qualitative User Feedback

For understanding clusters, the majority of participants mentioned that the TORUS with panning provided for less overlap between nodes and was better at spreading nodes and clusters more spatially, thus reducing visual clutter and increasing readability. While our automatic panning optimises the view a user sees first, interaction further helps to find the best view possible. Some participants reported that CLUSTER NUMBER required more panning to identify the graph structure. For NODE CLUSTER, on the other side, participants reported that they rarely used interactive panning, except for verifying their answer. This is supported by our measurements that NODE
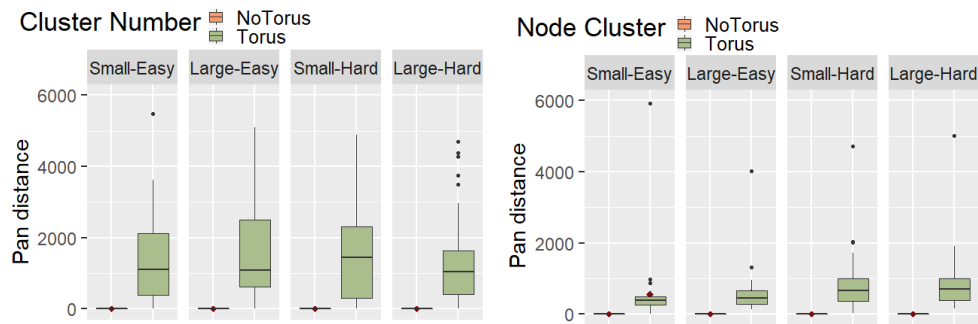
Fig. 6.22.: Box plots of panning distance results of NOTORUS and TORUS.

CLUSTER, required less user interaction (majority less than 1000 pixels per trial), as shown in Figure 6.22. There are 4 participants who reported they favoured NOTORUS as it gives a full overview of the networks and did not require to identify the same piece of cluster wrapped around top-bottom or left-right.

6.5 Discussion

Our study results indicate that NOTORUS is sometimes faster than TORUS when counting the number of clusters for HARD (Fig. 6.20(c)), but the error rate in the NOTORUS condition was generally much higher (Fig. 6.20(a)). We believe participants were sometimes faster with NOTORUS because they were simply guessing the answer. Our new study is intended to be complementary to the studies described in chapter 5 by evaluating an important task not considered in that study. Since our new layout algorithm produces high-quality torus layout with fewer crossings, larger incidence angles and less stress, we would expect that the results from the previous chapter for low-level connectivity understanding and path following tasks would also be reproducible for larger networks using our new algorithm. However, we leave such an evaluation to future work.

6.6 Conclusion and Future Work

In this chapter, we have explored new network layout algorithms that take advantage of the toroidal topology such that when wrapping specific links around the boundaries, it allows further relaxation of node positions to reduce visual clutter,

compared with our torus layout algorithm in chapter 5 and standard non-wrapped layouts. We addressed **RG4** (subsection 1.2.2).

We presented a new algorithm based on Pairwise gradient descent for torus-based layout of networks that is more robust in terms of graph aesthetics and running time than All-Pairs torus layout method described in chapter 5. Furthermore, we improved the resulting layouts by using a novel algorithm that minimises the number of link wrappings across the boundary by centring elements of interest in the display.

Using this algorithm, we have investigated whether interactive toroidal wrapped layouts provide advantages over traditional flat network layouts for higher-level tasks of understanding network structures. Both an analysis of graph aesthetics metrics and a user study have clearly shown that they do: participants were able to more accurately determine the number of clusters in a graph and more quickly and accurately determine whether two nodes are in the same cluster with the toroidal wrapped layouts. Furthermore, participants preferred the toroidal wrapped layouts to the standard non-wrapped flat node-link. This is the first demonstration that torus-based layouts provide real-world benefits over traditional node-link layouts.

Our research also reveals several directions for future research, including exploration of the proposed torus layout approach for understanding complex multidimensional data that needs cleaner representations such as MDS (described in subsection 9.4.1), investigating if the proposed layout methods could also provide benefits for understanding specific real-world complex and dense network structures, such as biological and social network applications (detailed in section 9.3) and evaluate real-world usefulness with domain experts. Furthermore, improving the network layout algorithm's runtime performance is a major direction (section 9.3).

Spherical Wrapping for Geographical Data Visualisations

7

> *The fact I couldn't move the pictures was frustrating [for the static map projections] and I think I didn't get many of the guesses right.*
>
> — **Anonymised user study participant, P4**
> Direction estimation tasks

We have investigated the utility of interactive cylindrical and toroidal wrapping of visualisations which have shown benefits for understanding cyclic time series data (chapter 4) and understanding the structure of network diagrams (chapter 6). Crucially, these visualisations are interactive and the viewer can pan the projection so that it "wraps" around the plane. Inspired by these results, we now investigate the utility of projections of the surface of a different geometric object, the sphere, onto a 2D plane. This chapter addresses **RG5**. As discussed in section 2.2, a plethora of map projection techniques exists with various affordances for wrapping (panning). However, existing studies exclusively focus on readability of static maps. In this chapter, we investigate the effect of 2D interactive spherical wrapping on different map projections for geographic comprehension tasks.

Beside geographic data, spherical projections have applications for non-geographic data. For example, there can be advantages to laying out networks (a type of abstract data which has no inherent geometry) on the surface of a sphere such that there is no arbitrary edge to the display or privileged centre (subsection 2.1.4). We explore the readability of networks laid out spherically and then projected to a flat surface using interactive spherical wrapping and rotation in the next chapter.
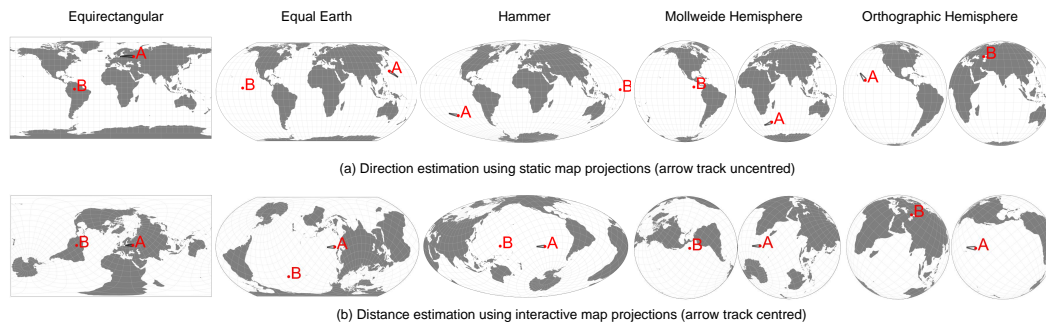
(a) Direction estimation using static map projections (arrow track uncentred)

(b) Distance estimation using interactive map projections (arrow track centred)

Fig. 7.1.: Top row: Direction estimation task of geographic data using the five different spherical projections evaluated in Study 1, with (a) and without (b) interaction.

Projection	Type	Area-preserving	Left/right edges	Top/bottom edges	Naturalistic
Equirectangular	continuous	no	straight	straight	very low
Equal Earth	continuous	yes	curved	straight	low
Hammer	continuous	yes	curved	curved	low
Mollweide Hemisphere	interrupted	yes	circular	circular	high
Orthographic Hemisphere	interrupted	no	circular	circular	very high

Fig. 7.2.: Key characteristics of the five map projections tested in Study 1.

7.1 Map Projection Techniques

In this section, we investigate the effect of interaction on readability of different map projection techniques and identify the projection techniques which best support geographic comprehension tasks, such as distance, area and direction estimation (section 7.2).

We chose five representative map projections for the study. To explore map readability, we aim to cover a wide range of distortion properties such as preservation of area, distance, shape, direction as discussed in chapter 2, and user preference [Šav+15]. We demonstrate the key characteristics of the map projections in Figure 7.2 and discuss the details as follows:

EQUIRECTANGULAR ▭ projects the earth onto a space-filling rectangle with the north and south poles extending along the top and bottom edges, respectively. Rectangular projection is the most widely used projection, with common variations including Mercator or Plate Carrée [Jen+17; Šav+15]. EQUIRECTANGULAR ▭ projections preserve distances along all meridians and are useful when differences in latitude are measured [Jen+17]. However, it does not preserve the relative size of areas. Furthermore, it has been found confusing for tasks requiring understanding how the edges connect to each other, such as predicting the path of air plane routes crossing the top and bottom edge of the map [Hen15].

EQUAL EARTH ⬭ is similar to EQUIRECTANGULAR ▭ but relaxes the rectangle with rounded corners, diminishing the strength of horizontal distortion near the poles. Furthermore, the similarly shaped Robinson projection has received good subjective ratings from viewers [Šav+15]. Unlike Robinson projection, EQUAL EARTH ⬭ preserves the relative size of the areas well.

HAMMER ⬭ further diminishes horizontal stretching by projecting the earth onto an ellipse such that the poles are points at the top and bottom. It preserves the relative size of areas and generally has similar properties to EQUAL EARTH ⬭ [Yan+18; Jen+17]. The similarly shaped Mollweide projection has been shown less confusing than EQUIRECTANGULAR ▭ when judging the continuity of air plane routes as described above [Hen15]. Furthermore, map readers prefer to see the poles as points rather than lines [Šav+15]. It has also been found pleasing to many cartographers than other projections due to its elliptical shape [Jen+17].

MOLLWEIDE HEMISPHERE ⬭ projects the earth onto two circles (hemispheres). This again helps diminish horizontal stretching but introduces the cost of new tears (interruption) that a viewer needs to mentally close.

ORTHOGRAPHIC HEMISPHERE ⬭⬭ is also hemispheric. It has a naturalistic appearance as it shows the globe (from both sides) seen from an infinite distance. However, compared with MOLLWEIDE HEMISPHERE ⬭⬭, it is not area-preserving.

7.1.1 Interaction techniques

For each projection, we created both STATIC and INTERACTIVE versions. With INTERACTIVE, a user can freely move regions of interest to the centre of the projection, thus reducing their distortion. Examples of the effect of interaction are shown in the top row in Figure 7.1.

We used D3 libraries for creating all the map projection techniques. For INTERACTIVE, we follow Yang et al. [Yan+18] and use versor dragging which controls three Euler angles. This allows the geographic start point of the gesture to follow the mouse cursor [Dav13].

We implemented ORTHOGRAPHIC HEMISPHERE ⬭⬭ with two Orthographic map projections with one showing the western and the other showing the eastern hemisphere. They are placed close together as shown in the top row of Figure 7.1. When one hemisphere is dragged, the Orthographic projection of the other hemisphere automatically adjusts three-axis rotation angles such that it shows the correct opposite hemisphere.

Spherical rotation of all projections is demonstrated in the OSF repository[1]

7.2 User Study 5: Map Projection Readability

The goal of our first study was to understand the effectiveness of (i) different projections for geographic data as well as the (ii) benefit of interactively changing the centre point of these projections.

7.2.1 Techniques

The techniques are EQUIRECTANGULAR ▭, EQUAL EARTH ⬭, HAMMER ⬭, MOLL-WEIDE HEMISPHERE ⬭, and ORTHOGRAPHIC HEMISPHERE ⬭, described in section 7.1. Each technique is given both static images without the ability to rotate, and with interactive spherical rotation, using the mouse. A user can rotate the visualisation such that when the view is panned off one side of the display, it either reappears on the opposite side (left-right) or is horizontally mirrored (top-top, bottom-bottom), as shown in the top-row of Figure 7.1. The area of the rectangular bounding box of each map projection condition was fixed at 700×350 pixels.

7.2.2 Tasks & Datasets

We selected three representative geographic data visualisation tasks. They were also used in existing map projection studies [Yan+18; Hen15; HAA16; Car10].

DISTANCE COMPARISON •••: *Which pair of points (pair A or pair B) represents the greater geographical distance on the surface of a globe?* Participants had to compare the true geographical distance (*as the crow flies*) between two pairs of points on the projection. Participants were provided with radio buttons to answer *A*, *B*, or *not sure*. The results of *not sure* were counted towards the number of incorrectly answered trials. We created data sets for two difficulties through extensive piloting, based on the difference of point distance across point pairs: 10% difference (easy) and 5% difference (difficult). The geographic point pairs were randomly chosen, constrained by their individual angular distance (measured in differences in geographical coordinates) between 40° (approx. 4444*km*) and 60° (approx. 6666*km*) [Yan+18], and a minimum 60° distance across point pairs. We did not set any upper-bound to

---

[1]Interactive examples can be found in `https://observablehq.com/@kun-ting`.
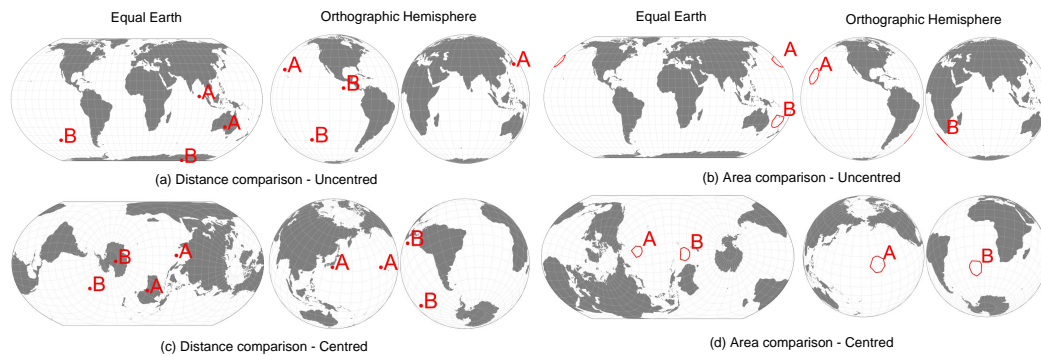
Fig. 7.3.: Sample trials of Equal Earth and Orthographic Hemisphere for distance and area comparison tasks. STATIC (a,b) may result in uncentred view. INTERACTIVE (c,d) allows a user to drag the map to find the best angle (centred) to answer the task. Direction tasks are shown in Figure 7.1

not to bias any projection. We created an additional quality control trial with 40% difference of node distance to test a participant's attention. An example is shown in Figure 7.3(a,c).

AREA COMPARISON ⬟⬤: *Which polygon (A or B) covers the greater geographical area on the surface of a globe?* Participants had to compare the size (area) of the polygons. Participants were provided with radio buttons to answer *A*, *B*, or *not sure*. ERROR, defined in subsection 7.2.6, is binary with not-sure counted as error. Again, we created data sets for two difficulties based on the difference in area they cover: 10% difference (easy) and 7% difference (difficult). Eight geographic points of convex polygons were randomly chosen using the same method as Yang et al. [Yan+18], constrained by the individual geographic area between 40 and 60, and a minimum 60° angular distance between centroids of pairwise polygons. There is no upper-bound for the same reason as above. We create an additional quality control trial with 40% difference in area to test a participant's attention. An example is shown in Figure 7.3(b,d).

DIRECTION ESTIMATION ↬: *Does the trajectory of dot A hit or miss dot B on the surface of a globe?* Participants had to assess whether the trajectory, indicated by an arrow track, from point *A* hits or misses point *B*. Participants were provided with radio buttons to answer *Hit, Miss*, or *not sure*. We randomly created pairs of geographic points (A, B, and arrow head) with a minimum angular distance of 60° between A and B. There is no upper-bound. For trials where the trajectory of A misses B, the angular distance between trajectory and B was constrained to 40°. Examples of this task are shown in the first two rows of Figure 7.1 for each projection technique, where A misses B for equal earth and orthographic hemisphere. For this task, there was only one level of difficulty. We create an additional quality control

trial with dot A at the centre and arrow track hitting to a dot B aligned horizontally at the centre to test a participant's attention.

7.2.3 Hypotheses

Our hypotheses were pre-registered with the Open Science Foundation: `https://osf.io/vctfu`.

M5.1: INTERACTIVE *has a lower error rate* than all static map projections across all tasks. Our intuition is that interaction allows regions of interest to be centred and, thus, their distortion reduced.

M5.2: INTERACTIVE *projections have longer task-completion time than static across all projections and tasks.* Users will spend time interacting to find an optimal centre point for each projection to solve the task.

M5.3: *For interactivity, users prefer* INTERACTIVE *to* STATIC *across all tasks.* Intuition as above.

M5.4: INTERACTIVE ORTHOGRAPHIC HEMISPHERE ◉ *has a lower error rate* than other interactive non-hemisphere (EQUIRECTANGULAR ▭, EQUAL EARTH ◯, HAMMER ◯) projections across all tasks. This is inspired by a virtual globe study by Yang et al. [Yan+18]. Our intuition is that an interactive view of the 3D globe will have similar benefits to the VR representation.

M5.5: *Users prefer* INTERACTIVE ORTHOGRAPHIC HEMISPHERE ◉ *over all other* INTERACTIVE *projections across tasks.* Intuition as above.

7.2.4 Experimental Setup

Design is within-subject per task, where each participant performed *one* task (DISTANCE COMPARISON •↔•, AREA COMPARISON ♠♠, DIRECTION ESTIMATION ↗) on all projections (EQUIRECTANGULAR ▭, EQUAL EARTH ◯, HAMMER ◯, MOLLWEIDE HEMISPHERE ◉, ORTHOGRAPHIC HEMISPHERE ◉) in both STATIC and INTERACTIVE and in all levels of difficulty (*easy, hard*). Each of these 10 conditions for AREA COMPARISON ♠♠ and DISTANCE COMPARISON •↔• was tested in 12 trials with two difficulty levels (6 easy, 6 hard). For DIRECTION ESTIMATION ↗, we reduced the number of trials to 8 as pilot participants reported the direction estimation was too difficult for long distance. Similar to our prior visualisation crowd-sourced studies described in section 4.2 and Brehmer et al. [Bre+18]'s studies, we randomly

inserted a quality control trial with low difficulty in addition to normal trials to each condition to test participants' attention. The study was blocked by interactivity. Within each block, the order of the map projection techniques was counterbalanced using William et al.'s Latin-square design [Wil49]. This technique resulted in 10 possible orderings for the 5 projections while the order of projections in each block was the same. Each recorded trial had a timeout of 20 seconds, inspired from pilot studies.

7.2.5 Participants and Procedures

We crowd-sourced the study via the *Prolific Academic system* [PS18]. Participants on Prolific Academic have been reported to produce data quality comparable to Amazon Mechanical Turk [Pee+17]. Many visualisation studies have used this platform, including our prior studies described in section 4.2 and Satriadi et al. [Sat+21]'s study. We hosted an online study on the Red Hat Enterprise Linux (RHEL7) system. We set a pre-screening criterion on performance that required a minimum approval rate of 95% and a minimum number of previous successful submissions of 10. We also limited our study to desktop users with larger screens. We paid £5 (i.e. £7.5/h), considered to be a good payment according to the Prolific Academic guidelines.

We recorded 120 participants who passed the attention check trials, completed the training and recorded trials. This comprised 4 full counterbalanced blocks of participants ($10 \times 4 \times 3$ tasks). 57 of our participants were females, 63 were males. The age of participants was between 18 and 55. 7 participants rank themselves as *regularly* using GIS or other tools to analyse geographical data. 105 occasionally read maps, e.g., using Google Map or GPS systems. 8 had very little experience with any sort of map.

Before starting the study, each participant had to complete a tutorial explaining projection techniques and tasks. The tutorial material contained *Tissot's indicatrix* [Sny87], a set of circular areas placed on both the poles of the equator, indicating the type and magnitude of area, shape, and angular distortion in a given projection. The setting has been inspired by Yang et al. [Yan+18]. An online demonstration of the study is available: `https://observablehq.com/@kun-ting/gansdawrap`

7.2.6 Dependent Variables

We measured **task-completion-*time*** (TIME) for each trial in milliseconds, counted between the first rendering of the visualisation and the mouse click of the *answer trial* button, which hid the visualisation and showed an interface for the participants to input their answers. We measured the ***error rate*** (ERROR) as the ratio of incorrect over all answers. We asked participants to ***rank*** (PREF) each map projection individually within the *static* and the *interactive* block according to their perceived effectiveness. We also ask participants to provide their justifications for the rankings as ***qualitative feedback***. After they completed both blocks, we recorded participants' preference of the interactivity between STATIC and INTERACTIVE individually for each projection, and their overall preference between STATIC and INTERACTIVE.

7.2.7 Statistical Analysis Methods

We used *sqrt*-transformation for TIME to meet the normality assumption. We then used linear mixed modelling to evaluate the effect of independent variables on the dependent variables [Bat+15]. We modelled all independent variables (five map projections, two interaction levels and two difficulty levels) and their interactions as fixed effects. We evaluated the significance of the inclusion of an independent variable or interaction terms using log-likelihood ratio. We then performed Tukey's HSD post-hoc tests for pairwise comparisons using the least square means [Len16]. We used predicted vs. residual and Q—Q plots to graphically evaluate the homoscedasticity and normality of the Pearson residuals respectively. For ERROR and PREF, as they did not meet the normality assumption, we used the *Friedman* test to evaluate the effect of the independent variable, as well as a Wilcoxon-Nemenyi-McDonald-Thompson test for pairwise comparisons. We also used the Wilcoxon signed-rank test for comparing the accuracy of static and interactive map projections. The confidence intervals are 95% for all the statistical testing. We demonstrate the *error rate* and *time* in Figure 7.4. We show interactivity and map projection rankings as stacked bar charts in Figure 7.5, Figure 7.6, and Figure A.1.

Following existing work which reports statistical results with standardised effect sizes [OJK18; Hum19; Yog+20] or simple effect sizes with confidence intervals [Bes+17; BAI17], we interpret the standardised effect size for a parametric test using Cohen's d classification, which is 0.2, 0.5, and 0.8 or greater for small, moderate, and large effects, respectively [Coh13]. For non-parametric tests, we interpret the

standardised effect size for a Wilcoxon's signed-rank test using Cohen's r classification, which is 0.1, 0.3, and 0.5 or greater for small, moderate, and large effects, respectively [Coh13; Pal13].

7.2.8 Results

We report on the most significant findings for DISTANCE COMPARISON ●••●, AREA COMPARISON ♠♠, and DIRECTION ESTIMATION ↵➚ visually in Figure 7.4 and Figure 7.5.

In the following, significance values are reported for $p < .05(*)$, $p < .01(**)$, and $p < .001(***)$, respectively, abbreviated by the number of stars in parenthesis.

For DISTANCE COMPARISON ●••●, we found INTERACTION $(***)$ and MAP PROJECTION × DIFFICULTY $(***)$ both had a significant effect on TIME. For INTERACTION, post hoc analysis shows that STATIC was faster than INTERACTIVE $(***)$. For MAP PROJECTION × DIFFICULTY, post hoc analysis shows that in EASY, EQUIRECTANGULAR ▭ and EQUAL EARTH ⬭ were faster than ORTHOGRAPHIC HEMISPHERE ◐◑ $(**)$ and MOLLWEIDE HEMISPHERE ◐◑ $(***)$. HAMMER ⬭ was also faster than MOLLWEIDE HEMISPHERE ◐◑ $(***)$. In HARD, EQUAL EARTH ⬭ was faster than ORTHOGRAPHIC HEMISPHERE ◐◑ $(*)$ and MOLLWEIDE HEMISPHERE ◐◑ $(***)$.

We found MAP PROJECTION had a significant effect on *error rate* in HARD $(*)$. ORTHOGRAPHIC HEMISPHERE ◐◑ was more accurate than EQUAL EARTH ⬭ $(*)$ and HAMMER ⬭ $(*)$. We also found INTERACTIVE was more accurate than STATIC in all MAP PROJECTION $(***)$.

In STATIC MAP PROJECTION, participants preferred EQUAL EARTH ⬭ over HAMMER ⬭ $(*)$, MOLLWEIDE HEMISPHERE ◐◑ $(*)$, and ORTHOGRAPHIC HEMISPHERE ◐◑ $(**)$.

For AREA COMPARISON ♠♠, we found INTERACTION $(***)$, MAP PROJECTION × DIFFICULTY $(***)$ and MAP PROJECTION × INTERACTION $(**)$ had a significant effect on *time*. For AREA COMPARISON ♠♠, STATIC was faster than INTERACTIVE $(***)$. For MAP PROJECTION × INTERACTION $(**)$, INTERACTIVE EQUAL EARTH ⬭ was faster than INTERACTIVE HAMMER ⬭ $(**)$.

