Euclidean, Spherical and Hyperbolic Networks: an empirical study of matching graph structure to best visualizations

Category: Theoretical and Empirical

Paper Type: Evaluation

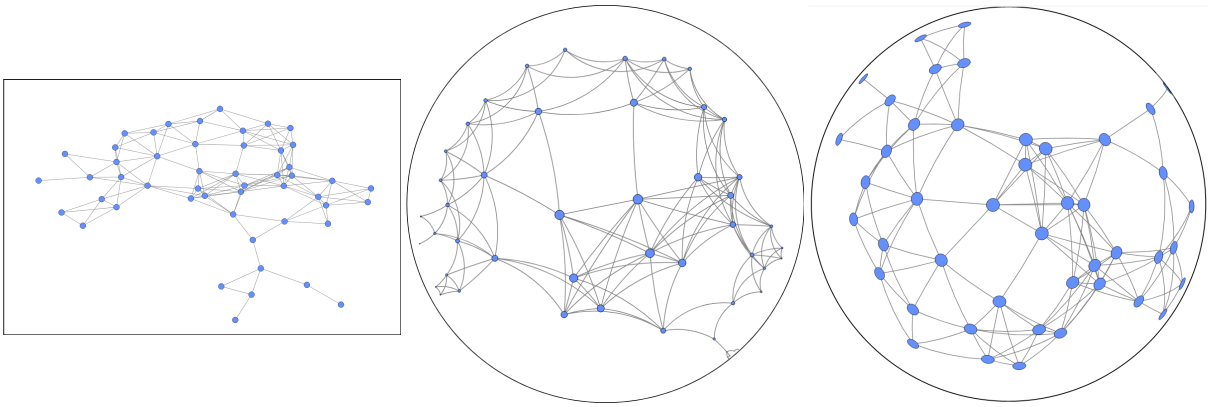


Fig. 1: The three conditions present in our study: (left) Euclidean, (middle) hyperbolic, (right) spherical.

Abstract—
We investigate the usability of Euclidean, spherical and hyperbolic geometries for network visualization. Several techniques have been proposed for both spherical and hyperbolic network visualization tools, based on the fact that some networks admit lower embedding error (distortion) in such non-Euclidean geometries. However, it is not yet known whether a lower embedding error translates to human subject benefits, such as better task accuracy or lower task completion times. We design, implement, conduct, and analyze a human subjects study to compare Euclidean, spherical and hyperbolic network visualizations using tasks that span the network task taxonomy. Among several statistically significant results, the evaluation shows that while hyperbolic visualization requires more time than the Euclidean and spherical visualization, accuracy is not negatively impacted.

Index Terms—Network visualization

1 Introduction

Many graph visualizations begin with a node-link diagram drawn in 2 dimensional Euclidean space, the space of the computer screen or a sheet of paper. However, representations in different geometries have potential visualization benefits, for example focus+context effects and the possibility to represent the underlying data more faithfully. In the context of graph visualization, spherical [12, 15, 27], hyperbolic [17, 22, 23], and toroidal [3, 7] geometries have been considered.

Although these visualization approaches have potential benefits over the standard Euclidean setting, there has been little work in evaluating their performance with human subjects studies. Thus, there is little one can say about when it is appropriate to make use of such non-Euclidean graph visualizations. Preliminary results comparing spherical, hyperbolic and Euclidean graph visualizations indicate some benefits, but are limited by the type of tasks and type of graphs considered [9]. In particular, the effectiveness of the specific geometry could depend on the type of graph used, since some graphs achieve lower embedding error in spherical or hyperbolic space [19].

In this paper we describe a human subject study that aims for a more comprehensive evaluation of the utility of spherical, hyperbolic and Euclidean graph visualizations. Specifically, we design an experiment to help answer the following research questions:

- **RQ1**: Does a lower embedding error of a graph in a geometry correspond to better task support?
- **RQ2**: How is effectiveness affected when using non-Euclidean graph visualizations over the traditional Euclidean setting?
- **RQ3**: Are certain tasks better suited to certain visualization geometries?