We found MAP PROJECTION had a significant effect on ERROR in INTERACTIVE $(*)$, EASY STATIC $(*)$ and HARD INTERACTIVE $(*)$. MOLLWEIDE HEMISPHERE ◐◑ was more accurate than EQUIRECTANGULAR ▭ $(*)$. STATIC EQUAL EARTH ⬭ $(*)$ and
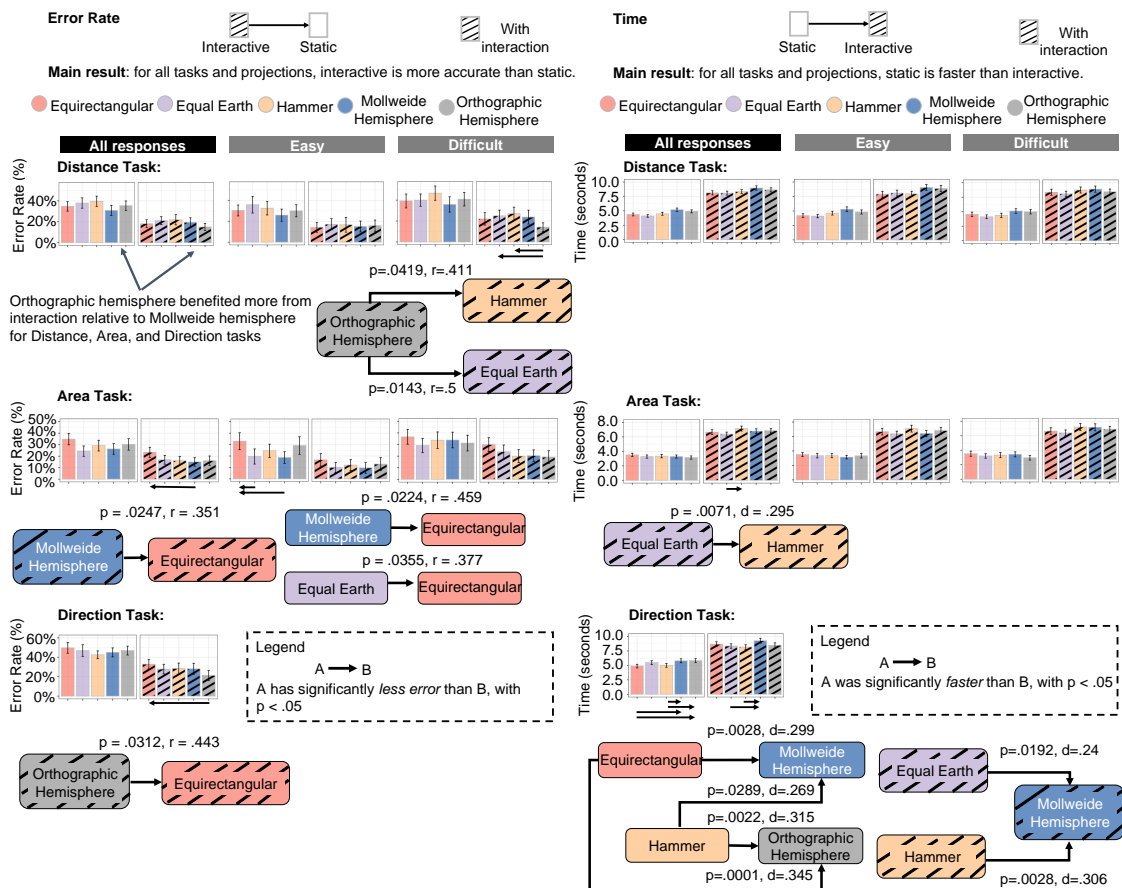
Fig. 7.4.: Error rate (left) and Time (right) results. Significant differences between projections are shown as arrows. Significant differences between interactive and non-interactive conditions are **omitted** to improve readability. Error bars indicate 95% confidence intervals. Effect size results for Cohen's r and Cohen's d [Coh13] are presented for ERROR and TIME, respectively. Statistically significant results are highlighted in flow diagrams below the bar charts. Bars and boxes with stripe patterns refer to interactive conditions. Overall, equal earth performed well for TIME and ERROR for some tasks. Orthographic hemisphere benefited more from interaction than the mollweide hemisphere for ERROR.
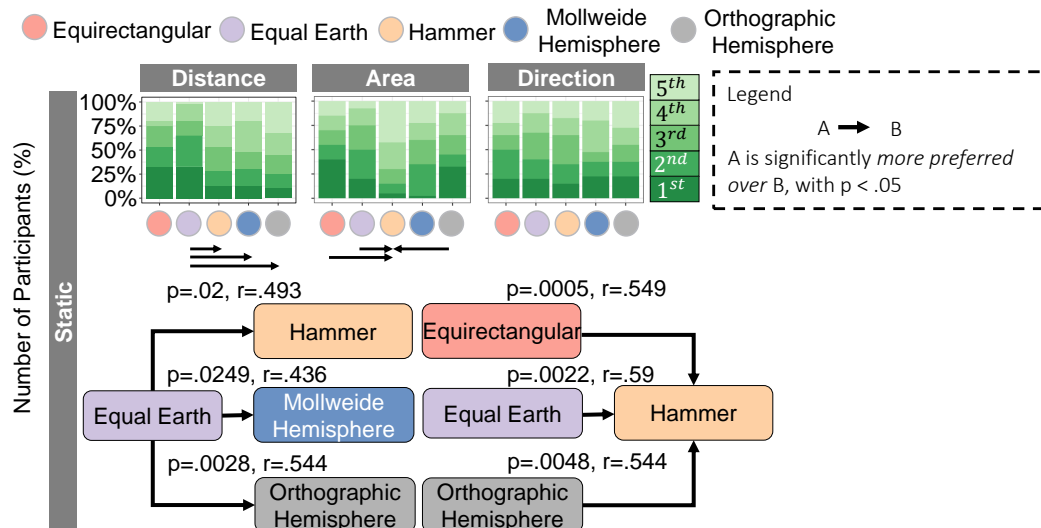
Fig. 7.5.: Subjective user rank of map projections within the *static* group for three tested tasks. Lower rank indicates stronger preference. Arrows indicate statistical significance with $p < 0.05$. Overall, equal earth was found to be preferred over hemispheric projections for the distance comparison task, while hammer was preferred over by other projections for area comparison task.

STATICMOLLWEIDE HEMISPHERE ⬭ (∗) were more accurate than STATIC EQUIRECT-ANGULAR ⬛ in EASY. INTERACTIVE MOLLWEIDE HEMISPHERE ⬭ and INTERACTIVE INTERACTIVE tended to be more accurate than INTERACTIVE EQUIRECTANGULAR ⬛, but not statistically significant (with $p = 0.09, 0.07$ respectively). We also found INTERACTIVE was more accurate than STATIC in all MAP PROJECTION (all ∗ ∗ ∗).

In STATIC MAP PROJECTION, participants preferred EQUIRECTANGULAR ⬛ (∗ ∗ ∗), EQUAL EARTH ⬭ (∗∗) and ORTHOGRAPHIC HEMISPHERE ⬭ (∗∗) than HAMMER ⬭.

For DIRECTION ESTIMATION ↗, we found MAP PROJECTION (∗ ∗ ∗), INTERACTION (∗ ∗ ∗) and MAP PROJECTION × INTERACTION (∗∗) had a significant effect on *time*. For DIRECTION ESTIMATION ↗, STATIC was faster than INTERACTIVE (∗ ∗ ∗). In STATIC MAP PROJECTION, EQUIRECTANGULAR ⬛ and EQUAL EARTH ⬭ were faster than MOLLWEIDE HEMISPHERE ⬭ and ORTHOGRAPHIC HEMISPHERE ⬭ (all ∗∗). In INTERACTIVE MAP PROJECTION, EQUAL EARTH ⬭ was faster than MOLLWEIDE HEMISPHERE ⬭ (∗∗).

We found MAP PROJECTION had a significant effect on *error rate* in INTERACTIVE. ORTHOGRAPHIC HEMISPHERE ⬭ was more accurate than EQUIRECTANGULAR ⬛ (∗). We also found INTERACTIVE was more accurate than STATIC in all MAP PROJECTION (∗ ∗ ∗).
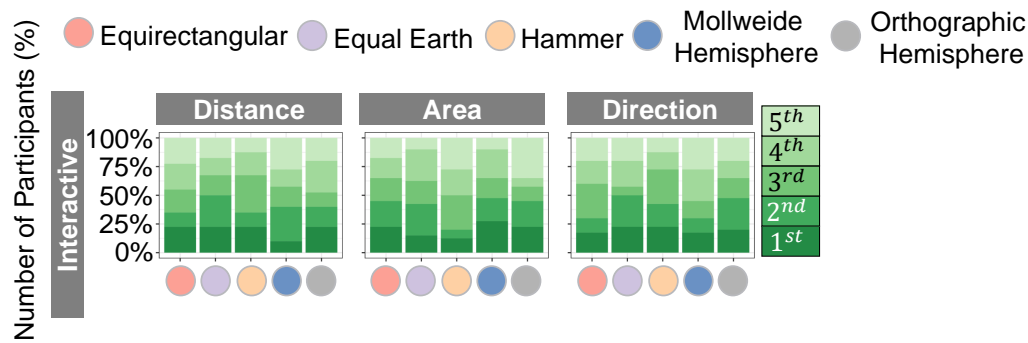
Fig. 7.6.: Study 5: Subjective user rank of map projections within the *interactive* group for three tested tasks. Higher rank indicates stronger preference. Interaction makes the preference between *interactive* projections similar (not statistically significant). Effect sizes can be found in Figure A.5.

We did not find a significant difference in preference for INTERACTIVE groups.

7.3 Key Findings and Discussion

Interaction improved error rate and is preferred over static map projections while taking participants longer time to complete (across all tasks). This main effect was found statistically significant with moderate and large effect sizes in all three tested tasks: DISTANCE COMPARISON ●→● (Figure A.2), AREA COMPARISON ◆● (Figure A.3), and DIRECTION ESTIMATION ↶ (Figure A.4). The corresponding error bar graphs are shown in Figure 7.4 and stacked bar chart of user preference of interactivity in Figure A.1). For DISTANCE COMPARISON ●→● and DIRECTION ESTIMATION ↶, INTERACTIVE is always better (significantly less error and more preferred) than STATIC. For AREA COMPARISON ◆●, INTERACTIVE has less error than the corresponding non-interactive projection but not necessarily all other non-interactive projections. INTERACTIVE was significantly preferred (moderate effects) for HAMMER ⬭ and EQUAL EARTH ⬭ but not necessarily other projections for AREA COMPARISON ◆●. However, INTERACTIVE was significantly preferred in the overall user rank over STATIC (top horizontal bar in Figure A.1) for DISTANCE COMPARISON ●→● (large effects), DIRECTION ESTIMATION ↶ (large effects) and AREA COMPARISON ◆● (moderate effects). INTERACTIVE is always slower (large effects) than STATIC (Figure 7.4-Time). Therefore, we accept M5.1, M5.2, M5.3.

These results provide strong evidence that, with interactions, participants were able to find a better projection centre than the default one in a static map. Some

participants explicitly mentioned the benefits of having interaction and their preference, e.g., *"when moved [,the interactive conditions makes] it easier to judge when [the areas] were both placed in the middle in the least distorted part of the map."* (P20, Area-INTERACTIVE), and *"The fact I couldn't move the pictures was frustrating [for the static conditions] and I think I didn't get many of the guesses right."* (P4, Direction-STATIC).

The choice of projections makes less difference and depends on tasks. However, overall, we found that equal earth and orthographi hemisphere performed well, while equirectangular may be a poor choice, organised in the following key findings.

In continuous projections, EQUAL EARTH ⬭ performed well in terms of ERROR for AREA COMPARISON ◆●, TIME for AREA COMPARISON ◆● and DIRECTION ESTIMATION ↗, and PREF-STATIC for both DISTANCE COMPARISON ●→● and AREA COMPARISON ◆●. EQUAL EARTH ⬭ was not significantly worse than any other continuous projections. We found that for static projections, EQUAL EARTH ⬭ tended to be more accurate (moderate effects) than EQUIRECTANGULAR ▭ for Area-Easy. With interaction, EQUAL EARTH ⬭ tended to be faster (small effects) than MOLLWEIDE HEMISPHERE ⬭ for DIRECTION estimation ↗, and HAMMER ⬭ for Area-All (Figure 7.4-Time). Though the time results are statistically significant, the small effect sizes seem to indicate that the choice of projection makes a slight difference [Hel+21; Coc+20; Hum19; SS19]. For PREF-STATIC, EQUAL EARTH ⬭ was significantly preferred over HAMMER ⬭ (moderate effect), MOLLWEIDE HEMISPHERE ⬭ (moderate effect), and ORTHOGRAPHIC HEMISPHERE ◎ (large effect) for DISTANCE COMPARISON ●→● (Figure 7.5-Distance). Furthermore, there is a strong evidence with important effects for PREF-STATIC that EQUIRECTANGULAR ▭, EQUAL EARTH ⬭, and ORTHOGRAPHIC HEMISPHERE ◎ were significantly preferred over HAMMER ⬭ (large effects) for AREA COMPARISON ◆● (Figure 7.5-Area). With interaction, there are no significant differences in user preference. This result is shown in Figure 7.6.

User preference of STATIC EQUAL EARTH ⬭ partially confirms Šavrič et al. [Šav+15], who found Robinson projection (which is similarl to EQUAL EARTH ⬭) was preferred over interrupted projections such as MOLLWEIDE HEMISPHERE ⬭ and Goode Homolosine. This is also supported by our participants' feedback where continuous maps are preferred over interrupted ones for STATIC. For example, the preference of STATIC EQUAL EARTH ⬭ was supported by our participants' feedback, as they mentioned *"I didn't feel comfortable when the hemispheres were split into two separate circle portions. I preferred [equal earth] when they were a continuous map surface."*

(P18, Distance-STATIC), and *"the map's edges [in equal earth projection] didn't feel as distorted as other maps such as equirectangular."* (P23, Area-STATIC.) We also identified many positive comments for EQUAL EARTH ⬭, e.g., *"[equal earth projection] gave me a better sense of global spatial reasoning"* (P34, Distance-STATIC), and *"the equal earth projection was the easiest to visualise as it seemed to have a good balance of not being too distorted without having the difficulty of wrap around visualisations that the hemisphere projections had."* (P37, Distance-STATIC).

Surprisingly, although HAMMER ⬭ is an equal-area projection, participants did not like it for area comparison in static maps (Figure 7.5-Area), This result partially differs from existing studies [Šav+15] where poles represented as points were preferred over poles represented as lines. We conjecture this effect is because when the target area is at the edges, the shapes are severely distorted, which makes it difficult to accurately accumulate its represented area, as participants mentioned *"Equirectangular only was distorted from top to bottom, while [hammer was] also distorted on the sides."* (P31, Area-STATIC).

Finally, participants disliked HAMMER ⬭ for STATIC-Area. For example, participants mentioned *"I think [equirectangular] felt easier to read my first choice vs what I rate number 5 [for hammer]."* (P17, Area-STATIC) and *"the hammer too oval to make sense of."* (P19, Direction-STATIC).

In hemispheric projections, interaction reduced ERROR of ORTHOGRAPHIC HEMISPHERE ◯◯ to a point that it tended to have a lower error rate than some interactive continuous projections for Distance-Hard and Direction, and not significantly slower than any interactive projections across all tasks. We found ORTHOGRAPHIC HEMISPHERE ◯◯ performed well. ORTHOGRAPHIC HEMISPHERE ◯◯ benefited more from interaction (large effects) for ERROR than MOLLWEIDE HEMISPHERE ⬭ (moderate effects) for DISTANCE COMPARISON •⟶•, while ORTHOGRAPHIC HEMISPHERE ◯◯ benefited slightly more from interaction with similar effect sizes for ERROR than MOLLWEIDE HEMISPHERE ⬭ for AREA COMPARISON ◆• and DIRECTION ESTIMATION ⟶ (Figure 7.4-ERROR). The effect size can be found: DISTANCE COMPARISON •⟶• (Figure A.2), AREA COMPARISON ◆• (Figure A.3), and DIRECTION ESTIMATION ⟶ (Figure A.4). This is also supported by a strong evidence that INTERACTIVE ORTHOGRAPHIC HEMISPHERE ◯◯ has a lower error rate (large effects) than EQUAL EARTH ⬭ and a lower error rate (moderate effects) than HAMMER ⬭ for Distance-Hard (Figure 7.4-Error). By contrast, even with interaction, MOLLWEIDE HEMISPHERE ⬭ was still slower (small effects) than non-hemisphere for DIRECTION ESTIMATION ⟶ (Figure 7.4-Time). INTERACTIVE ORTHOGRAPHIC HEMISPHERE ◯◯ did not have a significantly lower error rate for AREA COMPARISON ◆• than

any interactive continuous projections, nor was there any significant difference in PREF-INTERACTIVE (Figure 7.6). Therefore, we reject M5.4, M5.5.

It was surprising that ORTHOGRAPHIC HEMISPHERE ◐ was comparable to other interactive projections for AREA COMPARISON ♠♦ since it is not an equal-area map projection. Meanwhile, the other non-equal-area map projection, i.e., EQUI-RECTANGULAR ▭ tended to produce more errors (moderate effects). We believe ORTHOGRAPHIC HEMISPHERE ◐ were perceived as less distorted than the other projections due to the "natural" orthographic distortion, which is similar to viewing the sphere at infinite distance (e.g. as if through a telescope), echoed by our participants. For example, one quoted response of DIRECTION ESTIMATION ↪: *"Orthographic Hemisphere seemed the most accurate to me. The round shape was really useful. The Mollweide Hemisphere was tricky, I didn't like the way the two points had to be on the separate circles for it to be accurate sometimes."* (P4, Direction-INTERACTIVE).

Meanwhile, it seems many people were good at mentally connecting both hemispheres: *"I found it easier to create comparisons that didn't have a lot of distortion with the option of having the two side-by-side globes."* (P18, Direction-INTERACTIVE).

Despite being hemispheric, MOLLWEIDE HEMISPHERE ⬭ was found to be less intuitive for some tasks by participants. We conjecture that there is a slight distortion near the edge of two circles which may make it confusing when centring the region of interest, as participants mentioned *"Orthographic hemisphere felt the easiest to compare because it was like looking at a globe, but the strange way of moving the mollewide hemisphere made it confusing, that is why that is my last option."* (P23, Area-INTERACTIVE), *"The mollweide was a bit difficult to drag and place arrow."* (P23, Direction-INTERACTIVE.).

However, there were also participants who did not like the hemispheric projections due to the need to inspect two separated spheres and instead they preferred the continuous maps in the non-hemispheric group. For example, *"I found it a lot easier to identify on the single maps rather than the double because sometimes they were spaced too far away"* (P11, Area-INTERACTIVE), *"it was hard to picture how Mollweide and Orthographic Hemispheres connected."* (P7, Distance-STATIC), and *"the map layout was easier to see because the targets were all on one map rather than two spheres."* (P36, Distance-INTERACTIVE).

Even with interaction, EQUIRECTANGULAR ▭ still tended to perform poorly in terms of ERROR for AREA COMPARISON ♠♦ and DIRECTION ESTIMATION ↪. Overall, for static projections, EQUIRECTANGULAR ▭ tended to have a higher error rate (moderate effects) than EQUAL EARTH ⬭ and MOLLWEIDE HEMISPHERE ⬭

(Figure 7.4-Error). To our surprise, with the ability to rotate to centre the region of interest, Equirectangular ▢ still tended to be outperformed for Error by Mollweide Hemisphere ◯ (moderate effect) for Area-All and by Orthographic Hemisphere ◯◯ (moderate effect) for Direction estimation ↰ (Figure 7.4-Error). This partially confirms Hennerdal et al.'s static map study where Static Equirectangular ▢ was found confusing when estimating the airplane route that wraps around [Hen15]. We conjecture that Equirectangular ▢ features the most distortion of all tested projections due to the high level of stretching near the poles, supported by participants' feedback. For example, Interactive Equirectangular ▢ was still found to have a greater distortion near the poles than other projections, as participants mentioned *"The sphere is easiest to visualize in my head, I was thinking about just drawing a line on a ball or other spherical shape. The ones with the biggest distortment on the edges confused me most."* (P13, Direction-Interactive) and *"The Mollweide Hemisphere and Orthographic Hemisphere- easier to see a side by side comparison of the areas. The Hammer- less areas of distortion compared to the Equal Earth and Equirectangular. Equal Earth had too much distortion as did the Equirectangular which in my opinion distorted the most."* (P21, Area-Interactive).

7.4 Limitations and Future Work

The statistically significant results with large effect sizes provide strong evidence that adding the spherical rotation interaction to static maps improves the accuracy, is strongly preferred, but at the cost of longer completion time across all tasks. However, despite being statistically significant, the differences between projections within static or interactive groups are of small sizes for Time, medium-sized for Error and large-sized for Pref-Static-Area (Figure 7.4, Figure 7.5). Although medium-sized differences in Error are likely to be noticeable in practical applications, these results only allow us to say Equal Earth ◯ and Orthographic Hemisphere ◯◯ performed well for some tasks [Hel+21; Coc+20; Hum19; SS19].

Although the results show statistically significant differences between the selected hemispheric and non-hemispheric projections across all tasks for Error, the results do not allow us to say that hemispheric projections are always more accurate than non-hemispheric projections. Arguably, the poor performance of non-hemispheric projections for Area Comparison ♠● and Direction estimation ↰ is entirely based on Equirectangular ▢ compared with either Mollweide Hemisphere ◯ or Orthographic Hemisphere ◯◯. If the results from the Equirectangular ▢ were not considered, the hemispheric and non-hemispheric projections seem to

have very similar error rates for AREA COMPARISON ◆◆ and DIRECTION ESTIMATION ↪. Similarly, the results do not allow us to say that non-hemispheric projections are always faster, as this seems only based on MOLLWEIDE HEMISPHERE ⬭ being significantly slower (small effects) than some non-hemispheric projections for STATIC and INTERACTIVE for the direction tasks.

Surprisingly, unlike the study by Yang et al. [Yan+18], we did not identify superior performance of ORTHOGRAPHIC HEMISPHERE ◑ compared to other map projections. We conjectured that rendering ORTHOGRAPHIC HEMISPHERE ◑ on a 2D flat display produced different effectiveness in perception and interaction compared to Yang et al's 3D globes in VR. Meanwhile, although not in all tasks, ORTHOGRAPHIC HEMISPHERE ◑ demonstrated some advantages for DISTANCE COMPARISON •⇢•, DIRECTION ESTIMATION ↪ and benefited more from interaction than MOLLWEIDE HEMISPHERE ⬭ for ERROR (section 7.3).

We also note that in our study design, we experimented with two levels of difficulties for DISTANCE COMPARISON •⇢• and AREA COMPARISON ◆◆, where the sole parameter we vary is the geographic distance between point pairs or pairs of areas. This allows us to evaluate map projection readability with easy as well as more challenging trials, as we believe increasing the geographic distance between two entities will make the task more challenging. However, there are other parameters that could also affect performance that we did not test, such as the geographic distance between individual pair of data points for DISTANCE COMPARISON •⇢•, size of geographic areas for AREA COMPARISON ◆◆, and the angular difference between geographic points for DIRECTION ESTIMATION ↪. We leave testing other geographic comprehension tasks and the other parameters to future work.

While this chapter exclusively focuses on interactive panning, another possible wrapping approach, as presented in section 3.2 is tiled-display, which we detail in subsection 9.4.4.

7.5 Conclusion

The overwhelming finding in the map projection study (section 7.3) confirms the benefits of interactive spherical wrapping for geographic comprehension tasks, addressing **RG5** (subsection 1.2.5).

The finding that interactive spherical wrapping of map projections led to significantly lower error rate and more preferred over standard map projections across all tasks

tested suggests that such interaction should be routinely provided in online world maps, for example in education or in reporting of world events, climate patterns, and so on. While less clear, our results indicated that EQUAL EARTH ⬭ was the best performing continuous projection, while the straight-forward ORTHOGRAPHIC HEMI-SPHERE ◕ was an effective hemispheric projection. Our results also indicate that even with interaction, EQUIRECTANGULAR ▭ may be a poor choice for comparing areas and estimating directions.

Comparing Spherical and Toroidal Wrapping for Network Structured Data

<div style="text-align:right">8</div>

> *Torus is the easiest to interpret because the clusters do not get distorted if the map is moved around. Equal Earth is actually easier than it looks to interpret because even though the clusters get distorted, if they are moved around, they actually bunch up together once they are put near the middle of the map.*
>
> — **Anonymised user study participant, P12**
>
> Cluster identification tasks

Spherical projection has application beyond geographic data. There can be advantages to laying out abstract data (data which has no inherent geometry) on the surface of a sphere such that there is no arbitrary edge to the display or privileged centre [Rod20]. Providing interactive panning has previously been found to be of benefit in understanding network layouts based on torus projection (chapter 6).

We address **RG5**. Inspired by the promising results of spherical wrapping on map projections, in this chapter we investigate the utility of the most successful interactive spherical projections from chapter 7, to compare sphere, torus, and non-wrapped flat layout. We investigate network layouts on the surface of a sphere projected onto a 2D plane. Again we allow the user to interactively pan the projection around the plane.

While the relationship between readability of geographic and network data on spherical projections has not been studied directly, there are commonalities in the analysis tasks that might be applicable for each. For example, understanding network clusters may involve comparing the relative size of their boundaries, similar to map area and shape comparison. Network path following tasks require the user to trace links (AKA 'edges') between nodes in the network while maps also require

| Flat | Torus | Equal Earth | Orthographic Hemisphere |
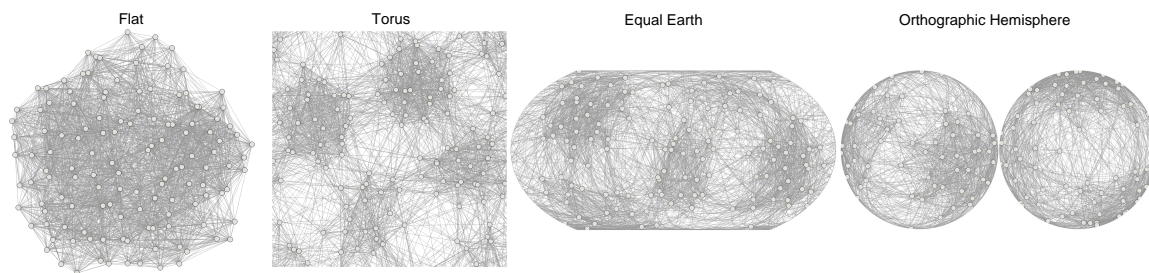
Fig. 8.1.: Study 6 compares the two interactive spherical projections found most effective in Study 5, and compares them to toroidal and standard 'flat' layout for network data.

understanding how regions connect, and in both cases splits or distortion due to spherical projection may present a challenge.

As we discuss in detail in chapter 2, while there has been much work to develop the algorithms for spherical projection of maps and algorithms for layout of networks on a 3D sphere, we find that interactive 2D displays of such spherical layouts of data (whether geographic or network) are not well studied. Also projections of 3D spherical layouts of network data have not been compared to projections of networks arranged on the surface of other 3D geometries, in particular toroidal layouts which have been shown in chapter 6.

8.1 Spherical Network Layout

In Study 5, we found that interaction (panning by spherical rotation) makes spherical geographic projections overwhelmingly more accurate for distance, area and direction tasks. A question, however, is whether such 2D interactive spherical projections are also useful for abstract data. As discussed in subsection 2.1.4, there have been a number of systems using immersive environments to visualise network data on 3D spherical surfaces or straightforward perspective projections of spheres. Various advantages have been claimed to the opportunities for embedding a network layout in the surface of a sphere—without boundary—such as centring any node of interest in the layout and layout of non-privileged centre [BM12; Per+20; Rod20]

Further, there are obvious disadvantages to projection, since all projections introduce some degree of distortion and discontinuity. There are therefore three questions, linked to **RG5** (subsection 1.2.5:

RQ5.1: *Which of the most promising projections from study 5 are the best for visualising the layout of a node-link diagram*?

RQ5.2: *Does a spherical projection have advantages in supporting network understanding tasks compared to conventional 2D layouts?*

RQ5.3: *Does a spherical projection provide perceptual benefits compared with arrangements on other 3D topologies, such as a torus?*

Before we can answer these questions, we need techniques to create effective layouts of complex network data on a spherical surface and to orient the projections optimally in 2D.

8.1.1 Plane, Spherical and Toroidal Stress Minimisation

The tasks we investigate are cluster understanding tasks and path following. Network clusters are loosely defined as subsets of nodes within the graph that are more highly connected to each other than would be expected from chance connection of edges within a graph of that density. More formally, a clustered graph has disjoint sets of nodes with positive *modularity*, a metric due to Newman which directly measures the connectivity of given clusters compared to overall connectivity [New06]. To support cluster understanding tasks we need a layout method which provides good separation between these clusters.

To support path following tasks, we need a layout method which spreads the network out relatively uniformly according to connectivity. This will help minimise crossings between edges.

We follow previous chapters (chapter 5 and chapter 6) which adopted a stress minimising approach. Stress-minimisation is a commonly used variant of a general-purpose force-directed layout and does a reasonable job of satisfying both of these readability criteria [HEH09; Pur02]. The *stress* metric (σ) for a given layout of a graph with n nodes in a 2D plane is defined (following Gansner et al. [GKN04]) as:

$$\sigma_{\text{plane}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij}(\delta_{ij} - d_{ij})^2 \quad , \qquad d_{ij} = |x_i - x_j|$$

where: δ_{ij} is the ideal separation between the 2D positions (x_i and x_j) of a pair of nodes (i, j) taken as the all-pairs shortest path length between them; d_{ij} is the actual distance between nodes i and j (in the plane this is Euclidean distance); and w_{ij} is a weighting which is applied to trade-off between the importance of short and long ideal distances, we follow the standard choice of $w_{ij} = 1/d_{ij}^2$.