We designed, implemented, ran, and analyzed a human subjects study that covers more of the possible graph task space and includes graphs representative of Euclidean, spherical, and hyperbolic space. We show that while the use of such geometries requires more interaction effort, they are no worse in time and accuracy overall compared to Euclidean visualizations. This partially confirms results and partially conflicts with results from earlier studies [4, 9]. A supplementary video summarizing our experiment and results can be found here `https://anonymous.4open.science/w/riemann-study-web-3AE6/VIS2024-nonEuclideanSurvey.mp4`.

2 Background and Related Work

We first give a brief background on non-Euclidean geometry and its use in graph visualization. We then introduce related work that we build upon in this paper.

2.1 Background

Consistent geometries are the spaces which satisfy the first four of Euclid's postulates from his classical *Elements* text: Euclidean, hyperbolic, and spherical (elliptical) geometries. Non-Euclidean geometries refer to both hyperbolic and spherical geometries, which are obtained by modifying Euclid's fifth postulate about parallel lines. Two lines A and B are said to be parallel if there is a third line, C, which intersects them both at right angles. In Euclidean geometry, the distance between A and B remains a fixed constant. The behavior is different in non-Euclidean geometries: in hyperbolic space, A and B will "curve away" and increase in distance the further from the intersection of C while in spherical space, A and B "curve into" each other eventually intersecting.

Non-Euclidean geometries have an interesting history with graph visualization dating back to the 1990s. They were first introduced with hyperbolic browsers for tree visualization, e.g., Lamping et al. [17] and Munzner [22,23]. Meanwhile, spherical geometry was used for force-directed graph layout [12,15] and for self-organizing maps [27,34].

The graph visualizations in our system that are seen in multiple figures throughout the paper (e.g. Fig. 1) are based on Multi-Dimensional Scaling (MDS) for graph drawing. MDS was first introduced in [30] for the purposes of statistical analysis of data, but has since become a popular dimension reduction [10] technique. MDS was first applied to graph drawing by Kamada and Kawai [14] and later improved via stress majorization in [11]. More recently, it has been shown that stochastic gradient descent is more consistent and requires fewer iterations to reach minimum stress levels than the stress majorization algorithm [36].

MDS can be extended to non-Euclidean geometries [19,20]. As in Euclidean space, these algorithms try to match the graph-theoretic distances in the graph to the geodesic distances (shortest distance between two points in the respective geometry) in the drawing space. A metric known as distortion [20,26] represents how well these graph-theoretic distances are preserved in the embedding. The distortion function (similar to the more familiar notion of stress) allows us to compare the quality of the embedding across different geometries and is defined as follows:

$$\binom{|V|}{2}^{-1} \sum_{i<j} \frac{|\delta(X_i, X_j) - d_{i,j}|}{d_{i,j}} \tag{1}$$

where X_i represents the position of vertex i in the drawing space, $d_{i,j}$ is the graph theoretic distance between vertex i and vertex j, and δ is the geodesic distance function between two points. Sala et al. show that certain graphs (e.g. trees) can be embedded with lower distortion in hyperbolic space than is possible in Euclidean space. Similarly, there are graphs that can be embedded with lower distortion on the sphere (e.g., 3-dimensional polytopes). Miller et al. show that lower distortion embeddings can be computed efficiently with the hyperbolic MDS algorithm [20] and with the spherical MDS algorithm [19].

2.2 Related Work

Yoghourdjian et al. [35] summarize and categorize over 100 papers that involve human-subjects empirical studies in graph visualization. They consider different features such as complexity, tasks, interactions, and size, with one of the key takeaways being that only 37% of studies in the survey used graphs with more than 100 nodes. Our experimental design is guided by the recommendations from Yoghourdjian et al.

Our study builds on a previous study by Du et al. [9], who introduced the iSphere graph visualization technique. iSphere maps a Euclidean drawing onto the sphere using inverse projections. A within-subjects human subjects study then compares Euclidean against hyperbolic and spherical layouts. Participants are presented with layouts of stochastic block model random graphs (graphs with clusters), and are asked to solve graph tasks in each of the three visualization styles. The layouts were derived from a stress majorization for Euclidean [11], a force-directed algorithm from [15] for hyperbolic, and the iSphere technique for the sphere. Du et al. Participants were presented with three tasks related to node degrees:

- **D1** Given a node A, find a target node B with the highest degree in A's neighborhood.
- **D2** Given two nodes A and B, find a target node C with the highest degree among the common neighbors of A and B.
- **D3** Given a path from node A to node B, find a target node C with the highest degree among the nodes on the path.

The study considers different screen sizes and graph sizes as additional dependent variables with the primary variable being the visualization style. Each graph had 3 clusters, with low or high graph modularity (a graph statistic quantifying cluster density). The main results indicate that the sphere and Euclidean visualizations provide comparable time and accuracy, while hyperbolic performed worse in some tasks. In particular, no significant differences were found between any of the three conditions for tasks **D1** and **D2**, while the hyperbolic condition was associated with significantly worse time and accuracy for task **D3**. Additionally, participants in the experiment consistently ranked the hyperbolic condition worse than both the Euclidean and spherical conditions.

Compared to the study of Du et al.'s, ours offers several improvements. First, we use 6 tasks that better cover the spectrum in the graph task taxonomy. Second, we use a consistent set of embedding algorithms across the three geometries: all layouts are MDS-based and optimize the same function in each geometry. Third, we use a more diverse set of graphs, in particular including graphs that can be embedded with lower distortion in each of the 3 geometries.

Chen [5] describes a systematic exploration of the design space for wrappable visualizations, which includes spherical and toroidal graph visualizations, but not hyperbolic visualizations. Chen et al. [6] compare several different drawing styles on the torus to the Euclidean plane and find that toroidal drawings have benefits in cluster identification tasks. Another study comparing Euclidean, toroidal, and several spherical drawing styles also finds that toroidal and spherical drawing styles outperform Euclidean drawings for cluster identification tasks [4].

Lens effects are visually similar to non-Euclidean visualizations. Fisheye views for graphs were first introduced by Sarkar and Brown [28] and have been used to provide focus+context in graph visualizations. A user study by Wang et al. [31] measures time and accuracy between flat drawings, a hyperbolic visualization, the iSphere technique [9], and the authors' own fisheye technique. The authors partially replicate some results from Du et al. [9] and observe that the fisheye view provided the lowest error and completion times with their data.

We classify the tasks used in our experiment along two dimensions of taxonomies: Lee et al. [18] provide a graph task taxonomy which covers graph-specific tasks at both a high and low level. Specifically, we use four topology-based tasks (adjacency, accessibility, common connection, connectivity) along with an overview task. Low level tasks for general data visualization are also applicable to graphs in many cases, so we additionally classify our tasks with the taxonomy by Amar et al. [1]. Specifically, our tasks include filtering, retrieving a value, comparing values, finding extrema, and determining range. Table 2 shows how our tasks fit in these two taxonomies.

3 Methodology and Design

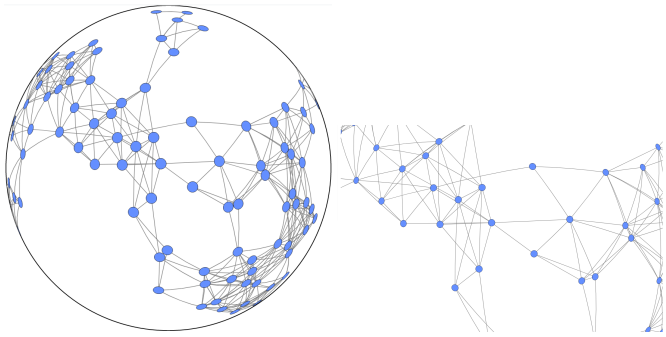
In this section we detail our experimental design. We were guided by prior human-subjects studies about graph visualization, extensive piloting, and the need to keep the study to a reasonable length.

3.1 Conditions

The conditions of interest are the geometries of the visualization: Euclidean, spherical, and hyperbolic. As much as feasible, the conditions differ only in the behaviour of the geometry with care taken to ensure everything else remains the same.