We follow previous recent work by Perry et al. [Per+20], in adapting stress-based graph layout to a spherical surface by redefining d_{ij} to arc-length on the sphere surface, or (assuming a unit sphere):

$$\sigma_{\text{sphere}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij}(\delta_{ij} - d_{ij})^2 \quad , \qquad d_{ij} = \arccos(x_i \cdot x_j)$$

where x_i and x_j are the 3D vector offsets of nodes i and j respectively from the sphere centroid and $(\cdot)$ is the inner product. For the layout to be reasonable, the ideal lengths δ must be chosen such that the largest corresponds to the largest separation possible on the unit sphere surface, which is π. Thus, we set the ideal length of all edges to $\pi/graphdiameter$.

The other layout against which we compare is a projection of a 3D torus, which, as discussed in chapter 6, has been shown to provide better separation between clusters than a flat (conventional) 2D layout. We use the same layout method which is also based on stress in the 2D plane but which, for each node pair, requires selecting the stress term from the set A of 9 possible torus adjacencies for that pair which contributes the least to the overall stress (as shown in Figure 5.1):

$$\sigma_{\text{torus}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij}\arg\min_{\alpha \in A}(\delta_{ij} - d_{ij\alpha})^2, d_{ij\alpha} = |x_i - x_{j\alpha}|$$

While Perry et al. follow the multi-dimensional scaling literature in using a *majorization* method to minimise σ_{sphere}, we follow chapter 6 in using the stochastic pairwise gradient descent approach developed by Zheng et al. [ZPG18] which can be adapted straightforwardly and effectively to obtain layout for all of $\sigma_{\text{plane}}, \sigma_{\text{sphere}}, \sigma_{\text{torus}}$.

Note that the layouts which result from minimising σ_{plane} and σ_{torus} are already in 2D. For the spherical layout obtained by minimising σ_{sphere}, we use either ORTHOGRAPHIC HEMISPHERE ◐ or EQUAL EARTH ⬭ projections to generate the stimuli for our study. The detailed pseudocode of our algorithms are available in `https://github.com/Kun-Ting/gansdawrap`.

8.1.2 Auto-pan Algorithms

For toroidal network layout, section 6.2 introduced an algorithm to automatically pan the toroidal layout horizontally and vertically to minimise the number of edges which wrap around at the boundaries. Spherical projections can also suffer when too

(a) ORTHOGRAPHIC HEMISPHERE ⊙⊙ - No pan

(b) ORTHOGRAPHIC HEMISPHERE ⊙⊙ - Best pan

(c) EQUAL EARTH ⬭ - No pan (left), edge pixel mask (inset)

(d) EQUAL EARTH ⬭ - Best pan (left), edge pixel mask (inset)
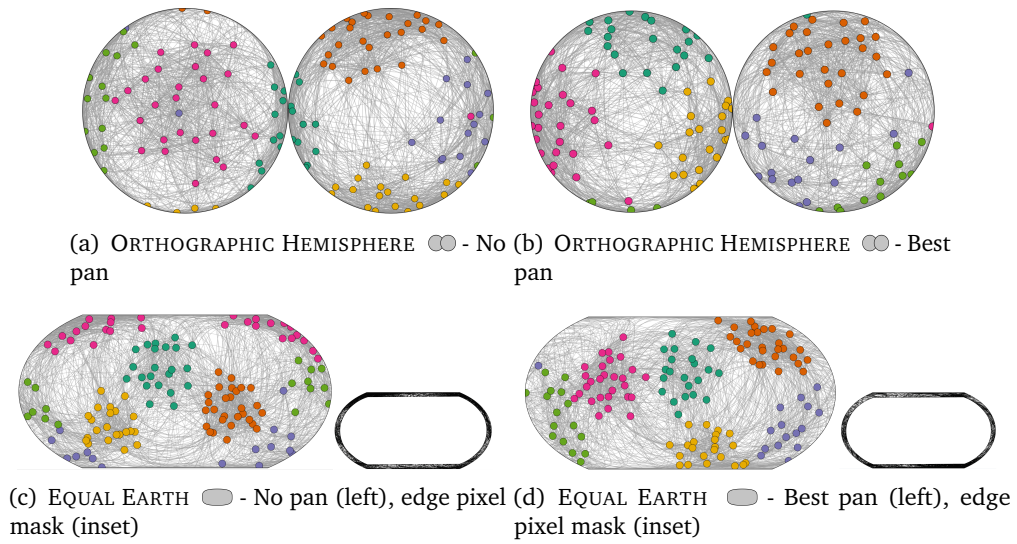
Fig. 8.2.: Before and after demonstration of our auto-pan algorithms of a graph with six clusters differentiated by colour (note: study graphs did not have colour). Without auto-pan, clusters can be split across the hemispheres in ORTHOGRAPHIC HEMISPHERE ⊙⊙ (a) or at the boundaries in EQUAL EARTH ⬭ (c). Auto-pan reduces the number of wrapped edges and thereby brings the clusters together (b) and (d).

many edges are split across the boundaries. Furthermore, edges are more distorted in spherical projections when they are near the edges. Therefore, for fair comparison with toroidal layouts, it was necessary to find a method to auto-rotate the sphere to reduce the numbers of such edges. However, while the toroidal auto-panning algorithm can be done with horizontal and vertical scans (linear time in the number of edges), the spherical layout does not permit such a trivial search algorithm. We therefore develop heuristics to perform auto-rotate for the spherical projections. For both, we choose a simple stochastic method of randomly selecting a large number (e.g., 1000 iterations) of three-axis spherical rotation angle triples (λ, ϕ, γ) and choosing the triple for which edges crossing (or near) boundaries is minimised.

For ORTHOGRAPHIC HEMISPHERE ⊙⊙ projection this crossing number is trivial to count precisely. Simply, for all pairs of nodes if the nodes are not on the same face, then they must cross a boundary. A suboptimal ORTHOGRAPHIC HEMISPHERE ⊙⊙ projection rotation, and the result of autorotation to minimise this crossing count is shown in Figure 8.2(a) and Figure 8.2(b), respectively.

For EQUAL EARTH ⬭ projection the edges are curved and so determining those that cross the boundary for a given geo-rotation is more complex. Further, in this projection edges near the periphery are significantly more distorted than those near the centre, so even visually determining if an edge that comes close to the boundary

continues on the same side of the projection or wraps around to the other side is not easy. Thus, instead of counting boundary crossings, we penalise all edges which come close to the boundaries. To compute the penalty we analyse the periphery of a monochrome bitmap of the projected edge paths. The penalty is then simply the number of black pixels. Example masked bitmaps are shown for sub-optimal and more-optimal rotations of an EQUAL EARTH ⬭ projection in Figure 8.2(c) and Figure 8.2(d), respectively.

8.1.3 Automatic Panning Results

We conducted a small empirical analysis of the Auto-Pan algorithm to assess the numbers of links wrapped for ORTHOGRAPHIC HEMISPHERE ⬤⬤ and the number of pixels (higher values indicate less wrappings) at the boundary of EQUAL EARTH ⬭ projections compared to 10 random rotations. Across 10 study graphs (SMALL+EASY, SMALL+HARD) used in our cluster understanding tasks, we found for ORTHOGRAPHIC HEMISPHERE ⬤⬤, the mean crossing count for random rotations was 262.16. With automatic panning, this number was improved by 25.6% and was reduced to 208.7. For EQUAL EARTH ⬭, our automatic panning increased the number of pixels at the boundary region by 12.1% (Section A.2: Table 2). We found these auto-pan algorithms resulted in a noticeable improvement in keeping clusters from being separated, as evidenced in Figure 8.2.

8.2 User Study 6: Spherical Network Projection Readability

In Study 5, we found EQUAL EARTH ⬭, had advantages in terms of error rate, time, and subjective user feedback. ORTHOGRAPHIC HEMISPHERE ⬤⬤, despite being an interrupted projection, benefited more from interaction than MOLLWEIDE HEMISPHERE ⬭ and performed well in terms of error rate for distance and direction tasks (section 7.3). Based on these findings, we chose interactive EQUAL EARTH ⬭ and ORTHOGRAPHIC HEMISPHERE ⬤⬤, to understand their performance in visualising networks. We compare these spherical projections to standard flat graph layout (FLAT ✧) and a projection of a TORUS ⬯ geometry (chapter 6). Our study was run in a similar way to Study 5 and included a new set of 96 participants through the Prolific platform.
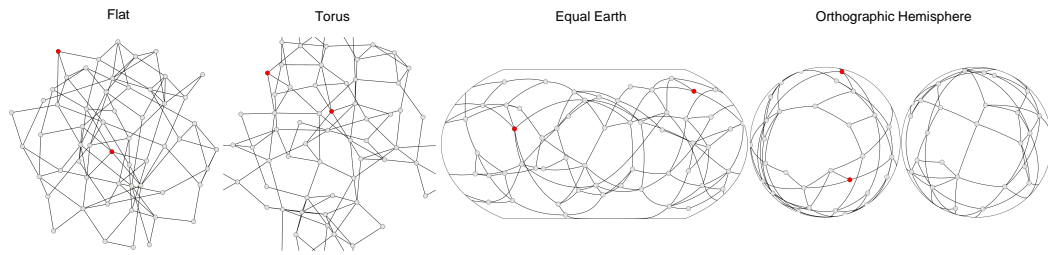
Fig. 8.3.: Example of one study graph laid out using 4 techniques described in subsection 8.1.1 for SHORTEST PATH NUMBER. The shortest path length is 3 in this example. Participants were provided with interactive panning for spherical and toroidal layouts to explore the network.

8.2.1 Techniques & Setup

The techniques in our study are FLAT ✧, TORUS ⬭, EQUAL EARTH ⬭, and ORTHOGRAPHIC HEMISPHERE ⬤⬤. Layouts are computed as described in section 8.1. All the techniques support interactive panning except for FLAT ✧ which does not wrap around. The area of the rectangular bounding box of each technique condition is the same. For FLAT ✧ and TORUS ⬭, the resolution is fixed at 650×650 pixels, and for EQUAL EARTH ⬭ and ORTHOGRAPHIC HEMISPHERE ⬤⬤, the resolution is fixed at 900×317 pixels. For CLUSTER NUMBER ⬭, we did not color clusters to not reveal any graph structure.

8.2.2 Tasks

Our study is intended to be complementary to the studies described in section 5.6 and section 6.4 by evaluating a new geometry sphere with important tasks considered in those studies. Inspired by those studies, we selected two representative network visualisation tasks comparing 2D layout of 3D surface topology.

CLUSTER NUMBER ⬭**:** *Please count the number of clusters in this graph.* Participants answered through radio buttons: choices ranged from 1 to 10. We created a quality control trial for each condition with two clusters with clearly marked boundaries to assess participants' attention. Participants who did not answer these trials correctly were excluded. An example of 5 clusters from LARGE + EASY graphs (subsection 8.2.3) using 4 different layouts is shown in Figure 8.1-Bottom.

SHORTEST PATH NUMBER ∿**:** *What is the shortest path length between the red nodes?*: Participants were required to count the smallest number of links between two red nodes. Radio buttons allowed them to answer between 1 to 6. Again, we created

a quality control trial for each layout condition, with shortest path length two and links on the path highlighted in red. An example of EASY graphs (subsection 8.2.3) using 4 different layouts is shown in Figure 8.3.

8.2.3 Data Sets

We prepared a separate graph corpus for each task, full details and stimuli are presented in our supplementary material. For CLUSTER NUMBER ⁂, we use graphs from subsection 6.3.1, generated using algorithms designed to simulate real-world community structures in graphs [BGW03; For10]. Graphs are grouped by two variables: difficulty in terms of graph modularity [New06] (EASY: modularity=0.4, HARD: modularity=0.3) and size (2 levels: SMALL: 68-80 nodes, 710-925 links, and LARGE: 126-134 nodes, 2310-2590 links). The number of clusters is between 4 and 7. For SHORTEST PATH NUMBER ∿ the clustered graphs were too dense, so we generated sparser graphs using scale-free models [BA99; WS98]. We chose graphs with two levels of density (EASY: 0.075, and HARD: 0.11) with 50 to 57 nodes. The shortest path length varied between 1 and 4.

8.2.4 Hypotheses

Hypotheses were pre-registered with the Open Science Foundation: `https://osf.io/equhp`.

G5.1: EQUAL EARTH ⬭ *and* TORUS ⬯ *have better task effectiveness for* CLUSTER NUMBER ⁂ *(in terms of time and error) than* ORTHOGRAPHIC HEMISPHERE ◉◉ (RQ5.1, RQ5.3) *or* FLAT ⇲ (RQ5.2). While ORTHOGRAPHIC HEMISPHERE ◉◉ performed well in Study 5, our pilot studies for CLUSTER NUMBER ⁂ revealed that cuts and distortion of clusters at the borders made them hard to count. For (RQ5.2), the inspiration was based on our prior cluster readability studies in section 6.4.

G5.2: *Participants will prefer* EQUAL EARTH ⬭ *and* TORUS ⬯ *to* ORTHOGRAPHIC HEMISPHERE ◉◉ (RQ5.1, RQ5.3) *or* FLAT ⇲ (RQ5.2) *for* CLUSTER NUMBER ⁂. Our early pilots indicated this preference — perhaps for the same reasons as above and inspiration from our prior studies section 6.4.

G5.3: TORUS ⬯ *has better task effectiveness (in terms of time and error) for* SHORTEST PATH NUMBER ∿ *than* FLAT ⇲ EQUAL EARTH ⬭ *or* ORTHOGRAPHIC HEMISPHERE ◉◉ (RQ5.3). This assumption is based on pilot studies and prior studies indicating curved links might hamper path tracing tasks [Du+17; Xu+12].

G5.4: *Participants will prefer* TORUS ⬭ *to* FLAT ✦, EQUAL EARTH ⬭ *or* ORTHO-GRAPHIC HEMISPHERE ◉ (RQ5.3) *for* SHORTEST PATH NUMBER ∿. Again, it was assumed due to distortion of links.

8.2.5 Experimental Design

We use a within-subjects design for each task with 4 techniques. Each participant was randomly assigned one of the tasks by the experimental software. For CLUSTER NUMBER ⁑, we used 2 levels of difficulty (Easy, Hard) × 2 sizes (Small, Large) × 5 repetitions. We randomly inserted one additional quality control trial to each layout condition. Each recorded trial had a timeout of 20 seconds to prevent participants from trying to perform precise link counting. This leaves us with a total of 80 recorded trials per participant. We counterbalanced the order of the techniques using a full-factorial design. The order of each level of difficulty and size in each technique was the same: SMALL+EASY→LARGE+EASY→ SMALL+HARD→LARGE+HARD. The order of trials for each technique within each level was randomised.

For SHORTEST PATH NUMBER ∿, we used 2 levels of difficulty (Easy, Hard) × 8 repetitions. There were 2 repetitions of each shortest path length per level. One additional quality control trial was added for each layout condition. This leaves us with a total of 64 trials per participant. Each recorded trial had a timeout of 30 seconds, informed by pilot studies. The order of each level of difficulty in each technique was the same: EASY→ HARD. The order of trials for each technique within each level was randomised.

8.2.6 Participants and Procedures

Setup and inclusion criteria for participants were the same as for Study 5. We recorded 96 participants (37 female, 59 male, age range [18,50]) who passed the attention check trials, completed the training and recorded trials. This comprised 2 fully counterbalanced blocks of participants (24 × 2 × 2 tasks). They ranked their familiarity with network diagrams as: 9 often seeing network diagrams; 62 occasionally; and 25 never.

For CLUSTER NUMBER ⁑, each participant was presented with a tutorial explaining the concept of network clusters. For TORUS ⬭ EQUAL EARTH ⬭, and OR-THOGRAPHIC HEMISPHERE ◉, animated videos were given to demonstrate the interactive panning or rotation. Each participant then completed training trials,
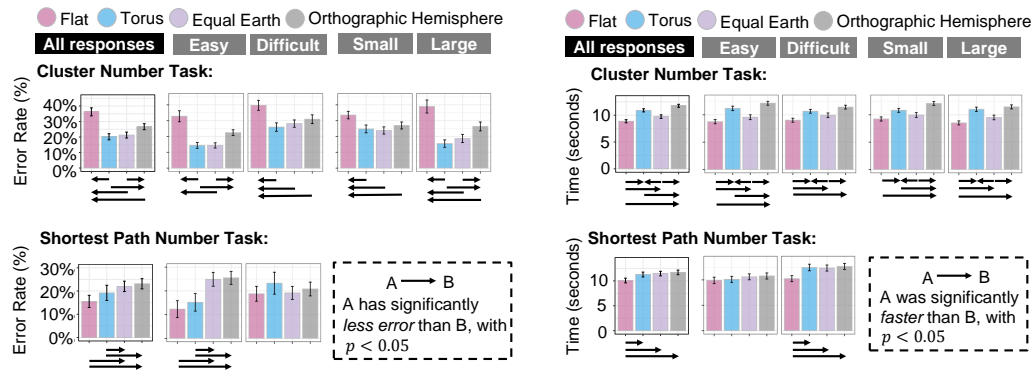
Fig. 8.4.: Error rate (left) and Time (right) results of Study 6. Error bars are 95% confidence intervals.

Significant differences between projections are shown as arrows. Detailed statistical results are available in Figure A.6. Effect sizes are available in Figure A.7.

similar to the recorded trials. For SHORTEST PATH NUMBER $\curlyvee$, each participant was presented with a tutorial explaining the concept. For TORUS ⬭, EQUAL EARTH ⬭, ORTHOGRAPHIC HEMISPHERE ⬬, animated videos were given to demonstrate the interactive panning or rotation. Training trials were similar to the recorded trials.

Specific instructions given for each task is available in the supplementary material[1].

8.2.7 Dependent Variable and Statistical Analysis Methods

We recorded task-completion time (TIME), task-error (ERROR), and subjective *confidence* as PREF (the smaller the more confident). We calculated ERROR as the normalised absolute difference between the correct answer and response. We used the same statistical analysis methods and standardised effect sizes from the first study (subsection 7.2.7).

8.2.8 Results

In the following, significance values are reported for $p < .05(*)$, $p < .01(**)$, and $p < .001(***)$, respectively, abbreviated by the number of stars in parenthesis.

For CLUSTER NUMBER, we found Layout had a significant effect on error rate in All Responses $(***)$, Easy $(***)$, Hard $(***)$, Small $(***)$, and Large $(***)$.

---

[1]An online demonstration of the study is available: `https://observablehq.com/@kun-ting/gansdawrap`
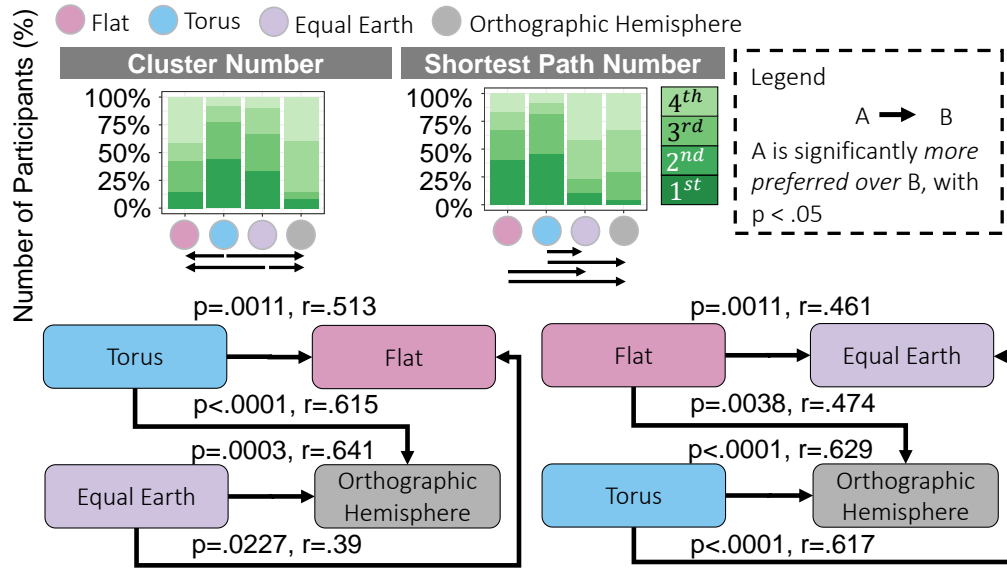
Fig. 8.5.: User confidence ranking of each condition for two tested tasks. Arrows indicate statistical significance with $p < 0.05$. Effect size results of Cohen's r [Coh13] are presented along with the arrows.

TORUS ⊖ (∗ ∗ ∗), EQUAL EARTH ⬭ (∗ ∗ ∗) and ORTHOGRAPHIC HEMISPHERE ◍ (∗) were more accurate than FLAT ✫. TORUS ⊖ (∗) and EQUAL EARTH ⬭ (∗∗) were more accurate than ORTHOGRAPHIC HEMISPHERE ◍. We also found TORUS ⊖ and EQUAL EARTH ⬭ were more accurate than FLAT ✫ for Easy (∗ ∗ ∗) and ORTHOGRAPHIC HEMISPHERE ◍ for Easy (∗ ∗ ∗). We found TORUS ⊖, EQUAL EARTH ⬭, and ORTHOGRAPHIC HEMISPHERE ◍ were more accurate than FLAT ✫ for Hard (∗ ∗ ∗), Small (∗ ∗ ∗). We also found TORUS ⊖ (∗ ∗ ∗), EQUAL EARTH ⬭ (∗ ∗ ∗) and ORTHOGRAPHIC HEMISPHERE ◍ (∗) were more accurate than FLAT ✫ for Large. TORUS ⊖ (∗) and EQUAL EARTH ⬭ (∗∗) were more accurate than ORTHOGRAPHIC HEMISPHERE ◍ for Large.

We found Layout (∗ ∗ ∗), Layout × Difficulty (∗ ∗ ∗) and Layout × Size (∗) had a significant effect on time. In Layout, FLAT ✫ is faster than TORUS ⊖, EQUAL EARTH ⬭ and ORTHOGRAPHIC HEMISPHERE ◍ (Layout ∗ ∗ ∗). TORUS ⊖ was found faster than ORTHOGRAPHIC HEMISPHERE ◍ (Layout ∗ ∗ ∗). EQUAL EARTH ⬭ was faster than TORUS ⊖ and ORTHOGRAPHIC HEMISPHERE ◍ (Layout ∗ ∗ ∗). In Layout × Difficulty, FLAT ✫ is faster than TORUS ⊖ (Easy/Hard ∗ ∗ ∗), EQUAL EARTH ⬭ (Easy ∗, Hard ∗ ∗ ∗), and ORTHOGRAPHIC HEMISPHERE ◍ (Easy/Hard ∗ ∗ ∗). EQUAL EARTH ⬭ is faster than Torus (Easy ∗ ∗ ∗, Hard ∗) and ORTHOGRAPHIC HEMISPHERE ◍ (Easy/Hard ∗ ∗ ∗). We also found TORUS ⊖ was faster than ORTHOGRAPHIC HEMISPHERE ◍ (Easy ∗). In Layout × Size, FLAT ✫ was faster than TORUS ⊖ (Small/Large ∗ ∗ ∗), EQUAL EARTH ⬭ (Large ∗ ∗ ∗), and ORTHOGRAPHIC HEMISPHERE

⬭⬭ (Small/Large ∗ ∗ ∗). Equal Earth ⬭ was found faster than Torus ⊖ (Small ∗∗, Large ∗ ∗ ∗), and Orthographic Hemisphere ⬭⬭ (Small/Large ∗ ∗ ∗). Torus ⊖ was also found faster than Orthographic Hemisphere ⬭⬭ (Small ∗ ∗ ∗).

For user ranking, participants preferred Torus ⊖ to Flat ✧ (∗∗) and Orthographic Hemisphere ⬭⬭ (∗ ∗ ∗). Participants preferred Equal Earth ⬭ to Flat ✧ (∗) and Orthographic Hemisphere ⬭⬭ (∗ ∗ ∗).

For ShortestPath, we found Layout had a significant effect on error rate in all (∗ ∗ ∗). We found for All Responses, Flat ✧ was more accurate than Equal Earth ⬭ (∗∗) and Orthographic Hemisphere ⬭⬭ (∗ ∗ ∗). Torus ⊖ was more accurate than Equal Earth ⬭ (∗∗) and Orthographic Hemisphere ⬭⬭ (∗∗). In Easy, we found Flat ✧ was more accurate than Equal Earth ⬭ (∗ ∗ ∗) for Easy, and Orthographic Hemisphere ⬭⬭ (∗ ∗ ∗) for Easy. Torus ⊖ was also more accurate than Equal Earth ⬭ (∗ ∗ ∗) for Easy and Orthographic Hemisphere ⬭⬭ (∗∗) for Easy. We did not find a significant difference in Hard.

We found Layout had a significant effect on time (∗ ∗ ∗). Flat ✧ was faster than Torus ⊖, Equal Earth ⬭ and Orthographic Hemisphere ⬭⬭ (∗ ∗ ∗). In Hard, we also found Flat ✧ was faster than Torus ⊖, Equal Earth ⬭ and Orthographic Hemisphere ⬭⬭ (∗ ∗ ∗). We did not find a significant difference in Easy. For user ranking, participants preferred Flat ✧ to Equal Earth ⬭ (∗∗) and Orthographic Hemisphere ⬭⬭ (∗∗). Torus ⊖ was also found preferred to Equal Earth ⬭ (∗ ∗ ∗) and Orthographic Hemisphere ⬭⬭ (∗ ∗ ∗).

8.3 Key Findings and Discussion

We report on the most significant findings for Cluster Number •:, Shortest Path Number ∿ visually in Figure 8.4 and Figure 8.5.

For Cluster Number •:, Equal Earth ⬭ and Torus ⊖ performed equally well and Equal Earth ⬭ was slightly faster than Torus ⊖, while they both significantly outperformed Orthographic Hemisphere ⬭⬭ in terms of Error, Time, and Pref. The poor performance of Orthographic Hemisphere ⬭⬭ compared with Equal Earth ⬭ and Torus ⊖ was found statistically significant with moderate and large effect sizes for Error (All, Easy, Large), Time (Easy, Large), and large effects for Pref (Figure 8.4-top and Figure 8.5). The statistics graphics of significant statistical results of Error and Time are shown in Figure A.6. Effect sizes of Error and Time results are shown in Figure A.7. Effect sizes of

PREF results are shown in Figure A.8. These results confirm **G5.1, G5.2** for (RQ5.1, RQ5.3). Surprisingly, while results from Study 5 (section 7.3) showed that interactive ORTHOGRAPHIC HEMISPHERE ◐ tended to be more accurate than EQUAL EARTH ⬭ for distance comparisons and not worse than any other projections for area comparisons, it turned out to be significantly worse than EQUAL EARTH ⬭ and TORUS ⬯ for reading CLUSTER NUMBER ⁘.

While it has the advantage of being a straightforward mapping from the sphere, ORTHOGRAPHIC HEMISPHERE ◐ is an interrupted and non-area-preserving projection. Therefore, we conjecture that this discontinuity caused participants to struggle to make out cluster boundaries as compared with continuous representations in TORUS ⬯ and EQUAL EARTH ⬭. Furthermore, there is no link distortion in TORUS ⬯. This was supported by participants' feedback, e.g., *"The 'Equal Earth' method felt much easier to distinguish every [individual] cluster."* (P20), and *"Orthographic Hemisphere uses 2 maps so it is much more difficult to interpret than the others [...] making it harder to isolate clusters."* (P12). Furthermore, EQUAL EARTH ⬭ and TORUS ⬯ were found to be easier to make out cluster boundaries, while the separated maps in ORTHOGRAPHIC HEMISPHERE ◐ made it confusing, as participants mentioned *"It was easier seeing the boundaries of the set number of clusters in [torus] representation."* (P11), and *"I found understanding the movement of the orthographic diagram quite challenging."* (P4).

Some participants explicitly mentioned that interaction improves the readability of clusters in TORUS ⬯ and EQUAL EARTH ⬭, while ORTHOGRAPHIC HEMISPHERE ◐ is confusing although it looks naturalistic, e.g., *"Torus is the easiest to interpret because the clusters do not get distorted if the map is moved around. Equal Earth is actually easier than it looks to interpret because even though the clusters get distorted, if they are moved around, they actually bunch up together once they are put near the middle of the map. Orthographic Hemisphere uses 2 maps so it is much more difficult to interpret than the others due to this making it harder to isolate clusters."* (P12), *"Second place is Torus because you can very easily navigate. Third place is equal earth because you can see almost all of the clusters at the same time, but it's a bit confusing because of distortion. Last place is Ortographic Hemisphere because it is very confusing"* (P16), and *"Orthographic was the most convenient one due to familiar shape of earth."* (P6).

We also note that although being statistically significant, the small effect sizes indicate that EQUAL EARTH ⬭ is slightly faster than TORUS ⬯ [Hel+21; Coc+20].

EQUAL EARTH ⬭, ORTHOGRAPHIC HEMISPHERE ◐ and TORUS ⬯ significantly resulted in less error than FLAT ✧. EQUAL EARTH ⬭ and TORUS ⬯ were signif-

icantly preferred over FLAT ✧ **but** TORUS ⬭ **and** ORTHOGRAPHIC HEMISPHERE ⬮ **took longer time than** FLAT ✧. This was found statistically significant with moderate and large effect sizes (Figure 8.4-top, Figure 8.5, Section A.2: Figure 15, Figure 16, and Figure 17), leading us to reject **G5.1** for (RQ5.2) and confirm **G5.2** for (RQ5.2). These results provide strong evidence that with automatic panning and interaction, participants were able to better identify the high-level network structures using spherical projections. They also confirm the results presented in section 6.4 where toroidal layouts with automatic panning significantly outperformed FLAT ✧ in terms of error for cluster understanding tasks. Some participants explicitly mentioned that good separation of clusters in continuous surfaces such as EQUAL EARTH ⬭ and TORUS ⬭ helped understanding.

For example, some participants explicitly mentioned that good separation of clusters in continuous surfaces such as TORUS ⬭ and EQUAL EARTH ⬭ helped understanding, e.g., *"Distance between the clusters as well as perspective played a huge role in my choices. The more further apart the clusters were the easier it was for me to count them."* (P18), *"I think "equal earth" was the easiest and most understandable."Node-link" was something like a challenge for me,it was very difficult to understand it."* (P28), *"I could see patterns more easily in the torus representation. It was easier seeing the boundaries of the set number of clusters in that representation."* (P11), and *"Torus seems to be the easiest, because it was just flat surface with possibility of dragging."* (P14), while FLAT ✧ tended to appear tangled, e.g., *"With the [*FLAT ✧*], it was sometimes apparent where the clusters were, but I feel like I chose 1 cluster as an answer too frequently with the tightly packed examples."* (P11).

Participants also mentioned the automatic panning helped them "see" without the need to interact e.g., *"[with* EQUAL EARTH ⬭*,] even though if we move the picture [it] is very hard to understand the clusters. [It] turned out to be easiest to find the clusters (by not moving the picture)."* (P37).

For SHORTEST PATH NUMBER ⌁, FLAT ✧ **and** TORUS ⬭ **performed equally well**, both significantly outperforming EQUAL EARTH ⬭ and ORTHOGRAPHIC HEMISPHERE ⬮ in terms of ERROR and PREF, while FLAT ✧ is faster than all representations. We found FLAT ✧ and TORUS ⬭ have a lower error rate than EQUAL EARTH ⬭ and ORTHOGRAPHIC HEMISPHERE ⬮ for all responses (moderate effects) and Easy (large effects) (Figure 8.4-bottom). The statistics graphics of significant statistical results of ERROR and TIME are shown in Figure A.6. Effect sizes of ERROR and TIME results are shown in Figure A.7. There is a strong evidence that TORUS ⬭ is more preferred (large effects) over EQUAL EARTH ⬭ and ORTHOGRAPHIC HEMISPHERE ⬮, while FLAT ✧ is more preferred (moderate effects) over EQUAL EARTH ⬭ and

ORTHOGRAPHIC HEMISPHERE ⊙⊙ (Figure 8.5). Effect sizes of PREF results are shown in Figure A.8. FLAT ✧ was the fastest for All Responses (small and moderate effects) and Hard (moderate and large effects). We therefore accept **G5.3, G5.4** for (RQ5.3) but reject **G5.3, G5.4** for FLAT ✧.

We conjecture that although TORUS ⬭ involves broken links across the boundaries, it appears similar to FLAT ✧ using straight links while the distortion of paths in EQUAL EARTH ⬭ might hamper path tracing tasks, as participants mentioned *"Flat surface is easier to read, it helps sometimes when you can also move it. Earth-like is just hard to use, especially when it's an equal globe."* (P9), and *"the warping in [*EQUAL EARTH ⬭*] made my eyes hurt a little therefore its in 4th place, and the torus being the most straight forward gets 1st with [*FLAT ✧*] in second as they're very similar."* (P13). On the other hand, ORTHOGRAPHIC HEMISPHERE ⊙⊙ has less link distortion but the interruption between two hemispheres and the strong perspective distortion near the boundary of the maps may make participants confused, supported by participants' feedback.

Although TORUS ⬭ involves broken links across the boundaries, it appears similar to FLAT ✧ using straight links while the distortion of paths in EQUAL EARTH ⬭ might hamper path tracing tasks: *"I am more familiar with the [*FLAT ✧*] and torus shaped representations."* (P6), and *"The flatter that it seemed, the easier it was."* (P20).

On the other hand, ORTHOGRAPHIC HEMISPHERE ⊙⊙ has less link distortion but the interruption between two hemispheres and the strong perspective distortion near the boundary of the maps may make participants confused. For example, *"Orthographic Hemisphere was even trickier because I couldn't get some of the dots on the same sphere sometimes and the time ran out."* *(P12)* and *"Orthographic Hemisphere is the last because I found it the most restrictive, since it is difficult to make the red points appear together since the curves of the links don't bend, making it harder to see all the ways they connect."* (P21).

Although automatic panning (subsection 8.1.2) provides some benefits for reducing split of clusters across the boundaries for CLUSTER NUMBER •፥, it seems it has less benefits for SHORTEST PATH NUMBER ∿, as participants mentioned they still need to position the image to see the full path to identify the shortest path connecting the nodes within the time limit for SHORTESTPATH. For example, *"Equal earth and orthographic hemisphere would make things more puzzling for me. When I had to wrap the map in order to find the fastest path for the nodes, it would usually take me more time to estimate and find the lines between the two red nodes. I would say that when curves are being shaped it is more complex to find the fastest path. It also made me*

anxious because I had to answer nearly at the last moment. Torus was almost as the simple node-link diagram. However, I found it easier to find the links between the nodes in a diagram that I had little or no interaction." (P35) and *"I was more confident using the Node-Link. It was simpler to use visually. With the other three, there were times when it was difficult to position the map in a way that I could identify the shortest path for connecting the nodes."* (P45).

Overall, these findings suggest that TORUS ⬭ presents a general solution being not only less error prone than FLAT ✧ or ORTHOGRAPHIC HEMISPHERE ⬤⬤ for CLUSTER NUMBER •⁚, but also comparable to FLAT ✧ for SHORTEST PATH NUMBER ∿.

8.4 Limitations and Future Work

A limitation of our study is that we only tested with one layout algorithm. While many different algorithms exist for FLAT ✧ and it is possible that other layout algorithms could be adapted to spherical and torus embeddings, doing so is not necessarily trivial. Also, it should be noted that there are algorithms which can optimise layout for a known set of clusters (i.e. where the cluster labelling is known in advance) [Meu+13; CPC09; GHK09; HGK10]. However, for the tasks tested in this paper we do not pre-identify the clusters but rather leave cluster identification as the user task.

8.5 Conclusion

The study results presented in this chapter confirm the benefits of topologically closed surfaces, such as the surface of a torus or sphere, when using node-link diagrams to investigate network structure. All of EQUAL EARTH ⬭, ORTHOGRAPHIC HEMISPHERE ⬤⬤, and TORUS ⬭ outperformed FLAT ✧ for cluster understanding tasks. While the spherical projections impeded path following tasks, it seems TORUS ⬭ may be a good general solution, being as accurate as FLAT ✧ for path following. Although they have been explored in research, 2D projections of toroidal and spherical network layouts are rarely seen in practice. This may be because until recently effective layout and projection methods for such geometries were not easily available. We intend to make all our algorithms and extensions of existing open-source tools available for easy consumption in web applications via GitHub and *npm* packages.

A limitation of our work is that there are many more possible spherical map projections than those evaluated here, although we tried to select the most representative techniques. Further work may extend such evaluation to other projection types. Another interesting topic for future investigation would be investigating spherical and torus projections of other types of abstract data representation, such as multi-dimensional scaling techniques of high-dimensional data. Another family of interactive techniques for exploring graphs involve applying spatial distortion around regions of interest to achieve a kind of "structure aware zooming", e.g. [Wan+18]. It would be interesting to compare the efficacy of these techniques against interactive sphere and torus projections.

Discussion, Future Work and Conclusion

<div style="text-align:right">9</div>

> *We shouldn't abbreviate the truth but rather get a new method of presentation.*
>
> — **Edward Tufte**
> (American statistician and emeritus professor
> of political science, statistics, and computer
> science at Yale University.)

In this final chapter we first summarise the primary research contributions described in the previous chapters (section 9.1). We then reflect on the utility of wrapped visualisation with regard to some practical examples of network visualisations (section 9.3) and consider other possibilities for future work (section 9.4).

9.1 Contributions

In this thesis, we argued that interactive wrapped visualisations provide perceptual benefits compared to standard visualisation techniques for presentation of a range of common data types that are "not flat", in the sense that they can not be easily presented on a 2D screen or surface without loss of information (as described in chapter 1). Examples of such data types that we have considered include cyclical data, geographic maps and networks of relational data. Figure 9.1 illustrates example visualisations of these data types that result in extremely promising task performance (in terms of accuracy) or graph layout aesthetics when we employ the concept of wrapped visualisation.

To investigate this new class of interactive wrapped visualisation and its effectiveness, we presented the first wrapped visualisation design space that unifies the notion of embedding data on 3D cylinder, sphere and torus topologies in order to create wrapped 2D projections. Our design space encompasses sophisticated panning interactions with the 2D projections and suggests novel applications for
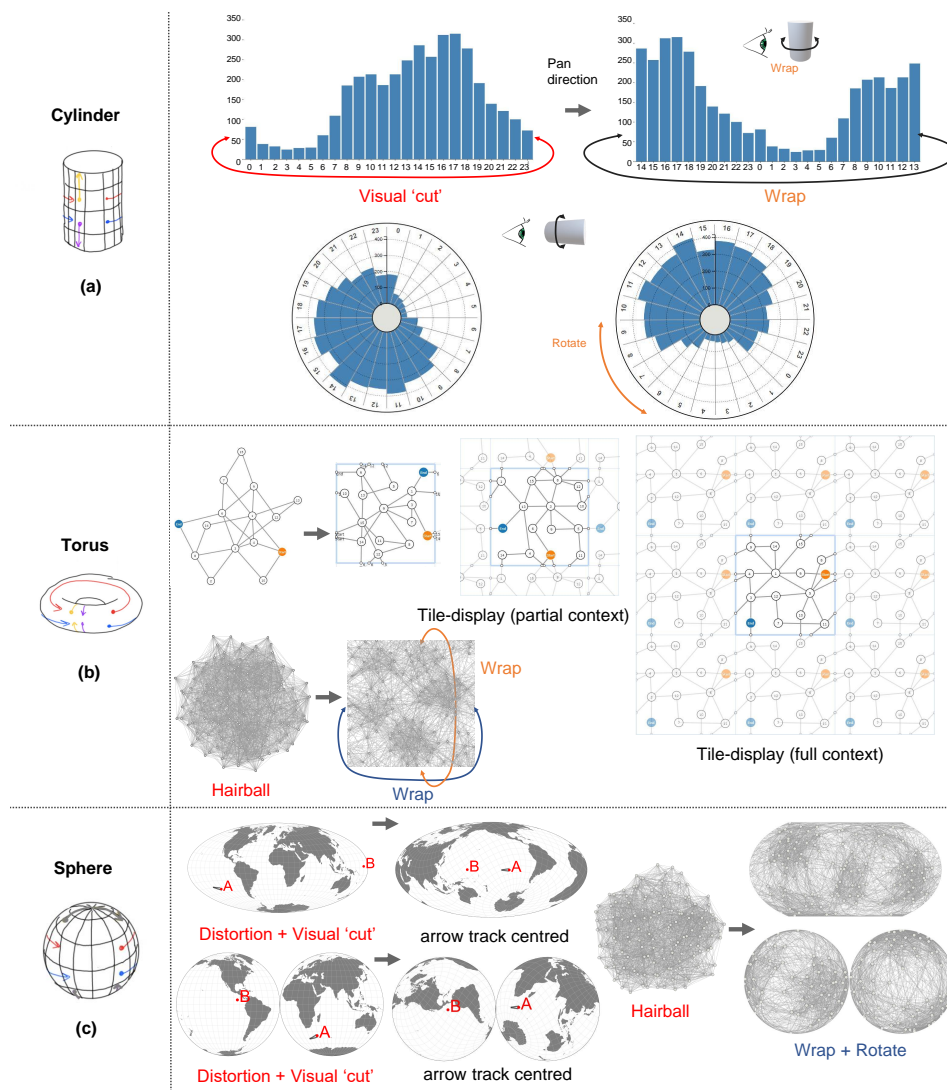
Fig. 9.1.: Summary of a common range of non-flat data types that could benefit from being laid out on a 2D wraparound topology.

multidimensional scaling plots and matrices (chapter 3). We have also created interactive web tools and network layout algorithms that realise the ideas from our design space to provide practical visualisations. The tools and layout algorithms are open-sourced to assist future research and development.

Through a series of six user studies with more than 350 participants investigating cyclical time series, networks, and geographic maps, our evaluation results show significant benefits for a number of tasks and visualisation types when compared to their standard unwrapped counterparts in task performance (specifically accuracy, but sometimes at the cost of completion time) and also in terms of user preference.

Specifically, we first (a) explored visual representation of real-world cyclical time series data presented in bar charts and concentric radial (or "polar") charts that can be mapped to a cylindrical topology and explored with novel cylindrical wrapping (which allows the mark to be panned left or right to connect the visual cut split across the left-and-right boundary at the centre) or rotation interaction (a technique that allows for rotating and centring bars for a more aligned comparison). Our results provide the first empirical evidence that interactive wrapping of cyclical time series in linear bar charts provides promising results in terms of accuracy over standard bar charts without such interaction for trend identification, while interactive wrapping of bar charts outperformed standard bar charts, polar charts, and rotatable polar charts in terms of accuracy for pairwise value comparisons (chapter 4).

Next, (b) we applied and evaluated ideas of torus wrapping, originally developed in theoretical graph drawing research [MT01; KK16], about embeddings of graphs on surfaces with topologies that allow for better "unfolding". With a graph corpus of 72 random networks (up to 15 nodes, 36 links) with scale-free or small-world properties (defined in subsection 2.1.1), we showed that torus-based layouts created by our layout algorithm afford better aesthetics than standard non-wrapped force-directed layouts (chapter 5) in terms of conventional measures like more equal edge length, lower "stress" (defined in subsection 2.1.8, less deviation of adjacent incident link (edge) angles from the ideal minimum angle (determined by a complete angle divided by the number of links connected to the node) between edges connecting to the same node, and fewer crossings. Using these layouts, we showed in two controlled user studies that torus layout with either additional context (tile repetition that shows network connectivity at the sides of the diagram, as seen in Figure 9.1(b)) or interactive panning provided significant performance improvement (in terms of error and time) over torus layout without either of these improvements, to the point that it is comparable to standard flat node-link layout, confirming benefits of both

wrapping approaches for torus described in our design space (section 3.1–wrapping dimension).

To evaluate realistically larger networks, we further refined our torus-based layout algorithm that better displays high-level network structures like clusters (subsets of nodes that are more tightly connected within the subset than outside the subsets). Our extensive experimental analysis with a large graph corpus (200 random graphs with community structures, up to 128 nodes and 2590 links, described in chapter 6) shows that our new toroidal layout algorithm is considerably more robust, consistently producing high-quality layouts in terms of improved aesthetics than standard non-wrapped force-directed node-link layouts and our previous torus layouts (chapter 5) including less stress, fewer link crossings, and greater cluster separation. Through our various design prototypes and studies we also established that wrapped visualisations (e.g., torus or sphere wrapping) benefit from an automatic method for choosing an initial centre (a technique we called "autopanning"). In chapter 6 and chapter 8 we presented new techniques for automatic panning of spherical and torus network visualisations. Our user evaluation showed that interactive torus wrapping with automatic pan improve task results in error by 62.7% and time by 32.3% (differences between means) over standard flat layouts for larger networks for cluster understanding tasks, and that it led to reduced panning interaction by users in some tasks for torus (chapter 6).

Finally, (c) we investigated another wrapping geometry, sphere, which has been widely considered for maps and networks but mostly on a static 2D plane (i.e. without support for interactive panning). We demonstrated how interactive spherical wrapping and rotation of geographic maps provides more accurate area, distance, and direction estimations (chapter 7). We also showed that interactive spherical wrapping and rotation of networks provide benefits over standard non-wrapped force-directed node-link layouts for cluster identification tasks, while the spherical wrapping view impedes path following tasks. Meanwhile, wrapped torus-based network visualisations work well overall, specifically better than non-wrapped force-directed node-link and interrupted spherical projects for cluster identification and being comparable to non-wrapped flat node-link for path following tasks (chapter 8).

The investigation results suggest that interactive wrapping of these common data visualisations (charts, maps, networks) around topologically closed surfaces (cylinder, sphere, torus) could be more routinely applied to common types of data visualisations, e.g., cylindrical wrapping for comparing values and inspecting trends or patterns in online charts with temporal periodicity (e.g. daily, weekly or monthly),

boundaryless views of online social networks without central or peripheral nodes, visual network cluster identification (e.g., social networks with highly connected community structures) using toroidal or spherical wrapping, and online world maps (e.g., for educational training purposes or international travel guide) with interactive spherical wrapping.

9.2 Discussion

It should be noted that the benchmark comparisons and network data studies described in (b) and (c) of the previous section compared only one variant of force-directed 2D layout (section 5.1) with various wrapped alternatives based on a similar stress minimisation approach. This choice followed a careful study design methodology of trying to ensure that the layout approach used in each condition was as similar as possible, with the only difference between the techniques under comparison being whether and how they are topologically wrapped. However, there are many different layout algorithms for 2D node-link visualisations such as orthogonal link layout by Batini et al. [BNT86; Kie+15], layered layout by Sugiyama et al. [STT81], network layout with data aggregation by Yoghourdjian et al. [Yog+18] and it is possible that other layout algorithms could be adapted to spherical and torus embedding. There are also algorithms which can optimise layout for a known set of clusters (i.e. where the cluster labelling is known in advance) [Meu+13; CPC09; GHK09; HGK10]. While these approaches might bring back some accuracy benefits while eliminating the (costly) need for interaction, for the tasks tested in this thesis we do not pre-identify the clusters but rather leave cluster identification as the user task as there are many real-world data analysis scenarios where pre-identified clusters are not available or can not be easily computed. We leave the exploration of other layout algorithms as future work.

Another interesting issue is that for spherical wrapping, the use of interactive wrapping and rotation changes the orientation of the maps. In the map projection study in chapter 7, we did not constrain the wrapping interaction, i.e. users could rotate the map in any direction. The resulting projected visualisation might therefore not be as intuitive for users after rotation, as north and south poles are no longer at the north and south, and may not fit with their conception of spatial data (maps) and non-spatial (networks). But for networks, such distortions may be less harmful as there is often no meaning to the spatial information, for example, there are no logical start and end points of the data, nor even a sensible "inside" or "outside".

Higher distortion might even help make clusters more visible by increasing their separation.

We also note that there are additional variations of parameters possible for further study, beyond those considered in our existing controlled experiment. These could include parameters such as the level of difficulty in the distance comparison tasks for geographic comprehension, as described in individual limitations sections in previous chapters.

In the following, we reiterate the value of the results and insights from the perspective of different data types (networks, maps, time series). For networks, we also provide an informal reflection on practical applications that leads us to identify limitations of our existing work and to call for more formal studies, particularly with application domain experts, in the future.

9.3 Network Visualisations

With respect to abstract data such as networks, the analysis of our experimental results show that the additional "spreading out" that is possible in torus-based layout improves graph layout aesthetics and achieves greater cluster separation than traditional flat node-link layout, and furthermore that the sphere may provide a better use of available space. To further explore the real-world usefulness of wrapping layout, we apply the torus and sphere topology-based layout algorithms described in chapter 6 and chapter 8 to other types of real-world networks. Note that the discussion below is based on our own informal reflections and observations. We recommend that future work explores these applications using more formal user-centred study.

In chapter 6 and chapter 8, we show that torus- and sphere-based network layouts provide benefits for larger complex networks for cluster identification tasks while torus works equally well to non-wrapped flat node-link layouts for sparse and smaller networks. To further investigate this question from the perspective of real-world utility, we explore networks that can be too complex when presenting using conventional layout techniques to reveal meaningful patterns. We argue that the additional flexibility afforded in torus or sphere topology may untangle dense structure to benefit readability.

Preliminarily, we explored a number of real-world social networks. These particular networks were chosen according to a number of criteria. First, they are all used

as examples in seminal social network visualisation papers. Second, they all have dense small-world [WS98], bipartite (defined in subsection 9.3.3) or community structure [GN02] which is difficult to represent in typical non-wrapped flat network diagrams [Sak+14; Wan+15], as described in section 2.1.

9.3.1 Football network

This football network represents American football games between Division IA colleges during the regular season in fall in 2000 by Girvan and Newman [GN02]. This dataset contains 115 nodes of teams (college names), 669 links of regular season games between the two teams they connect, and 9-14 clusters, i.e., subsets of nodes where nodes are highly connected within the subset than outside the subsets (also known as communities defined by Newman [New06]). However, such networks usually have scale-free or small-world properties, and thus when representing the entire network, many links cross the diagram leading to a densely connected layout such that it can not be untangled sufficiently to understand connectivity between teams, as seen in Figure 9.2(a).

This network has been previously evaluated in social network visualisation literature. For example, Wang et al. [Wan+15] proposed methods to highlight visual ambiguities as a result of highly connected network structure that leads to visual cluster overlap. They encoded cluster information in the layouts based on pre-computed community detection to better show cluster boundaries. Compared with Wang et al.'s approach, our torus and sphere-based layouts do not take any advantage of pre-identified cluster information.

Using our layout approaches discussed in chapter 6 and chapter 8, Figure 9.2(b-c) shows that toroidal and spherical representations of the football network have a better use of available space to provide greater separation between hub nodes. This example also shows that the toroidal or spherical layouts may result in potential advantages for better distribution of node positions across the plane that lead to reduced visual clutter, and thus may support hubs identification tasks.