Euclidean (E): The Euclidean condition represents a standard node-link diagram, drawn in the flat, 2D space of the computer screen. Nodes are drawn as circles and edges drawn as straight line segments between the adjacent nodes. Node radius is set to 10 pixels, with edge width set to 1.5 pixels. The drawing is scaled so that it just fits the viewing window upon start. The border of the viewing window is drawn as 1 pixel wide black line, indicating the boundary. An example is shown in Fig. 1(b).

Hyperbolic (H): For the hyperbolic condition the graph is first embedded in hyperbolic space and then node positions on the computer screen are computed using a Poincarè projection. Nodes are still drawn as circles, with edges becoming circular arc segments, which represent geodesics in hyperbolic space. In the Poincarè projection, lengths and areas decrease at an exponential rate with the distance from the center of the view. For this reason, while nodes in the center of the projection have a radius of 10 pixels, they get smaller the further from the center they are. Similarly, edge width is 1.5 pixels near the center, and decreases away from it. The boundary of the disk is drawn with a solid 1 pixel black circle. An example is shown in Fig. 1(c).



(a) (b)

Fig. 2: Illustration of zooming in condition **S**: (a) is the default size and (b) shows a zoomed-in view on the center.

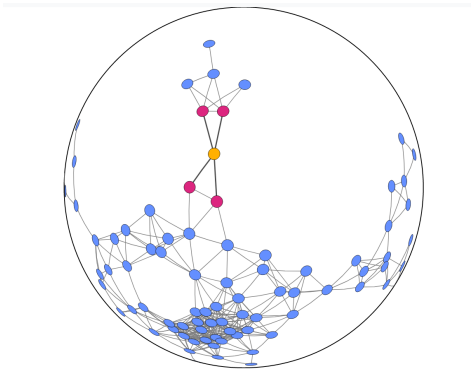


Fig. 3: Hovering over a node changes its color to yellow. All neighbors also change colors to magenta. The edges between the hovered node and its neighbors are emphasized by increased thickness.

Spherical (**S**): For the spherical condition the graph is first embedded on a sphere and the view provided shows half of the sphere (the "view-from-space" orthographic projection). Nodes are drawn as circles with 10 pixel radius at the center, but due to the projection become ellipses near the boundary. Edges are circular arc segments that represent geodesics in spherical space. The width of the edges is 1.5 pixels. The boundary of the projection is drawn with a solid 1 pixel black circle. Unique to the sphere, due to the orthographic projection, half of the drawing space is always obscured at one time. An example is shown in Fig. 1(a).

3.2 Interactions

Introducing interactions also introduces complexity and potentially confounding factors. However, some interactions are needed in order to make task-solving feasible, especially in the **H** and **S** conditions. We provide a small set of interactions (pan, zoom, hover, recenter, reset) as described below. We record all interactions by a participant, and utilize them in our analysis.

3.2.1 Pan

The geometric pan interaction allows users to bring an area of interest in the center. It is implemented via the standard "grab and drag" fashion in all three conditions. Note that the visual effect is different (e.g., on the sphere it corresponds to rotation). Panning is a common interaction employed in prior graph visualization studies [9, 24, 29].

3.2.2 Zoom

Zooming is another common interaction that is particularly useful for graphs with more than a few dozen vertices and zooming has been employed in prior graph visualization studies [9, 31]. There are two different notions of zoom in non-Euclidean geometries [20]. In Euclidean space, zooming is akin to bringing the plane closer or further away from the viewer, so our implementation is analogous for non-Euclidean spaces (bringing the sphere or hyperbolic plane closer or further away). Note that the other type of zooming (e.g., in spherical space one can change the size of the sphere) can be very useful in non-Euclidean spaces, but for consistency, we made available only the simple one; see Fig. 2.

3.2.3 Recenter around click

A interaction typically present in hyperbolic browsing systems [20] is the recenter interaction. When the user double-clicks on a point, the system automatically pans to make the clicked point the center of the layout. This interaction is provided in all three conditions. An example is shown in Fig. 4.

3.2.4 Hover

Early feedback indicated that participants were unsure exactly where they were clicking to perform the recenter interaction. With this in mind, we provided a hover interaction. When the mouse cursor enters the boundary of a node, the node color changes, the boundary and incident edges are thickened and the neighbors change color. The benefits of static and motion highlighting of nodes and their neighbors in node-link diagrams, especially when solving visual tasks such as ours, have been studied before [21, 32]. This interaction functions identically in all 3 geometry systems, and an example is found in Fig. 3.