9.3.2 Recipe network

This recipe network represents cooking ingredients (nodes) connected by weighted links (edges) based on how frequent they are used together in recipes by Ahn et al. [Ahn+11]. This dataset was derived by Okoe et al. [OJK18] who removed
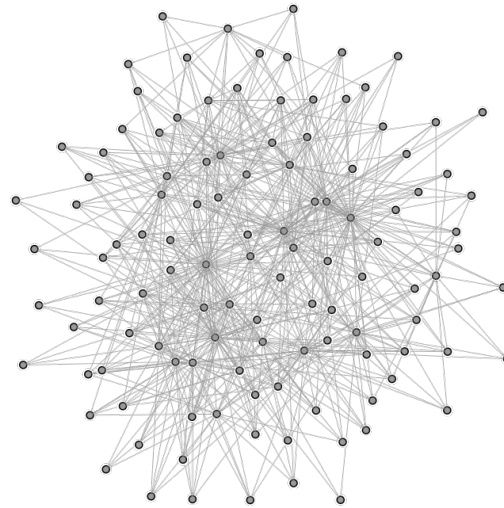
(a) Football-NoTorus
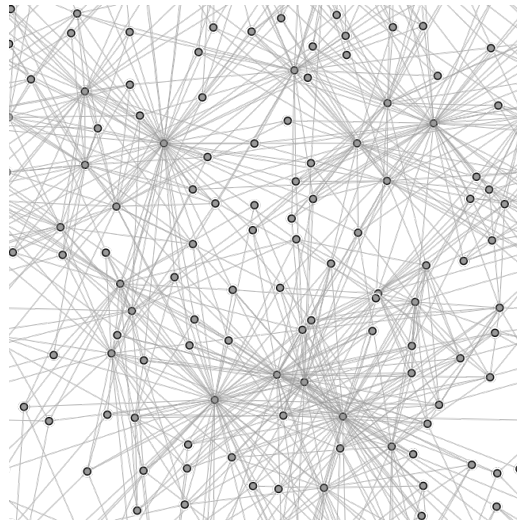
(b) Football-Torus
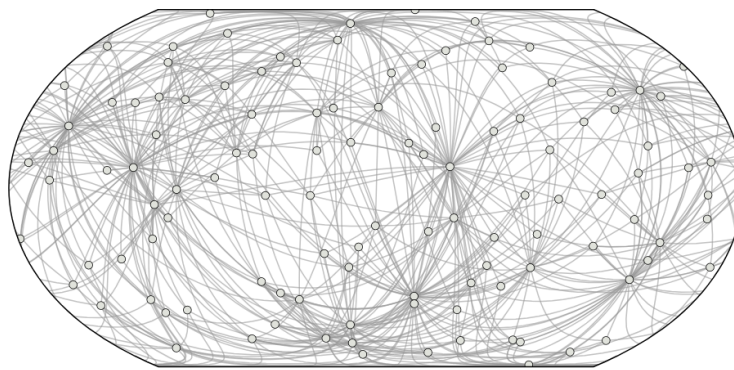
(c) Football-Equal Earth ⬭

Fig. 9.2.: Non-wrapped node-link, toroidal, and spherical representations of an American College football network (115 nodes, 669 links) [GN02], laid out using the algorithms discussed in chapter 6 and chapter 8. The toroidal and spherical layout in (b) and (c) provide better use of the available space to provide greater separation between hub nodes.
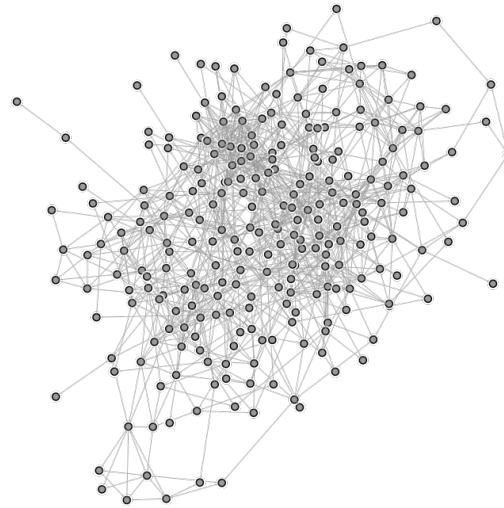
(a) Recipe-NoTorus
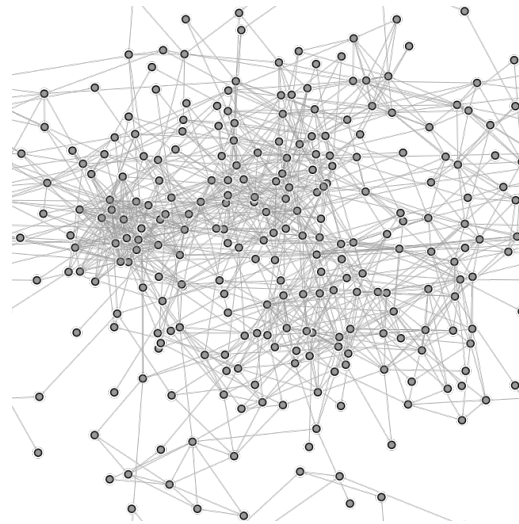
(b) Recipe-Torus
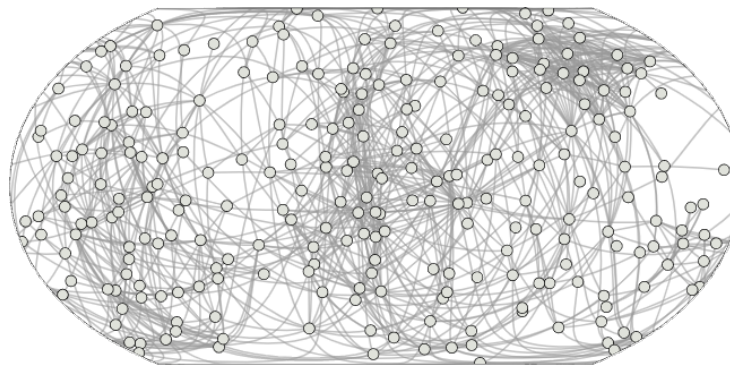
(c) Recipe-Equal Earth ⬭

Fig. 9.3.: Non-wrapped flat node link, torus, and sphere representation of recipe network, laid out using the algorithms discussed in chapter 6 and chapter 8. The dataset was originally created by Ahn et al. [Ahn+11] and derived by Okoe et al. [OJK18]. It has 258 nodes, 1090 links, and 8 clusters. The toroidal and spherical layout in (b) and (c) make it easier than the node-link representations in (a) to identify that there are three main clusters in the network, which supports cluster understanding tasks.

disconnected components and removed low-weight edges, resulting in 258 nodes, 1090 edges, and 8 clusters. Variants of this network have been evaluated in user-centred studies previously in network visualisation literature for its usability by a wide range of participants by Saket et al. [Sak+14] and Okoe et al. [OJK18]. For example, Okoe et al. evaluated readability of node-link visualisations against adjacency matrix, a common alternative to node-link visualisation, finding that node-link performed better than matrix for low-level graph exploration tasks such as path following tasks in terms of accuracy and speed. However, similar to Saket et al. [Sak+14] and other existing work [Meu+13; CPC09; GHK09; HGK10] which can optimise layout for a known set of clusters (i.e. where the cluster labelling is known in advance), Okoe et al.'s network layout also relied on pre-identified cluster information. However, for the layouts and tasks tested in this thesis we do not pre-identify clusters but rather base on a simple generic approach like multidimensional scaling (described in subsection 9.4.1) or force-directed (described in chapter 1) in general, which implicitly group nodes by finding suitable node positions that minimise the difference between pairwise node distance and their graph theoretic distance (e.g., computed by an all-pairs shortest path method).

In our example, we show the layouts where we treat all the links (edges) equally weighted. Figure 9.3(a) shows that for the recipe network in typical force-directed node-link layout, there are many links crossing the diagram leading to a dense, cluttered "hairball", making it hard to determine the major clusters or structures in the underlying network's data. The toroidal and spherical layout in Figure 9.3(b) and Figure 9.3(c) show that the layout appears more spread out with the clusters more distinct, making it easier than the traditional flat node-link representations to identify that there are three main clusters in the network. This example shows that the toroidal or spherical layouts result in potential advantages for cluster identification and supports cluster understanding tasks.

9.3.3 News co-mention network

Compared with previous two examples showing typical community structures in social networks where tasks such as hubs or cluster identification may be beneficial for network layouts that wrap around topologies, the next example we consider has a different structure, i.e., it is *bipartite*. A bipartite network models relationships between two distinct sets of things (nodes), i.e., the intersection of the two sets is the empty set, where relationships (links) can only occur between the sets (i.e. from a node in one set to a node in the other set), not within the same set (i.e. from a node in one set to another node in the same set). This type of network is
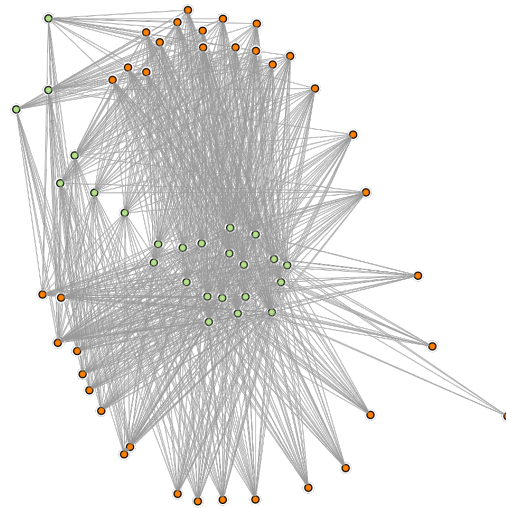
(a) News-co-mention-NoTorus
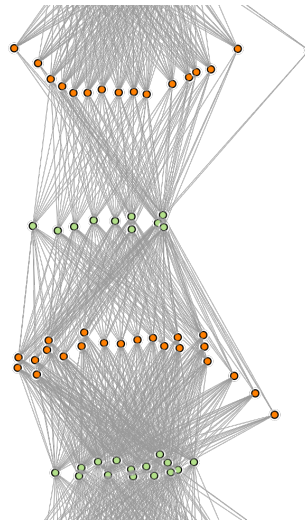
(b) News-co-mention-Torus
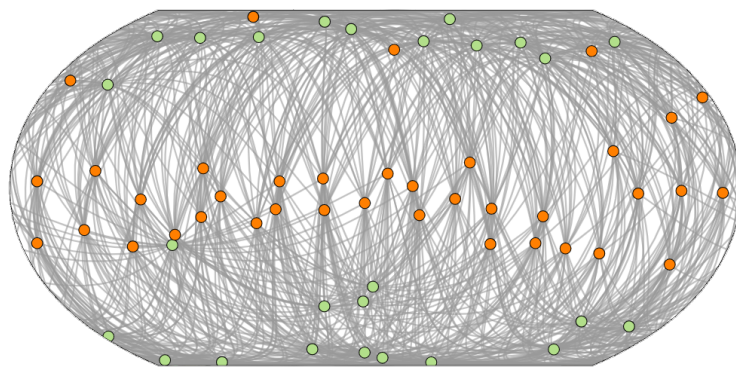
(c) News-co-mention-Equal Earth ⬭

Fig. 9.4.: (a-b) shows node-link and toroidal representations of a news co-mention bipartite network (65 nodes, 899 links). The dataset is credited to Faulkner et al. [Fau+20]. There are two types of nodes: group and topic. The link indicates the topic is mentioned in the group. colours indicate group (orange) or topic (light green). The toroidal layout in (b) reveals a visual pattern of four bands of alternating group and topic nodes, where groups of nodes alternate their positions and wrap around in one vertical direction. However, this is somehow confusing when seeing four groups rather than two groups in a bipartite network, but it is potentially easier than the flat layout to identify the pattern that reveals its underlying data structure.

ubiquitous in a variety of domains, e.g. to represent relationships between two different types of things, such as the relationships between women and events in Davis Southern Women social network [DGG09], between users and their preferred movies for recommendation systems [Jan+10], or between groups and topics such as the news co-mention network studied by Faulkner et al. [Fau+20].

Figure 9.4 shows the news co-mention network studied by Faulkner et al. [Fau+20]. It is a bipartite network of two sets: groups of people and topics with which they are associated in the news network. There are a total of 39 groups and 26 topics leading to 65 nodes. There are 899 associated connections (links) between groups and topics or issues. A link indicates the topic is mentioned in the group. Colours indicate group (orange) or topic (light green). The flat layout in (a) is based on a standard force-directed layout and does not do a good job of separating group nodes from topic nodes. That is, we must carefully inspect the links to see that there are no links between topics and no links between groups. However, the toroidal layout in Figure 9.4(b) is much clearer as it separates nodes into four rows, where from the top row to the bottom it represents topics, groups, and then repeating topics, and groups, in an alternate fashion. Links at the bottom row wrap around to the top row, showing the connectivity between groups (bottom row) and topics (top row). We have found that such bipartite networks are typically arranged by the toroidal layout algorithm such that they only wrap on one side (i.e. cylindrical wrapping). Within this cylindrical arrangement, we also find that dense-bipartite networks such as this one are arranged into several bands, with the nodes from the two partitions divided into two bands each. A viewer can use touch or mouse drag to interactively wrap the network vertically to observe the connectivity between any two adjacent rows. By contrast, the spherical layout in Figure 9.4(c) does not reveal any obvious structure. However, since such spherical projections are really cylindrical (wrapping in one dimension rather than two dimensions, as we saw in our design space - chapter 3), we conjecture that a similar banded layout should be possible on the sphere with a modified force model.

It seems such torus layouts of a bipartite network may reveal a visual pattern of alternating sets of groups and topics that may be potentially better to identify bipartite relationships than the flat layout. However, a potential disadvantage is that representing bipartite relationships in four rows may cause confusion to understand that there are only two sets. We would expect that a trained user would be able to easily recognise such alternating layers as a bipartite structure (and would suggest a future study to evaluate this hypothesis).

9.3.4 Product space network

In previous subsections we have seen examples where 2D network layouts that wrap around continuous 3D torus or sphere surfaces may provide benefits over their non-wrapped flat representations. This subsection gives a counterexample that shows wrapped layouts may not be suitable for complex networks whose subsets contain many sparse tree-like structures. We explore the product space network derived from international trade data over decades, created by the Harvard's Growth Lab using Harmonised System (HS) format [Har19]. This network represents 866 products (nodes) connected by 2532 weighted links (edges) based on the similarities of know-how required to produce the two products where they connect. Tools that visualise such networks have been presented by Simoes et al. [SH11] and its network visualisations widely used by various communities [Obe20; CMV20] and social dynamics research [Li+19].

Figure 9.5(a) shows that in the typical non-wrapped node-link representation it is straightforward to observe that nodes further away from the dense structure have a very clear tree-like structure. By contrast, it is almost impossible to discern any high-level network structure from the dense structure due to tightly connected nodes within the structure. The torus (Figure 9.5(b)) and spherical (Figure 9.5(c)) layouts further untangles the dense structure. However, not only are nodes from the dense structure more "spread out" than non-wrapped layout, but the tree structures are also pushed further towards the edge of the display that "wrap around". As shown in Figure 9.5(b) it sometimes becomes ambiguous whether the tree structures extending from the top boundary are wrapped to the bottom, and the tree structures extending from the bottom wrapped to the top actually connect or are distinct, due to the visual overlap when these tree branches are spread out and wrapped. Similarly, Figure 9.5(c) shows that the tree branches wrapped across the left-and-right boundary may lead to visual ambiguities in their connectivity without introducing artefacts to explicitly highlighting each distinct tree structure.

Since visualising such a network that is mostly fairly sparse, but has a dense core on toroidal or spherical topology does not really seem to take advantage of the torus topology's ability to reduce dense clutter (as we saw in chapter 6), we propose that another way to use toroidal layout for such a graph is to analyse just the dense part in isolation - perhaps as a detailed view that complements a more conventional flat layout of the whole graph. To test this, we extracted the largest bi-connected subset (i.e. the largest subgraph where removing any one node leaves the subgraph connected) of the original product space network. This leaves us a smaller network with 385 nodes and 2031 links.
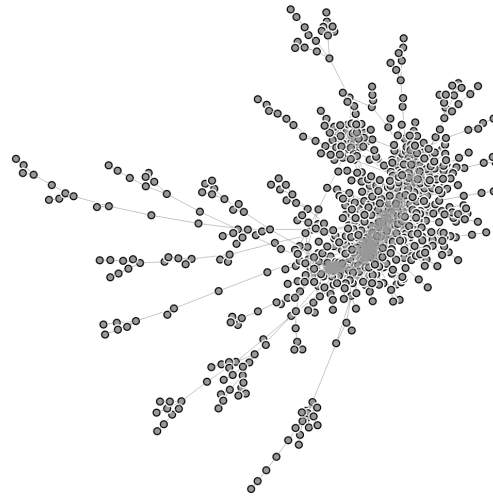
(a) ProductSpace-NoTorus
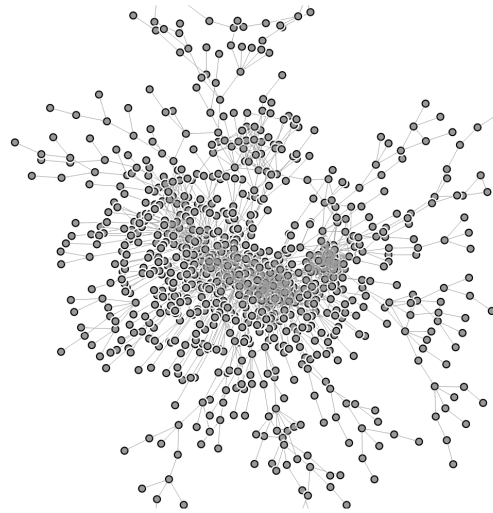
(b) ProductSpace-Torus
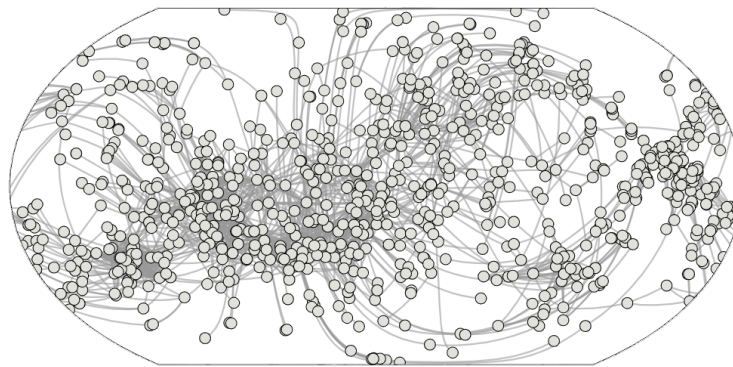
(c) ProductSpace-Equal Earth ⬭

Fig. 9.5.: Product space network laid out using standard flat, and our torus and equal earth visualisations. The data sets are from the original product space network created by Harvard Growth Lab [Har19].
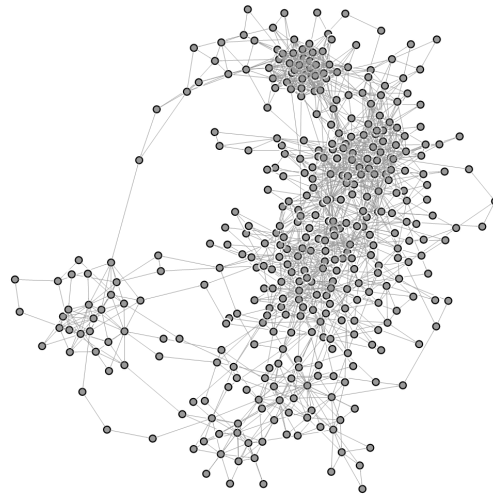
(a) ProductSpace-NoTorus
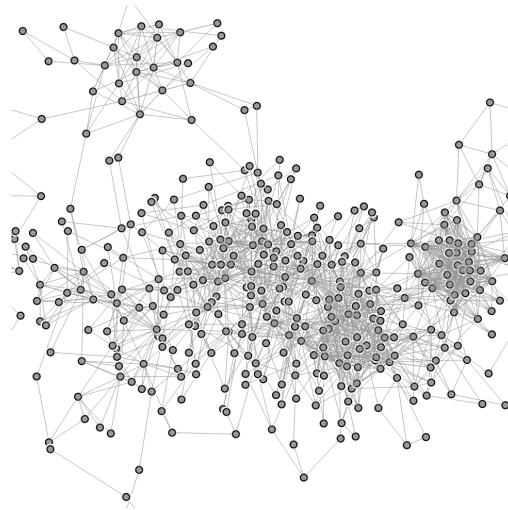
(b) ProductSpace-Torus
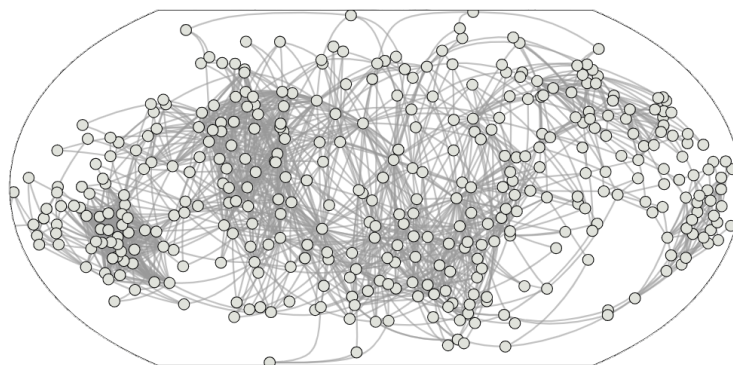
(c) ProductSpace-Equal Earth 🔘

Fig. 9.6.: Product space data network of derived datasets from Harvard Growth Lab

Figure 9.6 shows that compared with non-wrapped flat layout in (a), sphere layout in (c) appears more spread out with clusters more distinct. However, there is a limited benefit in the torus layout (b) as opposed to the non-wrapped (a), as the dense structure remains dense. Adapting our layout algorithm from chapter 6 to considering more advanced layout methods such as weighted link wrapping such that a dense structure could be more aggressively wrapped, while leaving the sparse tree structure unwrapped with a low weight of link wrapping may do a better job of revealing the dense structure while preserving the tree structure. Schemes for automating, or making interactive, the selection of a dense subgraph within a larger graph for detailed inspection in a toroidal or spherical "lens" is an interesting direction for future research.

In summary, in above examples, we have seen networks which have dense small-world (subsection 9.3.1, subsection 9.3.4), bipartite (subsection 9.3.3) or community structure (subsection 9.3.2) which is difficult to represent in typical non-wrapped flat network diagrams. While wrapped toroidal or spherical layouts of these examples may suggest interesting applications, the exploration also shows some limitations of toroidal or spherical layouts for product space or bipartite networks. We would like to further improve the algorithms and investigate if torus or spherical representations could be practically usable by domain experts. While this section describes an informal reflection of practical applications of interactive wrapping, we consider a more formal user-centred study with domain experts to further investigate the effectiveness and usability.

Meanwhile, developing effective network layout algorithms is key to real-world practical network visualisations and assists future research in wider communities. While our refined layout algorithm proposed in chapter 6 converges significantly more quickly than our first toroidal layout approach in chapter 5, the need to compare 9 different alternatives for link wrappings at each iteration means that it is still considerably slower than the corresponding layout algorithms for traditional node-link diagram layout. Improving its speed is a major direction for future research.

Another future direction is to explore whether the extra flexibility in wrapping networks in 2D plane of topologically closing surfaces also provide benefits for more complex multivariate network visualisation that involve representation of both node-link and their attributes [Nob+19] such as socio-semantic of internet community networks encode social connections (e.g., friendship, co-citation, co-occurrence, mentorship) among social actors and cultural associations (e.g., similarities between opinions, cultural schemes, key concepts of publications) [RB20; RSH21; Rot21]
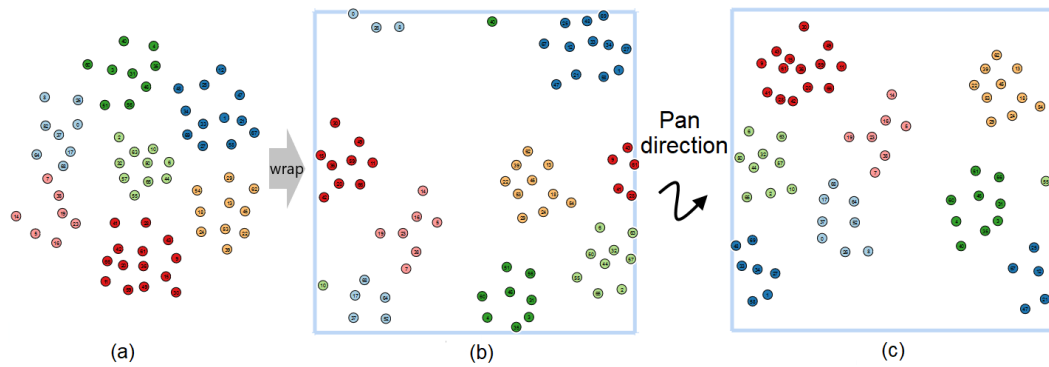
Fig. 9.7.: Examples of torus panning on an MDS using the Hepta data set [TU20] with already known clustering information: (a) traditional (unwrapped) MDS plot on a 2D-plane; (b, c) two configurations wrapped MDS plot laid-out on a 2D torus, using the algorithm presented in chapter 6. The dataset was adapted from a Benchmark data set from Fundamental Cluster Problem Suite (FCPS).

and protein-protein interaction networks that are highly connected with dense structure.

9.4 Other possibilities for future work

In this section, we discuss other possibilities for future work, including wrapping visualisations for other data types discussed in our design space that have not been evaluated, such as multidimensional scaling plots, self-organising maps, and cyclical time series with multiple levels of periodic dimensions. Apart from the wrapping approach using interactive panning, possibilities of static wrapped display using tile repetition for geographic maps are also presented.

9.4.1 Interactive wrapping for multidimensional scaling

As introduced in subsection 2.1.8, many real-world data structures contain objects that have multiple attributes (e.g., a patient has temperature, weight, height, blood pressure, etc). In such multidimensional data analysis, one important goal for dimension reduction methods such as multidimensional scaling (MDS) (introduced in subsection 2.1.8) is to seek to optimise the pairwise low-dimensional distance (e.g. Euclidean distance) between data points to as well as possible to represent the pairwise similarity or dissimilarity in the original data; and ultimately, to reveal the underlying structure for convenient perception of clusters, correlation between data points, ratings, and similarities between statistics data.

This process, known as stress minimisation [Mar+12; Bia+20], solves the same type of problem we consider for optimising node placement to simulate their graph theoretic distance in this thesis. However, like force-directed node placement problem, stress minimisation is also a non-trivial problem, and thus many heuristics algorithms exist, such as standard gradient descent, stochastic gradient descent by Bottou [Bot10], stress majorisation by Gansner et al. [GKN04] and Leeuw and Mair [DM09].

For complex data (such as larger and denser datasets), MDS plots need a cleaner presentation to discern density distribution or identifying high-level structures such as clusters [VH08; LCY19]. Based on the similarity between MDS and network layout algorithms explained in more detail in subsection 2.1.8, and the promising results of improved layout quality (in terms of aesthetics) when wrapping a network on a torus topology, and improved task performance (in terms of accuracy) when introduced with interactive wrapping for cluster identification tasks for sphere or torus topology, we argue that it is also likely to lead to interesting applications for MDS.

Figure 9.7 shows MDS plots of a common Hepta data set which has been evaluated by Thrun [TU20]. It has already known clustering information. Using the network layout algorithms proposed in chapter 6, it shows that MDS can potentially wrap around a 2D projected torus where the arrangement of data points have possible additional spreading and make better use of available space, while exploring with interactive panning. Future work should evaluate the effect of torus wrapping on perception, e.g. using interactive wrapping or tile display.

9.4.2 Interactive wrapping for self-organising maps

Another example of arrangements of abstract data that seeks to optimise the layouts to better understand their relationships or the structure is self-organising maps (SOMs). SOMs, described in chapter 2 are commonly used as an abstract data structure that represents multidimensional data, i.e., a patient has weight, height, blood pressure, etc in lower dimensions[Koh82; Squ+05; WT06a]. Discerning high-level structure, such as clusters, is an important task in SOMs. In SOM, data points can be grouped into clusters where data points in spatially nearby clusters have more similar values than data points in distal clusters.

Many methods exist for preserving the topological structure of multidimensional data in lower dimensions [Koh82] on self-organising maps of documents. In order to alleviate the "privileged centre effect" (or border effect studied by Wu et

(a) TorusSOM   (b) SphereSOM
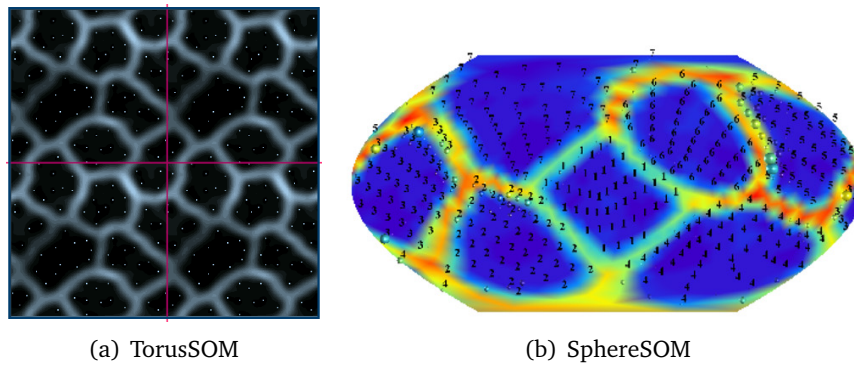
Fig. 9.8.: SOM rendered on (a) a torus layout with tile repetition connecting regions of data wrapped across the boundaries. Image is credited to [Ult03]; (b) a 2D spherical projection where regions split across the left-and-right boundaries. The image at (a) and (b) are credited to Ultsch et al. [Ult03] and Wu et al. [WT06a], respectively.

al. [WT06a]), where clusters at the centre have more neighbours than those at the boundary and thus are updated more often than those at the border, SOMs have been embedded on a sphere or torus. Potential advantages include no concept of border so that all the clusters may be considered when updating neighbouring relationships. Figure 9.8(a) shows a toroidal tile repetition of SOM where the clusters wrap around in both horizontal and vertical directions. Similarly, Figure 9.8(b) shows equal earth representation of SOMs where clusters wrap around horizontally.