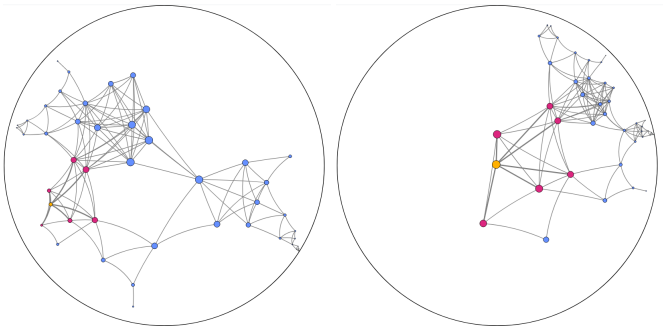


Fig. 4: Re-centering a node by double-clicking on it: (a) shows a node which is to be re-centered (yellow) and (b) shows the yellow node brought to the center of viewing window. This action is smoothly interpolated from the start to end positions.

3.2.5 Reset Visualization

Early participant feedback indicated that with too much zooming and panning, one can get "lost in space", especially in the hyperbolic condition where the entire graph can become invisible if forced too far from the center. This justified the inclusion of a reset button which restores the state of the visualization prior to any interactions.

3.3 Fixed parameters

Here we briefly discuss the fixed parameters of the experiment including the data (graphs and layout) and aesthetic choices (sizes and colors).

3.3.1 Graphs

The selection of graphs was a careful choice in order to be able to effectively answer **RQ1**. We initially intended to use only real-world graphs taken from [19] but ultimately decided that controlling for size and density was desirable for most graphs to reduce confounding factors. For each geometry, we generated two random graphs in a scheme nearly identical to the Random Geometric Graph generator for Euclidean space [25], which has already been generalized to hyperbolic space [16] and generalizes naturally to the sphere. These models first uniformly distribute $|V|$ points in the desired geometry, then connect two points with an edge if they are within a geodesic radius r.

We slightly modify this procedure by only taking the *first* $|E|$ shortest edges (sorted by length). In this way, we can specify both $|V|$ and $|E|$ in the graph to obtain the desired size and density while still capturing

	|V|	|E|	avg_degree	**E**_distortion	**H**_distortion	**S**_distortion
E_rand_small	50	150	6.0000	**0.1033**	0.1362	0.1217
E_rand_large	100	400	8.0000	**0.1011**	0.1312	0.1495
dwt_162	162	510	6.2963	**0.0928**	0.2217	0.1191
H_rand_small	50	150	6.0000	0.2465	**0.0913**	0.1143
H_rand_large	100	400	8.0000	0.2459	**0.1234**	0.1829
tree_150	150	149	1.9867	0.2815	**0.1514**	0.2068
S_rand_small	50	150	6.0000	0.2443	0.1121	**0.0897**
S_rand_large	100	400	8.0000	0.2450	0.1439	**0.1060**
dodecahedron_3	230	240	2.0870	0.3014	0.3615	**0.0544**

Table 1: Summary of the graphs that appeared in our experiment along with their embedding scores in each geometry. Accompanying images of each graph and layout can found in supplementary material.

the target space. We additionally ensure that all generated graphs are connected. These graphs are named E_rand_small, E_rand_large, S_rand_small, S_rand_large, H_rand_small, and H_rand_large. and are summarized in Table 1. We confirmed that the distortion values of the embeddings were indeed lower in the appropriate geometry.

We included three additional graphs of interest that would not appear from the generator but are representative of the types of graphs that can be embedded with low distortion in their "natural geometry". Euclidean space is natural for the wire mesh graph *dwt_162*. Hyperbolic space is natural for a tree on 150 nodes, *tree_150*. Spherical space is natural for the graph representation of the planar solid dodecahedron, subdivided three times, *dodecahedron_3*. These graphs are also summarized in Table 1 and seen in Fig. 5.