However, above existing research focused on visualising SOMs either visualising in 3D or projecting to a static 2D plane. The effect of interactive wrapping of these representations has not previously been presented. Based on our study results, we might hypothesise that interactive wrapping or tile display (Figure 9.8(a)) reinforces understanding about the neighbourhood relationship of SOMs wrapped across the edge of the display, leaving exploration of such an effect on usability and readability of SOMs to a future work.

9.4.3 Interactive torus wrapping for cyclical time series

We have shown in chapter 4 that one dimensional (cylindrical) wrapping provides considerable benefits for understanding real-world cyclical time series with one periodic dimension (e.g., 24 hours over a day, or 12 months over a year). However, cyclical time series has multiple levels of periodicity (e.g. weekly as well as daily). Effective visual representation of time series data for multiple levels of temporal
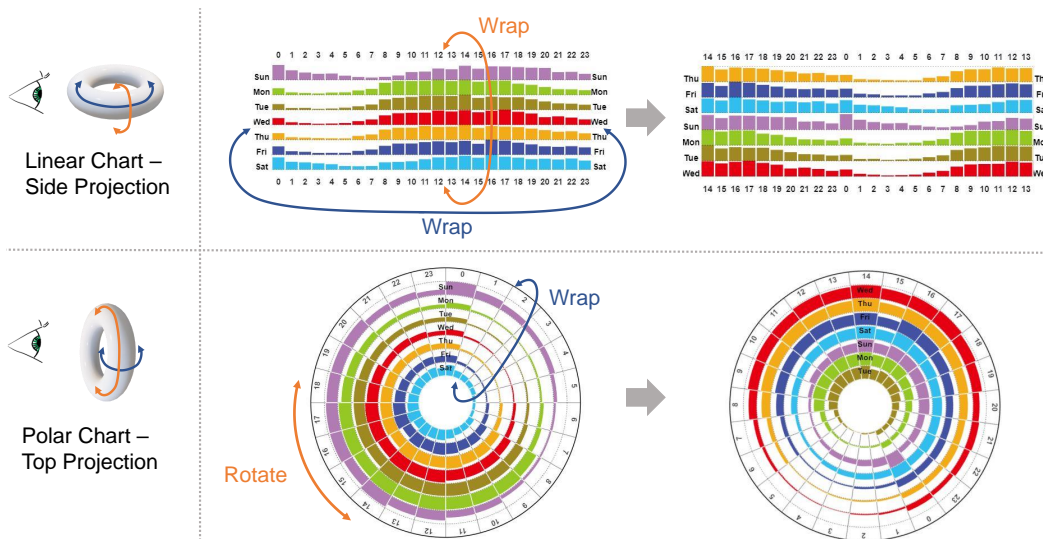
Fig. 9.9.: Cyclical time series using torus wrapping and rotation can help to understand the visual cut that are otherwise split across the left-and-right boundary and top-and-bottom boundary. The data source is from Kaggle open dataset [Tro22].

dimensions to reveal meaningful trends and patterns, and improving their readability remain open questions [Bre+16; Bac+17].

Extending the concept of one-dimensional wrapping, we have introduced two-dimensional toroidal wrapping for cyclical time series data that show multiple levels of periodicity (e.g., daily and weekly, in subsection 3.4.1), including 2D bar chart arrays arranged on a side-projected torus for interactive wrapping, polar chart arrays arranged on a side-projected torus for rotatable panning and wrapping (Figure 9.9), and furthermore tiled display by replicating linear bar charts of cyclical time series on the sides of the charts both horizontally and vertically (Figure 9.10) to provide full connectivity wrapped across the boundaries without being hindered (when reading the charts) by an arbitrary visual "cut" (defined in chapter 1).

In 2D linear bar chart arrays, having 0am on the left, 11pm on the right, or Sunday on the top, Saturday on the bottom (Figure 9.9-upper left) is completely arbitrary. Similarly, in polar chart arrays, having the bars (hours) arranged along layered circumferences of the chart in an order analogous to a natural clock (Figure 9.9-lower left) across Sunday to Saturday is not the only option. Using interactive wrapping proposed in our design space (section 3.4), a viewer can move the linear charts horizontally to see a continuous range of bars between, e.g., 11pm and 6am without being split by the cut across the left-and-right boundary, while panning the charts vertically connect the cuts between Friday and Sunday (Figure 9.9-upper right). Interactively rotating a polar chart (derived from our design space
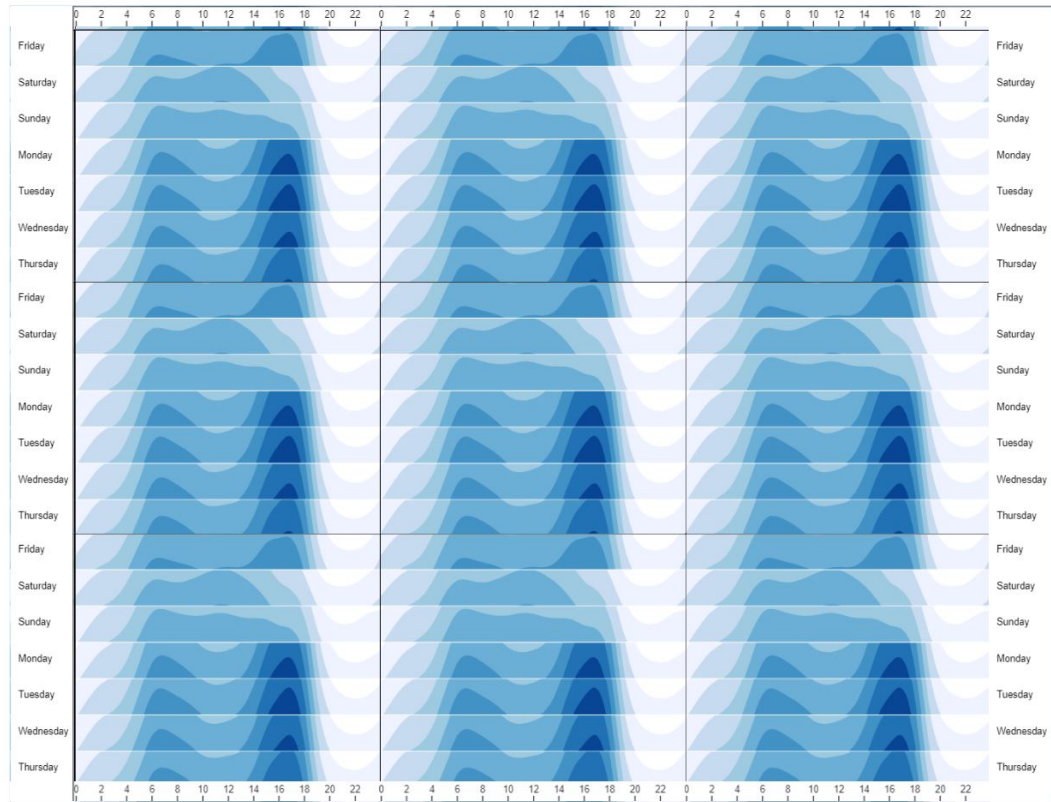
Fig. 9.10.: Example of cyclical time series of New South Wales traffic accidents represented on a 2D projected torus topology with 3×3 tile display as described in subsection 3.4.1. Two periodic dimensions such as 24 hours of a day across 7 days a week repeat across the top, bottom, left, and right edges of the original chart (shown in the central tile) without repetition. It allows a viewer to observe continuous ranges of both dimensions across the chart's boundaries with any break in the temporal dimensions.
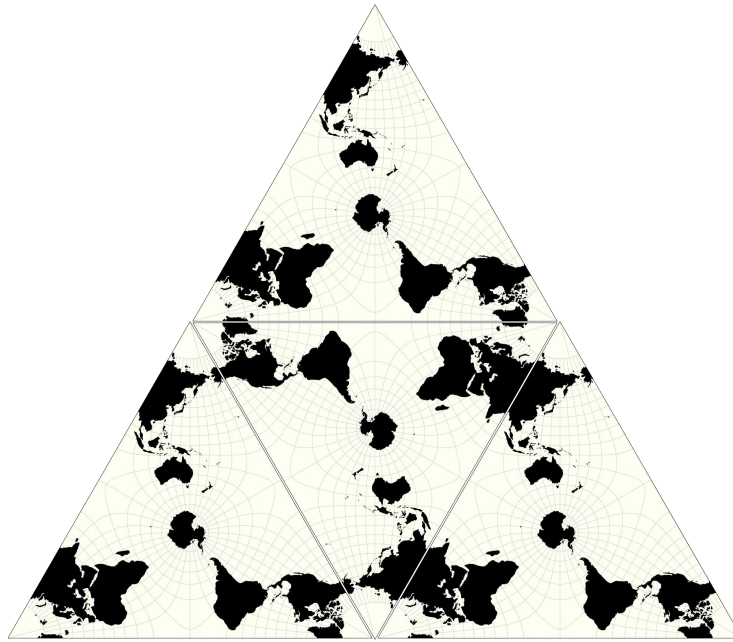
Fig. 9.11.: Tile display of maps in Lees tetrahedron projection; image adapted from Lees tetrahedron projection and credited to Lee

in subsection 3.4.1) along the hour dimension allows for centring the hour period between, e.g., 9am and 8pm for a more aligned comparison during this period, while wrapping the polar charts moves Wednesday from the inside of the polar visualisation to its outside to provide a continuous view between, e.g., Friday and Sunday (Figure 9.9-lower right).

Based on the promising results of readability of linear bar charts with cylindrical wrapping (subsection 4.2.8), we might hypothesise that, for more complicated case of 2D temporal data exploration in a 2D torus topology, interactive wrapping with two temporal dimensions may still outperform the rotational and wraparound polar chart. In future, we intend to further investigate this hypothesis and also compare with the effect of static tile repetition without such interaction.

9.4.4 Other possibilities for future work

In section 3.1 we presented two approaches of wrapping: tile display and interactive wrapping. Our evaluation results show that adding tile repetition to torus layouts provides significant benefits for understanding link wrapping across the boundaries, while after adding interactive panning, torus layouts without tile display performs equally well with either the tile display or non-wrapped layouts (chapter 5). However,
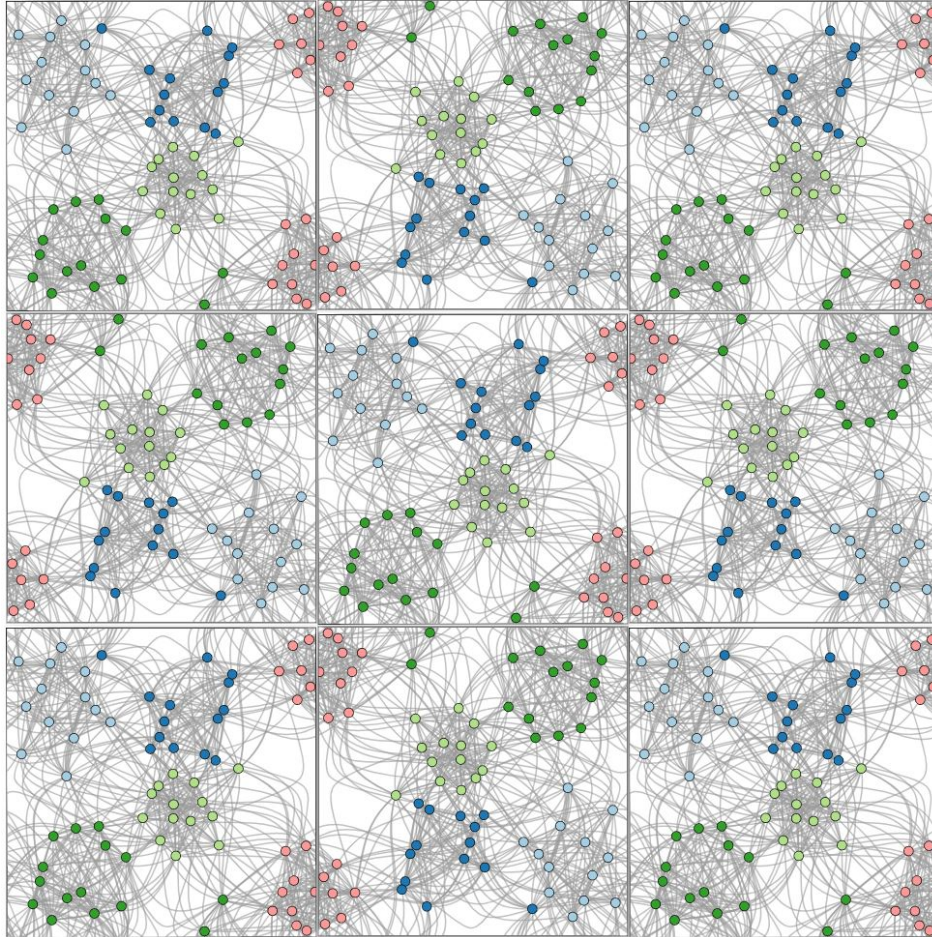
Fig. 9.12.: Pierceś Quincuncial projection of "Wrapping" dimension in both spatial directions: horizontally and vertically repeated tiles. The repeated tiles show continuity of data points at the boundaries which preserves the data point movement wrapped across the boundaries.

it has not been shown whether tile displays of other topologies considered in our design space, such as cylinder (Figure 3.4) or sphere (Figure 9.11 and Figure 9.12), also provide benefits over static/interactive wrapped (without tile repetition) or non-wrapped representations.

It is also possible to further investigate the aspects of the wrapping method that drive better performance. From a theoretical perspective, it would be interesting to consider what aspects of the wrapping method driving the better performance. Is it getting to see the data from multiple views? Or is it the active aspect of getting to manipulate the data to the desired view? We cannot easily say whether the interactive panning we provided is better than a passive animation of wrapping affording different views.

While answering these questions is beyond the scope of this thesis. One way to do this would be to repeat the study that was done in chapter 4, chapter 6, chapter 7, and video record the interactions (active group). Then, "yoke" each participant in this group to a new set of participants in a second group, who watches video recordings of the interaction rather than doing the interactions themselves (passive yoked group). If both groups perform similarly well for the interactive wrapped visualisations compared with the static ones, that would suggest that the benefits of this new method is the multiple views, but if the active group does better, that would suggest that interacting with the visualisations is key. We leave this to future work.

9.5 Wrapping up

We began this thesis by reviewing the related research on data visualisation, revisiting existing data that wraps around topologies, network layout algorithms, and empirical evidence (chapter 2). Afterwards, we presented a design space exploration which identifies various data types that wrapping visualisations makes sense, and our interactive tools design that create novel applications (chapter 3). We then discussed each of the wrapping topologies and effectiveness evaluation: cylindrical (chapter 4), toroidal (chapter 5 – smaller networks and chapter 6 – larger networks), and spherical topologies (chapter 7 – maps and chapter 8 – networks). We gave a discussion of the benefits and limitations of interactive wrapping and provide our view on the future of pannable wrapped visualisations and the potential challenges that provide implications to future research (chapter 9).

Finally, we concluded this thesis with a contribution and a clear take-away message: It's a Wrap! Visualisations can be brought out of the box to better analyse the trends, identify patterns, and explore the relationships within the data; ultimately, we hope it helps make better decisions.

Bibliography

[AJB16]   Muhammad Adnan, Mike Just, and Lynne Baillie. "Investigating time series visualisations to improve the user experience". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 5444–5455 (cit. on pp. 4, 36, 37, 74, 77, 78, 89, 90).

[Ahn+11]   Yong-Yeol Ahn, Sebastian E Ahnert, James P Bagrow, and Albert-László Barabási. "Flavor network and the principles of food pairing". In: *Scientific reports* 1.1 (2011), pp. 1–7 (cit. on pp. 17, 191, 193).

[Aig+11]   Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, 2011 (cit. on p. 36).

[ACG14]   Danielle Albers, Michael Correll, and Michael Gleicher. "Task-driven evaluation of aggregation in time series visualization". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, pp. 551–560 (cit. on pp. 77, 80).

[Alp+13]   Basak Alper, Benjamin Bach, Nathalie Henry Riche, Tobias Isenberg, and Jean-Daniel Fekete. "Weighted graph comparison techniques for brain connectivity analysis". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2013, pp. 483–492 (cit. on pp. 19, 21, 28).

[AA06]   Natalia Andrienko and Gennady Andrienko. *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer Science & Business Media, 2006 (cit. on p. 36).

[Bac+17]   Benjamin Bach, Pierre Dragicevic, Daniel Archambault, Christophe Hurter, and Sheelagh Carpendale. "A descriptive framework for temporal data visualizations based on generalized space-time cubes". In: *Computer Graphics Forum*. Vol. 36. 6. Wiley Online Library. 2017, pp. 36–61 (cit. on pp. 36, 204).

[Bac+15a]   Benjamin Bach, Nathalie Henry-Riche, Tim Dwyer, et al. "Small MultiPiles: Piling time to explore temporal patterns in dynamic networks". In: *Computer Graphics Forum*. Vol. 34. 3. Wiley Online Library. 2015, pp. 31–40 (cit. on p. 36).

[Bac+16]   Benjamin Bach, Nathalie Henry Riche, Christophe Hurter, Kim Marriott, and Tim Dwyer. "Towards unambiguous edge bundling: Investigating confluent drawings for network visualization". In: *IEEE transactions on visualization and computer graphics* 23.1 (2016), pp. 541–550 (cit. on p. 19).

[Bac+15b] Benjamin Bach, Conglei Shi, Nicolas Heulot, et al. "Time curves: Folding time to visualize patterns of temporal evolution in data". In: *IEEE transactions on visualization and computer graphics* 22.1 (2015), pp. 559–568 (cit. on pp. 36, 53).

[BA99] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *science* 286.5439 (1999), pp. 509–512 (cit. on pp. 17, 28, 174).

[Bat+15] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. "Fitting Linear Mixed-Effects Models Using **lme4**". en. In: *Journal of Statistical Software* 67.1 (2015) (cit. on p. 156).

[BNT86] Carlo Batini, Enrico Nardelli, and Roberto Tamassia. "A layout algorithm for data flow diagrams". In: *IEEE Transactions on Software Engineering* 4 (1986), pp. 538–546 (cit. on pp. 16, 117, 189).

[Bat+98] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998 (cit. on p. 16).

[Beh+16] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. "Matrix reordering methods for table and network visualization". In: *Computer Graphics Forum*. Vol. 35. 3. Wiley Online Library. 2016, pp. 693–716 (cit. on pp. 28, 64).

[Ben+07] Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. "The aesthetics of graph visualization." In: *Computational Aesthetics* 2007 (2007), pp. 57–64 (cit. on p. 99).

[BAI17] Lonni Besançon, Mehdi Ammi, and Tobias Isenberg. "Pressure-based gain factor control for mobile 3D interaction using locally-coupled devices". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 1831–1842 (cit. on p. 156).

[Bes+17] Lonni Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. "Mouse, Tactile, and Tangible Input for 3D Manipulation". In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. Denver, United States, May 2017, pp. 4727–4740 (cit. on p. 156).

[Bia+20] Rongzheng Bian, Yumeng Xue, Liang Zhou, et al. "Implicit multidimensional projection of local subspaces". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2020), pp. 1558–1568 (cit. on pp. 29, 202).

[Bin+19] Carla Binucci, Ulrik Brandes, Tim Dwyer, et al. "10 Reasons to Get Interested in Graph Drawing". In: *Computing and Software Science*. Springer, 2019, pp. 85–104 (cit. on p. 16).

[BH98] Eric Bonabeau and Florian Hénaux. "Self-organizing maps for drawing large graphs". In: *Information Processing Letters* 67.4 (1998), pp. 177–184 (cit. on p. 31).

[BG05]   Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005 (cit. on p. 29).

[Bor+18]   Rita Borgo, Luana Micallef, Benjamin Bach, Fintan McGee, and Bongshin Lee. "Information visualization evaluation using crowdsourcing". In: *Computer Graphics Forum*. Vol. 37. 3. Wiley Online Library. 2018, pp. 573–595 (cit. on p. 38).

[BD13]   Michael Bostock and Jason Davies. "Code as Cartography". en. In: *The Cartographic Journal* 50.2 (May 2013), pp. 129–135 (cit. on pp. 3, 34).

[Bos20]   Mike Bostok. online. Last accessd: April 1st, 2020. 2020 (cit. on p. 33).

[Bos22]   Mike Bostok. *https://d3js.org/*. 2011 (accessed 15 May, 2022) (cit. on pp. 79, 135).

[Bot10]   Léon Bottou. "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186 (cit. on p. 202).

[BGW03]   Ulrik Brandes, Marco Gaertler, and Dorothea Wagner. "Experiments on graph clustering algorithms". In: *European Symposium on Algorithms*. Springer. 2003, pp. 568–579 (cit. on pp. 127, 174).

[BP08]   Ulrik Brandes and Christian Pich. "An experimental study on distance-based graph drawing". In: *International Symposium on Graph Drawing*. Springer. 2008, pp. 218–229 (cit. on pp. 16, 95).

[BM12]   Richard Brath and Peter MacMurchy. "Sphere-based information visualization: Challenges and benefits". In: *2012 16th International Conference on Information Visualisation*. IEEE. 2012, pp. 1–6 (cit. on pp. 20, 168).

[Bre+16]   Matthew Brehmer, Bongshin Lee, Benjamin Bach, Nathalie Henry Riche, and Tamara Munzner. "Timelines revisited: A design space and considerations for expressive storytelling". In: *IEEE transactions on visualization and computer graphics* 23.9 (2016), pp. 2151–2164 (cit. on pp. 36, 204).

[Bre+18]   Matthew Brehmer, Bongshin Lee, Petra Isenberg, and Eun Kyoung Choe. "Visualizing ranges over time on mobile phones: a task-based crowdsourced evaluation". In: *IEEE transactions on visualization and computer graphics* 25.1 (2018), pp. 619–629 (cit. on pp. 4, 36, 37, 77, 80, 89, 90, 154).

[BM13]   Matthew Brehmer and Tamara Munzner. "A multi-level typology of abstract visualization tasks". In: *IEEE transactions on visualization and computer graphics* 19.12 (2013), pp. 2376–2385 (cit. on p. 77).

[Bur+20]   Michael Burch, Weidong Huang, Mathew Wakefield, et al. "The state of the art in empirical user evaluation of graph visualizations". In: *IEEE Access* 9 (2020), pp. 4173–4198 (cit. on pp. 6, 17, 28).

[Car10]   Claus-Christian Carbon. "The Earth is flat when personally significant experiences with the sphericity of the Earth are absent". In: *Cognition* 116.1 (2010), pp. 130–135 (cit. on pp. 33, 152).

[Car+17]   Marina Montero Carrero, Ward De Paepe, Svend Bram, and Francesco Contino. "Thermodynamic Analysis of Water Injection in a Micro Gas Turbine: Sankey and Grassmann Diagrams". In: *Energy Procedia* 105 (2017). 8th International Conference on Applied Energy, ICAE2016, 8-11 October 2016, Beijing, China, pp. 1414–1419 (cit. on p. 57).

[CH18]   Hsia-Ching Chang and Suliman Hawamdeh. *Analytics and Knowledge Management*. eng. Data Analytics Applications. CRC Press, 2018 (cit. on p. 29).

[Che+21a]   Kun-Ting Chen, Tim Dwyer, Benjamin Bach, and Kim Marriott. "It's a Wrap: Toroidal Wrapping of Network Visualisations Supports Cluster Understanding Tasks". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–12 (cit. on p. 13).

[Che+21b]   Kun-Ting Chen, Tim Dwyer, Benjamin Bach, and Kim Marriott. "Rotate or Wrap? Interactive Visualisations of Cyclical Data on Cylindrical or Toroidal Topologies". In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2021), pp. 727–736 (cit. on p. 12).

[Che+20]   Kun-Ting Chen, Tim Dwyer, Kim Marriott, and Benjamin Bach. "DoughNets: Visualising Networks Using Torus Wrapping". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–11 (cit. on p. 13).

[Che+22]   Kun-Ting Chen, Tim Dwyer, Yalong Yang, Benjamin Bach, and Kim Marriott. "GAN'SDA Wrap: Geographic And Network Structured Data on surfaces that Wrap around". In: *arXiv preprint arXiv:2202.10845* (2022) (cit. on pp. 13, 14).

[CMV20]   Gloria Cicerone, Philip McCann, and Viktor A Venhorst. "Promoting regional growth and innovation: relatedness, revealed comparative advantage and the product space". In: *Journal of Economic Geography* 20.1 (2020), pp. 293–316 (cit. on p. 197).

[CM84]   William S Cleveland and Robert McGill. "Graphical perception: Theory, experimentation, and application to the development of graphical methods". In: *Journal of the American statistical association* 79.387 (1984), pp. 531–554 (cit. on pp. 37, 74).

[Coc+20]   Andy Cockburn, Pierre Dragicevic, Lonni Besançon, and Carl Gutwin. "Threats of a replication crisis in empirical computer science". In: *Communications of the ACM* 63.8 (2020), pp. 70–79 (cit. on pp. 161, 164, 179).

[Coh13]   Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013 (cit. on pp. 156–158, 177, 230, 231, 234).

[CPC09]   Christopher Collins, Gerald Penn, and Sheelagh Carpendale. "Bubble sets: Revealing set relations with isocontours over existing visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 1009–1016 (cit. on pp. 18, 182, 189, 194).

[Cor+12]   Michael Correll, Danielle Albers, Steven Franconeri, and Michael Gleicher. "Comparing averages in time series data". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 1095–1104 (cit. on pp. 77, 80).

[CC08]   Michael AA Cox and Trevor F Cox. "Multidimensional scaling". In: *Handbook of data visualization*. Springer, 2008, pp. 315–347 (cit. on p. 29).

[Dav13]   Jason Davies. *Rotate the World*. 2013 (cit. on pp. 3, 26, 34, 151).

[DGG09]   Allison Davis, Burleigh Bradford Gardner, and Mary R Gardner. *Deep South: A social anthropological study of caste and class*. Univ of South Carolina Press, 2009 (cit. on p. 196).

[DM09]   Jan De Leeuw and Patrick Mair. "Multidimensional scaling using majorization: SMACOF in R". In: *Journal of statistical software* 31.1 (2009), pp. 1–30 (cit. on pp. 29, 51, 202).

[Dev+19]   Sabin Devkota, Reyan Ahmed, Felice De Luca, Katherine E Isaacs, and Stephen Kobourov. "Stress-Plus-X (SPX) Graph Layout". In: *International Symposium on Graph Drawing and Network Visualization*. Springer. 2019, pp. 291–304 (cit. on pp. 16, 95, 136).

[Di +20]   Sara Di Bartolomeo, Aditeya Pandey, Aristotelis Leventidis, et al. "Evaluating the effect of timeline shape on visualization task performance". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–12 (cit. on p. 80).

[Du+17]   Fan Du, Nan Cao, Yu-Ru Lin, Panpan Xu, and Hanghang Tong. "isphere: Focus+ context sphere visualization for interactive large graph exploration". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 2916–2927 (cit. on pp. 21, 174).

[DGK11]   Christian A Duncan, Michael T Goodrich, and Stephen G Kobourov. "Planar drawings of higher-genus graphs." In: *Journal of Graph Algorithms and Applications* 15.1 (2011), pp. 7–32 (cit. on p. 24).

[Dwy09]   Tim Dwyer. "Scalable, versatile and simple constrained graph layout". In: *Computer Graphics Forum*. Vol. 28. 3. Wiley Online Library. 2009, pp. 991–998 (cit. on pp. 16, 95).

[DKM06]   Tim Dwyer, Yehuda Koren, and Kim Marriott. "IPSep-CoLa: An incremental procedure for separation constraint layout of graphs". In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006), pp. 821–828 (cit. on p. 99).

[Dwy+09]   Tim Dwyer, Bongshin Lee, Danyel Fisher, et al. "A comparison of user-generated and automatic graph layouts". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 961–968 (cit. on pp. 27, 98, 103, 136).

[Dwy+08]   Tim Dwyer, Kim Marriott, Falk Schreiber, et al. "Exploration of networks using overview+ detail with constraint-based cooperative layout". In: *IEEE transactions on visualization and computer graphics* 14.6 (2008), pp. 1293–1300 (cit. on pp. 16, 103).

[DMW08]   Tim Dwyer, Kim Marriott, and Michael Wybrow. "Topology preserving constrained graph layout". In: *International Symposium on Graph Drawing*. Springer. 2008, pp. 230–241 (cit. on pp. 16, 96, 97, 99, 122, 136).

[Dwy+13]   Tim Dwyer, Nathalie Henry Riche, Kim Marriott, and Christopher Mears. "Edge compression techniques for visualization of dense directed graphs". In: *IEEE transactions on visualization and computer graphics* 19.12 (2013), pp. 2596–2605 (cit. on pp. 15, 19).

[Efr+14]   Alon Efrat, Yifan Hu, Stephen G Kobourov, and Sergey Pupyrev. "MapSets: Visualizing embedded and clustered graphs". In: *International Symposium on Graph Drawing*. Springer. 2014, pp. 452–463 (cit. on p. 18).

[Fau+20]   Nicholas Faulkner, Kim Borg, Kun Zhao, and Liam Smith. "Measuring Social Inclusion: The Inclusive Australia Social Inclusion Index". In: (2020) (cit. on pp. 195, 196).

[Fek15]   Jean-Daniel Fekete. "Reorder. js: A javascript library to reorder tables and networks". In: *IEEE VIS 2015*. 2015 (cit. on p. 28).

[FMF12]   Andy Field, Jeremy Miles, and Zoë Field. *Discovering statistics using R*. 2012 (cit. on pp. 83, 143).

[For10]   Santo Fortunato. "Community detection in graphs". In: *Physics reports* 486.3-5 (2010), pp. 75–174 (cit. on pp. 21, 28, 66, 127, 174).

[Fuc+13]   Johannes Fuchs, Fabian Fischer, Florian Mansmann, Enrico Bertini, and Petra Isenberg. "Evaluation of alternative glyph designs for time series data in a small multiple setting". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2013, pp. 3237–3246 (cit. on pp. 37, 77).

[Gad+17]   Ujwal Gadiraju, Sebastian Möller, Martin Nöllenburg, et al. "Crowdsourcing versus the laboratory: towards human-centered experiments using the crowd". In: *Evaluation in the crowd. Crowdsourcing and human-centered experiments*. Springer, 2017, pp. 6–26 (cit. on p. 38).

[GHK09]   Emden R Gansner, Yifan Hu, and Stephen G Kobourov. "Gmap: Drawing graphs as maps". In: *International Symposium on Graph Drawing*. Springer. 2009, pp. 405–407 (cit. on pp. 18, 182, 189, 194).

[GKN04]   Emden R Gansner, Yehuda Koren, and Stephen North. "Graph drawing by stress majorization". In: *International Symposium on Graph Drawing*. Springer. 2004, pp. 239–250 (cit. on pp. 18, 30, 169, 202).

[Gar71]   Martin Gardner. *Martin Gardner's Sixth Book of Mathematical Diversions from Scientific American*. University of Chicago Press, 1971 (cit. on p. 44).

[GFC05]   Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. "On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis". In: *Information Visualization* 4.2 (2005), pp. 114–135 (cit. on p. 59).

[GN02]   Michelle Girvan and Mark EJ Newman. "Community structure in social and biological networks". In: *Proceedings of the national academy of sciences* 99.12 (2002), pp. 7821–7826 (cit. on pp. 6, 17, 191, 192).

[GAS17]   Alfred Gray, Elsa Abbena, and Simon Salamon. *Modern differential geometry of curves and surfaces with Mathematica®*. Chapman and Hall/CRC, 2017 (cit. on p. 44).

[GPK11]   Nicolas Greffard, Fabien Picarougne, and Pascale Kuntz. "Visual community detection: An evaluation of 2d, 3d perspective and 3d stereoscopic displays". In: *International Symposium on Graph Drawing*. Springer. 2011, pp. 215–225 (cit. on p. 19).

[Har99]   Robert L Harris. *Information graphics: A comprehensive illustrated reference*. Oxford University Press, USA, 1999 (cit. on p. 36).

[Har19]   The Growth Lab at Harvard University. *Product Space Networks*. Version V1. 2019 (cit. on pp. 197, 198).

[HB10]   Jeffrey Heer and Michael Bostock. "Crowdsourcing graphical perception: using mechanical turk to assess visualization design". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2010, pp. 203–212 (cit. on p. 74).

[HKA09]   Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. "Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2009, pp. 1303–1312 (cit. on p. 63).

[Hel+21]   Jouni Helske, Satu Helske, Matthew Cooper, Anders Ynnerman, and Lonni Besancon. "Can visualization alleviate dichotomous thinking Effects of visual representations on the cliff effect". In: *IEEE Transactions on Visualization and Computer Graphics* (2021) (cit. on pp. 161, 164, 179).

[Hen15]   Pontus Hennerdal. "Beyond the periphery: Child and adult understanding of world map continuity". In: *Annals of the Association of American Geographers* 105.4 (2015), pp. 773–790 (cit. on pp. 3, 22, 33, 46, 150–152, 164).

[Her+18]   Lukas Herman, Vojtech Jurik, Zdenek Stachon, et al. "Evaluation of user performance in interactive and static 3D maps". In: *ISPRS International Journal of Geo-Information* 7.11 (2018), p. 415 (cit. on p. 34).

[HAA16]   Florian Hruby, Mónica Chico Avelino, and Raymundo Montoya Ayala. "Journey to the end of the world Map–How edges of world maps shape the spatial mind". In: *Journal for Geographic Information Science* 1 (2016), pp. 314–323 (cit. on pp. 3, 22, 33, 34, 152).

[HGK10]   Yifan Hu, Emden R Gansner, and Stephen Kobourov. "Visualizing graphs and clusters as maps". In: *IEEE Computer Graphics and Applications* 30.6 (2010), pp. 54–66 (cit. on pp. 182, 189, 194).

[HEH09]   Weidong Huang, Peter Eades, and Seok-Hee Hong. "Measuring effectiveness of graph visualizations: A cognitive load perspective". In: *Information Visualization* 8.3 (2009), pp. 139–152 (cit. on pp. 98, 137, 169).

[HHE08]   Weidong Huang, Seok-Hee Hong, and Peter Eades. "Effects of crossing angles". In: *2008 IEEE Pacific Visualization Symposium*. IEEE. 2008, pp. 41–46 (cit. on p. 27).

[Hum19]   Transparent Statistics in Human–Computer Interaction Working Group. *Transparent Statistics Guidelines*. (Available at https://transparentstats.github.io/guidelines). June 2019 (cit. on pp. 156, 161, 164).

[HCD11]   Stéphane Huot, Olivier Chapuis, and Pierre Dragicevic. "TorusDesktop: pointing via the backdoor is sometimes shorter". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 829–838 (cit. on p. 62).

[HTE11]   Christophe Hurter, Alexandru Telea, and Ozan Ersoy. "Moleview: An attribute and structure-based semantic lens for large element-based plots". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2600–2609 (cit. on p. 19).

[IMO08]   Stephen Ingram, Tamara Munzner, and Marc Olano. "Glimmer: Multilevel MDS on the GPU". In: *IEEE Transactions on Visualization and Computer Graphics* 15.2 (2008), pp. 249–261 (cit. on pp. 29, 30).

[Ito00]   Masahiro Ito. "The characteristics of the torus self organizing map". In: *Proceedings 16th Fuzzy Systerm Symposium Akita, 2000*. Japan Society for Fuzzy and Systems. 2000 (cit. on pp. 41, 59, 62).

[Jai+19]   Raghvendra Jain, Sra Sontisirikit, Sopon Iamsirithaworn, and Helmut Prendinger. "Prediction of dengue outbreaks based on disease surveillance, meteorological and socio-economic data". In: *BMC infectious diseases* 19.1 (2019), pp. 1–16 (cit. on p. 36).

[Jan+10]   Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010 (cit. on p. 196).

[Jen+16]   Bernhard Jenny, Johannes Liem, Bojan Šavrič, and William M. Putman. "Interactive video maps: A year in the life of Earth's CO2". In: *Journal of Maps* 12.sup1 (Nov. 2016), pp. 36–42 (cit. on p. 34).

[Jen+17]   Bernhard Jenny, Bojan Šavrič, Nicholas D Arnold, Brooke E Marston, and Charles A Preppernau. "A guide to selecting map projections for world and hemisphere maps". In: *Choosing a map projection*. Springer, 2017, pp. 213–228 (cit. on pp. 3, 32, 33, 150, 151).

[KK+89]   Tomihisa Kamada, Satoru Kawai, et al. "An algorithm for drawing general undirected graphs". In: *Information processing letters* 31.1 (1989), pp. 7–15 (cit. on p. 16).

[Kar+20]   Alireza Karduni, Ryan Wesslen, Isaac Cho, and Wenwen Dou. "Du Bois Wrapped Bar Chart: Visualizing Categorical Data with Disproportionate Values". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–12 (cit. on p. 74).

[Kie+15]   Steve Kieffer, Tim Dwyer, Kim Marriott, and Michael Wybrow. "Hola: Human-like orthogonal network layout". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015), pp. 349–358 (cit. on pp. 16, 99, 103, 117, 189).

[KV07]   Toomas Kirt and Ene Vainik. "Comparison of the methods of self-organizing maps and multidimensional scaling in analysis of Estonian emotion concepts". In: *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*. 2007, pp. 113–120 (cit. on pp. 29, 31).

[Knu93]   Donald Ervin Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. Vol. 1. AcM Press New York, 1993 (cit. on pp. 21, 51).

[KL08]   Stephen G Kobourov and Matthew Landis. "Morphing planar graphs in spherical space". In: (2008) (cit. on pp. 20, 50).

[KW05]   Stephen G Kobourov and Kevin Wampler. "Non-Euclidean spring embedders". In: *IEEE Transactions on Visualization and Computer Graphics* 11.6 (2005), pp. 757–767 (cit. on pp. 20, 25).

[KK16]   William Kocay and Donald L Kreher. *Graphs, algorithms, and optimization*. Chapman and Hall/CRC, 2016 (cit. on pp. 7, 8, 25, 116, 187).

[Koh82]   Teuvo Kohonen. "Self-organized formation of topologically correct feature maps". In: *Biological cybernetics* 43.1 (1982), pp. 59–69 (cit. on pp. 31, 68, 202).

[Kwo+16]   Oh-Hyun Kwon, Chris Muelder, Kyungwon Lee, and Kwan-Liu Ma. "A study of layout, rendering, and interaction methods for immersive graph visualization". In: *IEEE transactions on visualization and computer graphics* 22.7 (2016), pp. 1802–1815 (cit. on pp. 8, 20, 21).

[Lam+11]   Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. "Empirical studies in information visualization: Seven scenarios". In: *IEEE transactions on visualization and computer graphics* 18.9 (2011), pp. 1520–1536 (cit. on p. 37).

[LRP95]   John Lamping, Ramana Rao, and Peter Pirolli. "A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1995, pp. 401–408 (cit. on p. 21).

[Lee+06]   Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. "Task Taxonomy for Graph Visualization". In: *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*. BELIV '06. Venice, Italy: ACM, 2006, pp. 1–5 (cit. on pp. 103, 140).

[Len16]   Russell V. Lenth. "Least-Squares Means: The *R* Package **lsmeans**". en. In: *Journal of Statistical Software* 69.1 (2016) (cit. on p. 156).

[Li+19]   Shan Li, Xun Li, Haohui Chen, and Wei Lang. "China's export evolution in the dynamic global product space from 2000 to 2011." In: *Current Science (00113891)* 117.3 (2019) (cit. on p. 197).

[Liu+14]   Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. "A survey on information visualization: recent advances and challenges". In: *The Visual Computer* 30.12 (2014), pp. 1373–1393 (cit. on pp. 1, 16, 28, 37, 38).

[LS20]   Jiawei Lu and Yain-Whar Si. "Clustering-based force-directed algorithms for 3D graph visualization". In: *The Journal of Supercomputing* 76.12 (2020), pp. 9654–9715 (cit. on pp. 16, 19).

[LCY19]   Yao Lu, Jukka Corander, and Zhirong Yang. "Doubly stochastic neighbor embedding on spheres". In: *Pattern Recognition Letters* 128 (2019), pp. 100–106 (cit. on pp. 30, 51, 53, 202).

[Man17]   Christopher Manning. *Spherical Force-Directed Layout*. 2017 (cit. on pp. 22, 50).

[Mar+12]   Rafael Messias Martins, Gabriel Faria Andery, Henry Heberle, et al. "Multidimensional projections for visual analysis of social networks". In: *Journal of Computer Science and Technology* 27.4 (2012), pp. 791–810 (cit. on pp. 30, 50, 202).

[Meu+13]   Wouter Meulemans, Nathalie Henry Riche, Bettina Speckmann, Basak Alper, and Tim Dwyer. "Kelpfusion: A hybrid set visualization technique". In: *IEEE transactions on visualization and computer graphics* 19.11 (2013), pp. 1846–1858 (cit. on pp. 182, 189, 194).

[Mis+07]   Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E Tarjan. "Clustering social networks". In: *International Workshop on Algorithms and Models for the Web-Graph*. Springer. 2007, pp. 56–67 (cit. on p. 127).

[MT01]   Bojan Mohar and Carsten Thomassen. *Graphs on surfaces*. Vol. 10. JHU press, 2001 (cit. on pp. 7, 23, 24, 187).

[Mun98]   Tamara Munzner. "Exploring large graphs in 3D hyperbolic space". In: *IEEE computer graphics and applications* 18.4 (1998), pp. 18–23 (cit. on p. 21).

[Mun14]   Tamara Munzner. *Visualization analysis and design*. CRC press, 2014 (cit. on pp. 1, 6, 16–18, 21, 28, 29, 37).

[New06]   Mark EJ Newman. "Modularity and community structure in networks". In: *Proceedings of the national academy of sciences* 103.23 (2006), pp. 8577–8582 (cit. on pp. 6, 17, 18, 127, 169, 174, 191).

[Nob+19]   Carolina Nobre, Miriah Meyer, Marc Streit, and Alexander Lex. "The state of the art in visualizing multivariate networks". In: *Computer Graphics Forum*. Vol. 38. 3. Wiley Online Library. 2019, pp. 807–832 (cit. on p. 200).

[Nob+20]   Carolina Nobre, Dylan Wootton, Lane Harrison, and Alexander Lex. "Evaluating multivariate network visualization techniques using a validated design and crowdsourcing approach". In: *Proceedings of the 2020 CHI conference on human factors in computing systems*. 2020, pp. 1–12 (cit. on pp. 37, 38).

[Obe20]   Camara K Obeng. *Export efficiency and diversification in Ghana*. 2020 (cit. on p. 197).

[OJK18]   Mershack Okoe, Radu Jianu, and Stephen Kobourov. "Node-link or adjacency matrices: Old question, new insights". In: *IEEE transactions on visualization and computer graphics* 25.10 (2018), pp. 2940–2952 (cit. on pp. 18, 27–29, 38, 156, 191, 193, 194).

[OG97]   Chong Jin Ong and Elmer G Gilbert. "The Gilbert-Johnson-Keerthi distance algorithm: A fast version for incremental motions". In: *Proceedings of International Conference on Robotics and Automation*. Vol. 2. IEEE. 1997, pp. 1183–1189 (cit. on p. 137).

[Pac19]   Python Package. *NetworkX graph library*. `https://networkx.github.io`. Last accessd: September thirteenth, 2020. 2019 (cit. on pp. 101, 127).

[PS18]   Stefan Palan and Christian Schitter. "Prolific. ac—A subject pool for online experiments". In: *Journal of Behavioral and Experimental Finance* 17 (2018), pp. 22–27 (cit. on pp. 38, 80, 155).

[Pal13]   Julie Pallant. *SPSS survival manual*. McGraw-hill education (UK), 2013 (cit. on p. 157).

[PM17]   Sofia Papazoglou and Kostas Mylonas. "An examination of alternative multidimensional scaling techniques". In: *Educational and Psychological Measurement* 77.3 (2017), pp. 429–448 (cit. on p. 53).

[Pee+17]   Eyal Peer, Laura Brandimarte, Sonam Samat, and Alessandro Acquisti. "Beyond the Turk: Alternative platforms for crowdsourcing behavioral research". In: *Journal of Experimental Social Psychology* 70 (2017), pp. 153–163 (cit. on pp. 38, 80, 155).

[Pei79]   Charles S Peirce. "A quincuncial projection of the sphere". In: *American Journal of Mathematics* 2.4 (1879), pp. 394–396 (cit. on pp. 21, 44, 46, 49).

[Per+20]   Scott Perry, Mason Sun Yin, Kathryn Gray, and Stephen Kobourov. "Drawing Graphs on the Sphere". In: *Proceedings of the International Conference on Advanced Visual Interfaces*. 2020, pp. 1–9 (cit. on pp. 7, 18, 20, 22, 23, 50, 168, 170).

[Pur97]   Helen Purchase. "Which aesthetic has the greatest effect on human understanding?" In: *International Symposium on Graph Drawing*. Springer. 1997, pp. 248–261 (cit. on p. 27).

[Pur12]   Helen C Purchase. *Experimental human-computer interaction: a practical guide with visual examples*. Cambridge University Press, 2012 (cit. on pp. 38, 79).

[Pur02]   Helen C Purchase. "Metrics for graph drawing aesthetics". In: *Journal of Visual Languages & Computing* 13.5 (2002), pp. 501–516 (cit. on pp. 94, 136, 137, 169).

[PCA02]   Helen C Purchase, David Carrington, and Jo-Anne Allder. "Empirical evaluation of aesthetics-based graph layout". In: *Empirical Software Engineering* 7.3 (2002), pp. 233–255 (cit. on p. 27).

[QE00]   Aaron Quigley and Peter Eades. "Fade: Graph drawing, clustering, and visual abstraction". In: *International Symposium on Graph Drawing*. Springer. 2000, pp. 197–210 (cit. on p. 125).

[Rit99]   Helge Ritter. "Self-organizing maps on non-euclidean spaces". In: *Kohonen maps*. Elsevier, 1999, pp. 97–109 (cit. on p. 41).

[RS22]   Roads and Maritime Services. *https://roads-waterways.transport.nsw.gov.au*. 2015 (accessed 15 May, 2022) (cit. on p. 61).

[Rod20]   Dario Rodighiero. "Drawing Network Visualizations on a Continuous, Spherical Surface". In: *2020 24th International Conference Information Visualisation (IV)*. IEEE. 2020, pp. 573–580 (cit. on pp. 7, 18, 20, 22, 46, 50, 167, 168).

[Rot21]   Camille Roth. "Socio-Semantic Systems". PhD thesis. Sorbonne Université, 2021 (cit. on pp. 6, 200).

[RB20]   Camille Roth and Nikita Basov. "The socio-semantic space of John Mohr". In: *Poetics* 78 (2020), p. 101437 (cit. on pp. 6, 200).

[RSH21]   Camille Roth, Jonathan St-Onge, and Katrin Herms. "Quoting is not Citing: Disentangling Affiliation and Interaction on Twitter". In: *International Conference on Complex Networks and Their Applications*. Springer. 2021, pp. 705–717 (cit. on p. 200).

[SED18]   Bahador Saket, Alex Endert, and Çağatay Demiralp. "Task-based effectiveness of basic visualizations". In: *IEEE transactions on visualization and computer graphics* 25.7 (2018), pp. 2505–2512 (cit. on p. 37).

[SSK14]   Bahador Saket, Paolo Simonetto, and Stephen Kobourov. "Group-level graph visualization taxonomy". In: *arXiv preprint arXiv:1403.7421* (2014) (cit. on pp. 127, 140).

[Sak+14]   Bahador Saket, Paolo Simonetto, Stephen Kobourov, and Katy Börner. "Node, node-link, and node-link-group diagrams: An evaluation". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2231–2240 (cit. on pp. 6, 18, 27, 191, 194).

[Sar15]   Peter Sarlin. "Data and dimension reduction for visual financial performance analysis". In: *Information Visualization* 14.2 (2015), pp. 148–167 (cit. on p. 31).

[Sat+21]   Kadek Ananta Satriadi, Barrett Ens, Tobias Czauderna, Maxime Cordeil, and Bernhard Jenny. "Quantitative Data Visualisation on Virtual Globes". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–14 (cit. on pp. 80, 155).

[Šav+15]   Bojan Šavric, Bernhard Jenny, Denis White, and Daniel R Strebe. "User preferences for world map projections". In: *Cartography and Geographic Information Science* 42.5 (2015), pp. 398–409 (cit. on pp. 33, 34, 150, 151, 161, 162).

[ŠPJ19]   Bojan Šavrič, Tom Patterson, and Bernhard Jenny. "The equal earth map projection". In: *International Journal of Geographical Information Science* 33.3 (2019), pp. 454–465 (cit. on p. 33).

[Sch13]   Marcus Schaefer. "The graph crossing number and its variants: A survey". In: *The Electronic Journal of Combinatorics* 1000 (2013), pp. 21–22 (cit. on p. 24).

[SS19]   Thomas Schäfer and Marcus A Schwarz. "The meaningfulness of effect sizes in psychological research: Differences between sub-disciplines and the impact of potential biases". In: *Frontiers in Psychology* 10 (2019), p. 813 (cit. on pp. 161, 164).

[Sch+21]   Sarah Schöttler, Yalong Yang, Hanspeter Pfister, and Benjamin Bach. "Visualizing and interacting with geospatial networks: A survey and design space". In: *Computer Graphics Forum*. Wiley Online Library. 2021 (cit. on p. 32).

[Sch+19]   Michail Schwab, Sicheng Hao, Olga Vitek, et al. "Evaluating pan and zoom timelines and sliders". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–12 (cit. on pp. 74, 79, 80).

[SPB21]   Michail Schwab, Aditeya Pandey, and Michelle A Borkin. "Evaluation of 1D Selection Techniques for Mobile Visualizations". In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–7 (cit. on p. 74).

[SH76]   Michael Ian Shamos and Dan Hoey. "Geometric intersection problems". In: *17th Annual Symposium on Foundations of Computer Science (sfcs 1976)*. IEEE. 1976, pp. 208–215 (cit. on p. 125).

[SW65]   S Shaphiro and M Wilk. "An analysis of variance test for normality". In: *Biometrika* 52.3 (1965), pp. 591–611 (cit. on pp. 83, 143).

[SH87]   David Simkin and Reid Hastie. "An information-processing analysis of graph perception". In: *Journal of the American Statistical Association* 82.398 (1987), pp. 454–465 (cit. on p. 37).

[SH11]   Alexander James Gaspar Simoes and César A Hidalgo. "The economic complexity observatory: An analytical tool for understanding the dynamics of economic development". In: *Workshops at the twenty-fifth AAAI conference on artificial intelligence*. 2011 (cit. on p. 197).

[Sim+11]   Paolo Simonetto, Daniel Archambault, David Auber, and Romain Bourqui. "ImPrEd: An Improved Force-Directed Algorithm that Prevents Nodes from Crossing Edges". In: *Computer Graphics Forum*. Vol. 30. 3. Wiley Online Library. 2011, pp. 1071–1080 (cit. on p. 99).

[Sin21]   Metro of Singapore. *https://mapa-metro.com/en/Singapore/Singapore/Singapore-MRT-map.htm*. 2010 (accessed 15 May, 2021) (cit. on p. 58).

[Sny97]   John P Snyder. *Flattening the earth: two thousand years of map projections*. University of Chicago Press, 1997 (cit. on pp. 1, 22, 32, 33).

[Sny87]   John Parr Snyder. *Map projections–A working manual*. Vol. 1395. US Government Printing Office, 1987 (cit. on pp. 1, 34, 155).

[Squ+05]   David McG Squire et al. "Visualization of cluster changes by comparing Self-Organizing Maps". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2005, pp. 410–419 (cit. on pp. 31, 202).

[STT81]   Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. "Methods for visual understanding of hierarchical system structures". In: *IEEE Transactions on Systems, Man, and Cybernetics* 11.2 (1981), pp. 109–125 (cit. on pp. 16, 117, 189).

[Suh+19]   Ashley Suh, Mustafa Hajij, Bei Wang, Carlos Scheidegger, and Paul Rosen. "Persistent homology guided force-directed graph layouts". In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2019), pp. 697–707 (cit. on p. 16).

[TSA14]   Justin Talbot, Vidya Setlur, and Anushka Anand. "Four experiments on the perception of bar charts". In: *IEEE transactions on visualization and computer graphics* 20.12 (2014), pp. 2152–2160 (cit. on pp. 4, 35, 74, 76).

[Tho+03]   Alun Thomas, Rob Cannings, Nicholas AM Monk, and Chris Cannings. *On the structure of protein–protein interaction networks*. 2003 (cit. on pp. 15, 17).

[TU20]   Michael C Thrun and Alfred Ultsch. "Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems". In: *Data in Brief* (2020), p. 105501 (cit. on pp. 70, 201, 202).

[TS08]   Christian Tominski and Heidrun Schumann. "Enhanced interactive spiral display". In: *SIGRAD 2008. The Annual SIGRAD Conference Special Theme: Interaction; November 27-28; 2008 Stockholm; Sweden*. 034. Linköping University Electronic Press. 2008, pp. 53–56 (cit. on p. 36).

[Tom+12]   Christian Tominski, Heidrun Schumann, Gennady Andrienko, and Natalia Andrienko. "Stacking-based visualization of trajectory attribute data". In: *IEEE Transactions on visualization and Computer Graphics* 18.12 (2012), pp. 2565–2574 (cit. on p. 36).

[Tro22]   P. Troshenkov. *https://www.kaggle.com/pavetr/nypdcollisions/version/1*. 2017 (accessed 15 May, 2022) (cit. on pp. 3, 59, 63, 75, 79, 204).

[Ult03]   Alfred Ultsch. "Maps for the visualization of high-dimensional data spaces". In: *Proc. Workshop on Self organizing Maps*. 2003, pp. 225–230 (cit. on pp. 26, 31, 41, 59, 62, 68, 69, 203).

[Van01]   Gino Van Den Bergen. "Proximity queries and penetration depth computation on 3d game objects". In: *Game developers conference*. Vol. 170. 2001 (cit. on p. 138).