3.3.2 Graph Layout

The layout algorithms used were chosen carefully to be as similar as possible in order to reduce noise resulting from layouts in our results. We opted for MDS-based approaches for each geometry, as they are available in all three geometries. Specifically, we use a Euclidean MDS layout from [36], spherical MDS from [19], and hyperbolic MDS from [20]. Each of the 9 graphs $\times$ 3 geometries $=$ 27 graph-layout pairs were visually inspected to ensure they produced reasonable results. A representative example is shown in Fig. 1, and all layouts can be found in supplementary material.

3.3.3 Style

We attempt to keep node sizes as consistent as possible across each condition, despite the differing behaviour of the geometries. At the exact geometric center of each condition, the node size is precisely 10 pixels. In Euclidean geometry, the size and shape is unchanging. Hyperbolic geometry has a strong focus+context where the size of nodes rapidly decrease as they approach the boundary. In spherical geometry, the node circles are stretched to become ellipses, eventually becoming infinitely thin when they disappear "behind" the globe.

For some tasks, we ask questions related to "highlighted" nodes, which appear 50% larger in radius than they would otherwise.

Each condition has a default white background, with a 1 pixel black border to indicate the visualization space. This border is a rectangle in **E**, but a circle in both **H** and **S** conditions due to the chosen projections.

We use colorblind-safe options for the nodes [33]: light blue for visualizing nodes by default, yellow for hovered nodes, magenta for neighbors of hovered nodes, and orange for "highlighted" nodes related to the question; see Fig. 3. The exact color scheme can be found in supplemental material.

Edges are 1.5 pixels in width and are grey in color. When an incident node has been hovered over by the participant, the edge doubles in thickness and its color changes to black.

3.3.4 Display Size

As the interactions and feel of the tools were developed with desktop PCs in mind, we decided to restrict the availability of the test on mobile, tablet and other smaller devices. We set the minimum width of a device to be 801 pixels and any device under that measurement would not be able to access any page of the tool. Controlling the use of such small devices removes a potentially confounding variable in the study.

Lee et al. graph task taxonomy [18]

Amar et al. task taxonomy [1]	Adjacency	Accessibility	Common Connection	Connectivity	Overview
Retrieve Value		T6	T2		
Filter	T1				
Compute Value	T5				
Find Extremum	D1, D3		D2	T3	
Sort					
Determine Range					T4
Characterize Dist.					
Find Anomalies					
Cluster					
Correlate					

Table 2: Task classification for our tasks and Du et al.'s tasks.

3.4 Tasks

In an effort to cover as broad a spectrum of tasks as possible, we use a table that represents the intersection between the graph task taxonomy of Lee et al. [18] and the more general visualization taxonomy of Amar et al. [1]; see Table 2.

First, we identified where Du et al.'s [9] three tasks fell in this table, shown in orange (**D1, D2, D3**). We then attempted to cover all columns (all types of graph tasks) while covering as many rows as possible. Our experiment utilizes 6 tasks, shown in blue in the table:

- **T1**: Count neighbors of a node with a given degree
- **T2**: Count common neighbors of two nodes
- **T3**: Determine the length of the shortest path between two nodes
- **T4**: Estimate the size (nodes or edges) of the graph
- **T5**: Determine the most frequent degree of a group of nodes
- **T6**: Count how many additional nodes are reachable from a given node in given number of steps

An instance for one of the above tasks is a layout of a graph, with highlighted relevant nodes (or node), and 4 possible answers (one of which is correct). The task instances in our study are generated programatically and ensure that there is exactly one correct answer. We also control the level of difficulty targeting under 30 seconds for completion. Possible answers are always shown in sorted order.

For each graph, we create 3 instances for each of our 6 tasks, resulting in $9 \times 6 \times 3 = 162$ total unique instances. An example of one instance of each task is shown in Fig. 6 and all 162 instances can be found via the web page; see Section 6.

3.5 Experimental design

We were aiming for a within-subjects experiment, as it important to be able to directly compare participant results by geometry. However, there are too many graph/task/geometry triples ($9 \times 6 \times 3 = 162$) making the experiment too long for one person.

Thus we settled on a within subjects design with respect to geometries and graphs (every participant sees all three geometries and all 9 graphs), but broke the tasks into 3 groups of 2 (each participant sees only two of the six tasks).