[VH08]   Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008) (cit. on p. 202).

[VW08]   Frank Van Ham and Martin Wattenberg. "Centrality based visualization of small world graphs". In: *Computer Graphics Forum*. Vol. 27. 3. Wiley Online Library. 2008, pp. 975–982 (cit. on p. 6).

[Van08]   Jarke J Van Wijk. "Unfolding the earth: myriahedral projections". In: *The Cartographic Journal* 45.1 (2008), pp. 32–42 (cit. on pp. 1, 33).

[VBW17]   Corinna Vehlow, Fabian Beck, and Daniel Weiskopf. "Visualizing group structures in graphs: A survey". In: *Computer Graphics Forum*. Vol. 36. 6. Wiley Online Library. 2017, pp. 201–225 (cit. on p. 18).

[Wal+19]   Manuela Waldner, Alexandra Diehl, Denis Gračanin, et al. "A comparison of radial and linear charts for visualizing daily patterns". In: *IEEE transactions on visualization and computer graphics* 26.1 (2019), pp. 1033–1042 (cit. on pp. 4, 36, 37, 63, 74, 77–80, 89, 90).

[Wal00]   Chris Walshaw. "A multilevel algorithm for force-directed graph drawing". In: *International Symposium on Graph Drawing*. Springer. 2000, pp. 171–182 (cit. on p. 125).

[Wan+15]   Yong Wang, Qiaomu Shen, Daniel Archambault, et al. "Ambiguityvis: Visualization of ambiguity in graph layouts". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015), pp. 359–368 (cit. on pp. 6, 18, 191).

[Wan+18]   Yunhai Wang, Yanyan Wang, Haifeng Zhang, et al. "Structure-aware fisheye views for efficient large graph exploration". In: *IEEE transactions on visualization and computer graphics* 25.1 (2018), pp. 566–575 (cit. on p. 183).

[War19]   Colin Ware. *Information visualization: perception for design*. Morgan Kaufmann, 2019 (cit. on p. 1).

[WM08]   Colin Ware and Peter Mitchell. "Visualizing graphs in three dimensions". In: *ACM Transactions on Applied Perception (TAP)* 5.1 (2008), pp. 1–15 (cit. on p. 19).

[War+02]   Colin Ware, Helen Purchase, Linda Colpoys, and Matthew McGill. "Cognitive measurements of graph aesthetics". In: *Information Visualization* 1.2 (2002), pp. 103–110 (cit. on pp. 94, 136).

[WS98]   Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks". In: *nature* 393.6684 (1998), pp. 440–442 (cit. on pp. 6, 17, 174, 191).

[15]   *WebCoLa: Constraint-Based Layout in the Browser*. 2015 (cit. on pp. 96, 100, 102, 135).

[Wil49]   Evan James Williams. "Experimental designs balanced for the estimation of residual effects of treatments". In: *Australian Journal of Chemistry* 2.2 (1949), pp. 149–168 (cit. on p. 155).

[WT06a]   Yingxin Wu and Masahiro Takatsuka. "Spherical self-organizing map using efficient indexed geodesic data structure". In: *Neural Networks* 19.6-7 (2006), pp. 900–910 (cit. on pp. 31, 41, 51, 52, 202, 203).

[WT06b]   Yingxin Wu and Masahiro Takatsuka. "Visualizing multivariate network on the surface of a sphere". In: *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation-Volume 60*. Citeseer. 2006, pp. 77–83 (cit. on pp. 22, 31).

[Xu+12]   Kai Xu, Chris Rooney, Peter Passmore, Dong-Han Ham, and Phong H Nguyen. "A user study on curved edges in graph visualization". In: *IEEE transactions on visualization and computer graphics* 18.12 (2012), pp. 2449–2456 (cit. on p. 174).

[Yan+19]   Yalong Yang, Tim Dwyer, Bernhard Jenny, et al. "Origin-Destination Flow Maps in Immersive Environments". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 693–703 (cit. on p. 35).

[Yan+18]   Yalong Yang, Bernhard Jenny, Tim Dwyer, et al. "Maps and globes in virtual reality". In: *Computer Graphics Forum*. Vol. 37. 3. Wiley Online Library. 2018, pp. 427–438 (cit. on pp. 20, 21, 151–155, 165).

[Yog+18]   Vahan Yoghourdjian, Tim Dwyer, Karsten Klein, Kim Marriott, and Michael Wybrow. "Graph thumbnails: Identifying and comparing multiple graphs at a glance". In: *IEEE Transactions on Visualization and Computer Graphics* 24.12 (2018), pp. 3081–3095 (cit. on pp. 6, 19, 117, 189).

[Yog+20]   Vahan Yoghourdjian, Yalong Yang, Tim Dwyer, et al. "Scalability of network visualisation from a cognitive load perspective". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2020), pp. 1677–1687 (cit. on pp. 6, 27, 156).

[Yu14]   Jiahua Yu. "A Practical Torus Embedding Algorithm and Its Implementation". PhD thesis. Applied Sciences: School of Computing Science, 2014 (cit. on p. 24).

[ZT99]   Jeff Zacks and Barbara Tversky. "Bars and lines: A study of graphic communication". In: *Memory & cognition* 27.6 (1999), pp. 1073–1079 (cit. on p. 74).

[Zha+22]   Yidan Zhang, Barrett Ens, Kadek Ananta Satriadi, Arnaud Prouzeau, and Sarah Goodwin. "TimeTables: Embodied Exploration of Immersive Spatio-Temporal Data". In: *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2022, pp. 599–605 (cit. on p. 36).

[Zha+11]   Jian Zhao, Fanny Chevalier, Emmanuel Pietriga, and Ravin Balakrishnan. "Exploratory analysis of time-series with chronolenses". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2422–2431 (cit. on p. 36).

[ZPG18]   Jonathan X Zheng, Samraat Pawar, and Dan FM Goodman. "Graph drawing by stochastic gradient descent". In: *IEEE transactions on visualization and computer graphics* 25.9 (2018), pp. 2738–2748 (cit. on pp. 6, 16, 30, 95, 119, 121, 122, 170).

Appendix
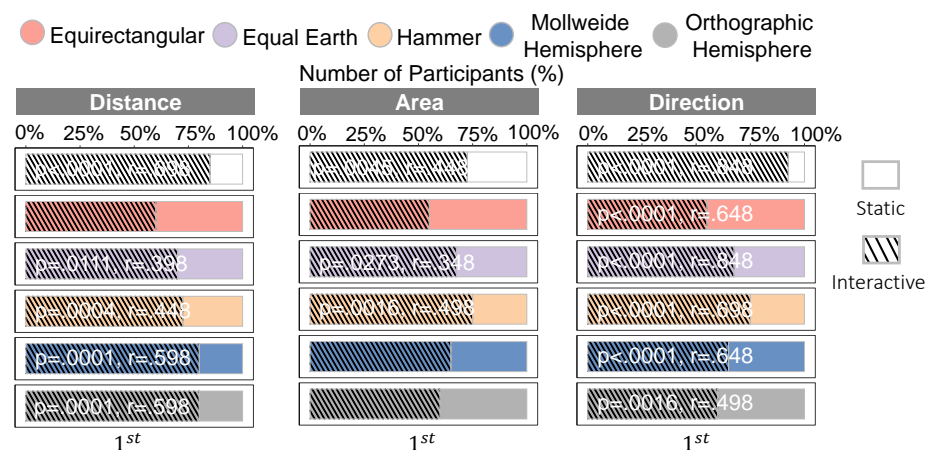
Fig. A.1.: Study 5: User preference rank of interactivity for overall (top) or within each map projection. Statistical significance results with $p < 0.05$ and effect sizes of Cohen's r are shown. x-axis shows the number of participants in percentage.

Fig. A.2.: Matrices of pairwise effect size results of ERROR and TIME of Study 5 for Distance task. Values indicate effect size results of Cohen's r and Cohen's d [Coh13] for ERROR and TIME, respectively.

Area Comparison

Error rate

All responses

	IMH	IHA	IEE	IER	SOH	SMH	SHA	SEE	SER
SEE									-0.346
SHA								0.173	-0.212
SMH							-0.117	0.051	-0.312
SOH						0.041	0.123	0.2	-0.138
IER					-0.216	-0.116	-0.221	-0.022	-0.445 (***)
IEE				-0.258	-0.426 (***)	-0.382	-0.444 (*)	-0.322 (**)	-0.536 (***)
IHA			-0.01	-0.289	-0.442 (***)	-0.37	-0.466 (***)	-0.243	-0.551 (***)
IMH		-0.05	-0.07	-0.351 (*)	-0.463 (***)	-0.453 (*)	-0.474 (*)	-0.334	-0.576 (***)
IOH	0.025	-0.04	-0.05	-0.287	-0.477 (***)	-0.382	-0.432	-0.319	-0.592 (***)

Easy

	IMH	IHA	IEE	IER	SOH	SMH	SHA	SEE	SER
SEE									-0.377 (*)
SHA								0.207	-0.307
SMH							-0.268	-0.022	-0.459 (*)
SOH						0.339	0.181	0.304	-0.066
IER					-0.353	-0.089	-0.317	-0.056	-0.555 (***)
IEE				-0.35	-0.524 (**)	-0.428	-0.528 (**)	-0.402 (*)	-0.613 (***)
IHA			0.19	-0.189	-0.497 (*)	-0.297	-0.463 (**)	-0.189	-0.593 (***)
IMH		-0.19	-0.01	-0.324	-0.538 (***)	-0.409 (*)	-0.549 (**)	-0.376	-0.585 (***)
IOH	0.16	0.031	0.145	-0.188	0.52 (*)	-0.235	-0.378	-0.194	-0.532 (***)

Hard

	IMH	IHA	IEE	IER	SOH	SMH	SHA	SEE	SER
SEE									-0.315
SHA								0.118	-0.102
SMH							0.007	0.12	-0.139
SOH						-0.108	-0.076	0.061	-0.193
IER					-0.038	-0.158	-0.113	-0.011	-0.319 (*)
IEE				-0.193	-0.261	-0.345	-0.33	-0.243	-0.434
IHA			-0.15	-0.38	-0.379	-0.436 (*)	-0.472 (**)	-0.29	-0.504 (**)
IMH		0.047	-0.1	-0.38	-0.36	-0.485 (**)	-0.394	-0.281	-0.576 (*)
IOH	-0.11	-0.03	-0.22	-0.41	-0.422 (**)	-0.48 (*)	-0.478	-0.434	-0.649 (**)

Time

All responses

	IMH	IHA	IEE	IER	SOH	SMH	SHA	SEE	SER
SEE									-0.125
SHA								0.035	-0.084
SMH							-0.025	0.013	-0.121
SOH						-0.0871	-0.102	-0.066	-0.204
IER					1.364 (***)	1.296 (***)	1.206 (***)	1.257 (***)	1.177 (***)
IEE				-0.095	1.522 (***)	1.436 (***)	1.302 (***)	1.37 (***)	1.28 (***)
IHA			0.295 (**)	0.165	1.776 (***)	1.693 (***)	1.554 (***)	1.622 (***)	1.539 (***)
IMH		-0.13	0.151	0.042	1.557 (***)	1.48 (***)	1.362 (***)	1.423 (***)	1.339 (***)
IOH	0.027	-0.114	0.195	0.07	1.802 (***)	1.71 (***)	1.547 (***)	1.624 (***)	1.536 (***)

Easy

	IMH	IHA	IEE	IER	SOH	SMH	SHA	SEE	SER
SEE									-0.076
SHA								0.004	-0.075
SMH							-0.121	-0.111	-0.205
SOH						0.115	-0.012	-0.007	-0.091
IER					1.279 (***)	1.385 (***)	1.237 (***)	1.212 (***)	1.184 (***)
IEE				-0.107	1.447 (***)	1.597 (***)	1.379 (***)	1.337 (***)	1.316 (***)
IHA			0.292	0.144	1.686 (***)	1.83 (***)	1.617 (***)	1.572 (***)	1.557 (***)
IMH		-0.248	0.02	-0.082	1.308 (***)	1.431 (***)	1.258 (***)	1.227 (***)	1.2 (***)
IOH	0.15	-0.109	0.187	0.051	1.629 (***)	1.78 (***)	1.557 (***)	1.51 (***)	1.495 (***)

Hard

	IMH	IHA	IEE	IER	SOH	SMH	SHA	SEE	SER
SEE									-0.175
SHA								0.065	-0.092
SMH							0.057	0.138	-0.04
SOH						-0.288	-0.187	-0.133	-0.324
IER					1.437 (***)	1.198 (***)	1.163 (***)	1.287 (***)	1.155 (***)
IEE				-0.083	1.582 (***)	1.286 (***)	1.226 (***)	1.386 (***)	1.232 (***)
IHA			0.294	0.183	1.849 (***)	1.559 (***)	1.485 (***)	1.653 (***)	1.504 (***)
IMH		-0.018	0.274	0.165	1.816 (***)	1.529 (***)	1.459 (***)	1.623 (***)	1.476 (***)
IOH	-0.097	-0.117	0.201	0.089	1.976 (***)	1.625 (***)	1.52 (***)	1.728 (***)	1.559 (***)

Legend (Error rate)

Row A – Column B: signed Effect size

+: A is *worse than* B in terms of mean error rate
-: A is *better than* B in terms of mean error rate
Statistically significant results are indicated by $p < .05 (*)$, $p < .01 (**)$, and $p < .001 (***)$

SER: Static Equirectangular
SEE: Static Equal Earth
SHA: Static Hammer
SMH: Static Mollweide Hemisphere
SOH: Static Orthographic Hemisphere

IER: Interactive Equirectangular
IEE: Interactive Equal Earth
IHA: Interactive Hammer
IMH: Interactive Mollweide Hemisphere
IOH: Interactive Orthographic Hemisphere

negligible effect / *Small effect* / *Moderate effect* / *Large effect*

Legend (Time)

Row A – Column B: signed Effect size

+: A is *worse than* B in terms of mean completion time
-: A is *better than* B in terms of mean completion time
Statistically significant results are indicated by $p < .05 (*)$, $p < .01 (**)$, and $p < .001 (***)$

SER: Static Equirectangular
SEE: Static Equal Earth
SHA: Static Hammer
SMH: Static Mollweide Hemisphere
SOH: Static Orthographic Hemisphere

IER: Interactive Equirectangular
IEE: Interactive Equal Earth
IHA: Interactive Hammer
IMH: Interactive Mollweide Hemisphere
IOH: Interactive Orthographic Hemisphere

negligible effect / *Small effect* / *Moderate effect* / *Large effect*

Fig. A.3.: Matrices of pairwise effect size results of ERROR and TIME of Study 5 for Area task. Values indicate effect size results of Cohen's r and Cohen's d [Coh13] for ERROR and TIME, respectively.

Direction Estimation

Error rate

All responses

	IMH	IHA	IEE	IER	SOH	SMH	SHA	SEE	SER
SEE									-0.087
SHA								-0.181	-0.346
SMH							0.111	-0.076	-0.198
SOH						0.082	0.24	-0.017	-0.127
IER					-0.551 (**)	-0.526 (*)	-0.425	-0.585	-0.571 (**)
IEE				-0.242	-0.665 (**)	-0.648 (***)	-0.566 (**)	-0.64 (**)	-0.641 (***)
IHA			0.058	-0.147	-0.677 (**)	-0.573 (*)	-0.575 (*)	-0.555 (**)	-0.695 (***)
IMH		-0.02	0.025	-0.19	-0.62 (***)	-0.591 (**)	-0.509 (*)	-0.617 (*)	-0.645 (**)
IOH	-0.33	-0.29	-0.28	-0.443 (*)	-0.786 (***)	-0.721 (***)	-0.731 (***)	-0.759 (***)	-0.777 (***)

Time

All responses

	IMH	IHA	IEE	IER	SOH	SMH	SHA	SEE	SER
SEE									0.207
SHA								-0.172	0.045
SMH							0.269 (*)	0.095	0.299 (**)
SOH						0.012	0.315 (**)	0.118	0.345 (***)
IER					0.917 (***)	0.855 (***)	1.223 (***)	0.979 (***)	1.22 (***)
IEE				-0.087	0.758 (***)	0.716 (***)	1.036 (***)	0.827 (***)	1.042 (***)
IHA			-0.048	-0.146	0.778 (***)	0.725 (***)	1.089 (***)	0.849 (***)	1.09 (***)
IMH		0.306 (**)	0.24 (*)	0.167	1.021 (***)	0.962 (***)	1.301 (***)	1.078 (***)	1.299 (***)
IOH	-0.233	-0.095	0.035	-0.06	0.928 (***)	0.855 (***)	1.266 (***)	0.991 (***)	1.257 (***)

Legend (Error rate)

Row A – Column B: signed Effect size

+: A is *worse than* B in terms of mean error rate
-: A is *better than* B in terms of mean error rate
Statistically significant results are indicated by $p < .05 (*)$, $p < .01 (**)$, and $p < .001 (***)$

SER: Static Equirectangular
SEE: Static Equal Earth
SHA: Static Hammer
SMH: Static Mollweide Hemisphere
SOH: Static Orthographic Hemisphere

IER: Interactive Equirectangular
IEE: Interactive Equal Earth
IHA: Interactive Hammer
IMH: Interactive Mollweide Hemisphere
IOH: Interactive Orthographic Hemisphere

negligible effect / *Small effect* / *Moderate effect* / *Large effect*

Legend (Time)

Row A – Column B: signed Effect size

+: A is *worse than* B in terms of mean completion time
-: A is *better than* B in terms of mean completion time
Statistically significant results are indicated by $p < .05 (*)$, $p < .01 (**)$, and $p < .001 (***)$

SER: Static Equirectangular
SEE: Static Equal Earth
SHA: Static Hammer
SMH: Static Mollweide Hemisphere
SOH: Static Orthographic Hemisphere

IER: Interactive Equirectangular
IEE: Interactive Equal Earth
IHA: Interactive Hammer
IMH: Interactive Mollweide Hemisphere
IOH: Interactive Orthographic Hemisphere

negligible effect / *Small effect* / *Moderate effect* / *Large effect*

Fig. A.4.: Matrices of pairwise effect size results of ERROR and TIME of Study 5 for Direction task. Values indicate effect size results of Cohen's r and Cohen's d [Coh13] for ERROR and TIME, respectively.

Fig. A.5.: Matrices of pairwise effect size results of subjective user rank of Study 5 map projections within the *static* (left) and *interactive* (right) groups for three tested tasks. Values indicate effect sizes results of Cohen's r.

Fig. A.6.: Study 6: Statistically significant results of ERROR and TIME for Cluster Number task (left) and Shortest Path Number task (right). Effect size results of Cohen's r and Cohen's d for ERROR and TIME are presented along with the arrows.

Cluster Number / Shortest Path Number — Error rate and Time matrices

Cluster Number — Error rate

All Responses

A \ B	Torus	Flat	EE
Flat	-0.654 (***)		
EE	0.098	-0.657 (***)	
OH	0.366 (**)	0.418 (*)	-0.452 (*)

Easy

A \ B	Torus	Flat	EE
Flat	-0.744 (***)		
EE	0.005	-0.770 (***)	
OH	0.683 (***)	0.668 (***)	-0.459

Hard

A \ B	Torus	Flat	EE
Flat	-0.559 (***)		
EE	0.166	-0.539 (***)	
OH	0.098	0.226	-0.439 (***)

Small

A \ B	Torus	Flat	EE
Flat	-0.491 (***)		
EE	-0.039	-0.594 (***)	
OH	0.216	0.195	-0.421 (***)

Large

A \ B	Torus	Flat	EE
Flat	-0.772 (***)		
EE	0.231	-0.719 (***)	
OH	0.509 (**)	0.617 (***)	-0.490 (*)

Cluster Number — Time

All Responses

A \ B	Torus	Flat	EE
Flat	0.458 (***)		
EE	-0.283 (***)	0.198 (***)	
OH	0.465 (***)	0.19 (***)	0.624 (***)

Easy

A \ B	Torus	Flat	EE
Flat	0.594 (***)		
EE	-0.415 (***)	0.202 (*)	
OH	0.633 (***)	0.228 (*)	0.796 (***)

Hard

A \ B	Torus	Flat	EE
Flat	0.343 (***)		
EE	-0.166 (*)	0.195 (***)	
OH	0.320 (***)	0.158	0.479 (***)

Small

A \ B	Torus	Flat	EE
Flat	0.361 (***)		
EE	-0.205 (**)	0.169	
OH	0.495 (***)	0.284 (***)	0.635 (***)

Large

A \ B	Torus	Flat	EE
Flat	1.331 (***)		
EE	-0.254 (***)	1.096 (***)	
OH	0.537 (***)	0.8284	0.670 (***)

Shortest Path Number — Error rate

All Responses

A \ B	Torus	Flat	EE
Flat	0.160		
EE	0.243 (**)	0.392 (**)	
OH	0.112	0.326 (**)	0.422 (***)

Easy

A \ B	Torus	Flat	EE
Flat	0.147		
EE	0.561 (***)	0.624 (***)	
OH	0.099	0.559 (**)	0.629 (***)

Hard

A \ B	Torus	Flat	EE
Flat	0.161		
EE	-0.108	0.058	
OH	0.120	0	0.132

Shortest Path Number — Time

All Responses

A \ B	Torus	Flat	EE
Flat	0.436 (***)		
EE	0.061	0.473 (***)	
OH	0.073	0.142	0.582 (***)

Easy

A \ B	Torus	Flat	EE
Flat	0.070		
EE	0.173	0.266	
OH	0.059	0.226	0.322

Hard

A \ B	Torus	Flat	EE
Flat	0.872 (***)		
EE	-0.035	0.664 (***)	
OH	0.091	0.070	0.862 (***)

Legend

Error

Row A – Column B: signed Effect size

+: A is *worse than* B in terms of mean error rate
-: A is *better than* B in terms of mean error rate
Statistically significant results are indicated by $p < .05$(*), $p < .01$(**), and $p < .001$(***)

Time

+: A is *worse than* B in terms of mean completion time
-: A is *better than* B in terms of mean completion time
Statistically significant results are indicated by $p < .05$(*), $p < .01$(**), and $p < .001$(***)

EE: Equal Earth
OH: Orthographic Hemisphere

negligible effect / *Small effect* / *Moderate effect* / *Large effect*

Fig. A.7.: Matrices of pairwise effect size results of ERROR and TIME of Study 6 for Cluster Number task (left) and Shortest Path Number task (right). Values indicate effect size results of Cohen's r and Cohen's d [Coh13] for ERROR and TIME, respectively.

Rank — PREF matrices
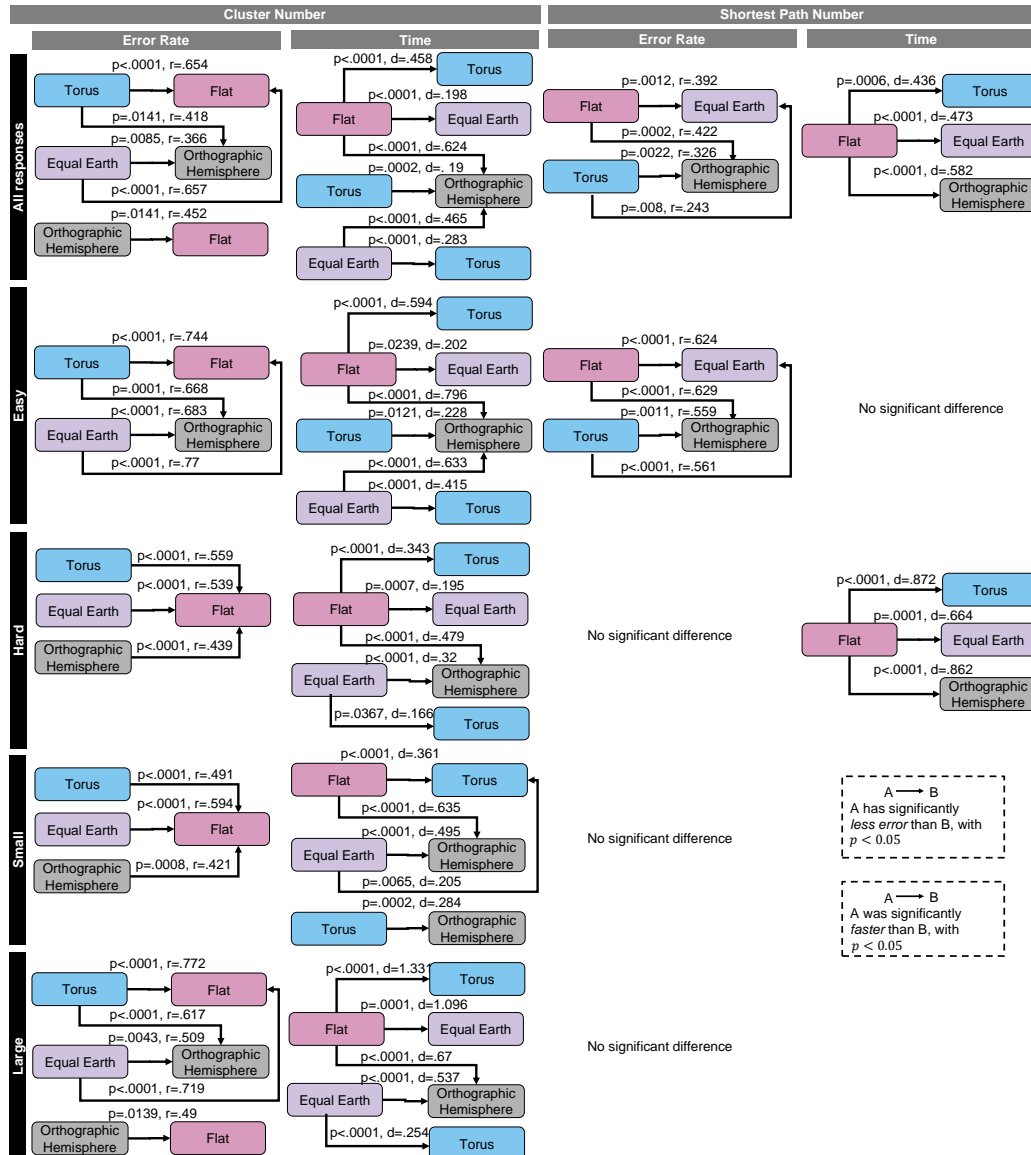
Cluster Number

A \ B	Torus	Flat	EE
Flat	-0.513 (**)		
EE	0.161	-0.390 (*)	
OH	0.641 (***)	0.615 (***)	0.196

Shortest Path Number

A \ B	Torus	Flat	EE
Flat	-0.190		
EE	0.617 (***)	0.461 (**)	
OH	-0.077	0.629 (***)	0.474 (**)

Legend

Row A – Column B: signed Effect size

+: A is *less preferred over* B
-: A is *more preferred over* B
Statistically significant results are indicated by $p < .05$(*), $p < .01$(**), and $p < .001$(***)

EE: Static Equal Earth
OH: Static Orthographic Hemisphere

negligible effect / *Small effect* / *Moderate effect* / *Large effect*

Fig. A.8.: Matrices of pairwise effect size results of PREF of Study 6 for Cluster Number task (left) and Shortest Path Number task (right). Values indicate effect size results of Cohen's r.