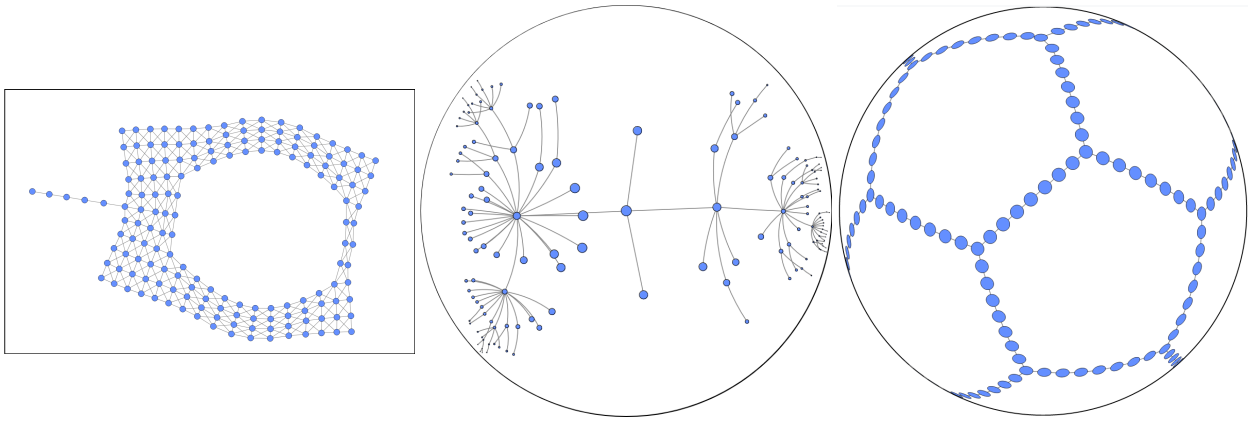
This results in a design where each participant answers 54 questions (3 geometries x 9 graphs x 2 tasks). The order in which participants used each geometry, as well as the order of the questions, is determined based on a balanced Latin square to reduce learning effects.

3.5.1 Experiment outline

The experiment was reviewed and approved by our Institutional Review Board (details omitted for anonymity). After consenting to participation in the study, the main body of the experiment contains 3 blocks, one per condition.

Each block begins with an introduction page. This page introduces definitions, descriptions of interactions, and an example visualization in the current condition. Participants are encouraged to spend time familiarizing themselves with the visualization and interactions using a small sample graph; an example is shown in supplementary material.

After the introduction, each block goes through 18 questions with the current condition, one for each graph-task pair. The final question page design is found in Fig. 6. After the last task, participants are asked to answer ten subjective questions about the condition (e.g., how aesthetically pleasing was this condition?) on a 1-5 Likert scale.



(a) (b) (c)

Fig. 5: (a) dwt_162 shown in the **E** condition (b) tree_150 shown in the **H** condition (c) dodecahedron_3 shown in the **S** condition. Each of these graphs are representative of the types of graphs which have lowest distortion in their respective spaces.

The order of the 3 blocks, as well as the order of the questions within each block, is determined based on a balanced Latin square.

The final page of the experiment collects general demographic data (e.g., age group, gender) and provides a window for free-form feedback.

3.5.2 Results from Pilot

During the design phase, we asked visualization researchers (not involved in the study beyond the pilot) to participate and provide feedback. This helped catch bugs and minor fixes, but more importantly we were able to identify and rectify several non-trivial issues. We briefly discuss three examples.

One such issue was the pan effect for S and H conditions. Several pilot participants noted that the pan direction felt “inverted" from Euclidean space. While we as designers imagined rotating the ball or hyperbolic plane, this was not how the effect was being perceived. We changed the pan effect so that the pan is always in the direction of the mouse cursor for all conditions.

We received several instances of feedback detailing that the question node was difficult to find (see Fig.3). Based on this feedback, we increased the size of the nodes of interest to allow them to further stand out (in addition to being of different color).

Some task instances were too difficult and others too easy. This feedback helped us control the difficulty level across conditions to hopefully avoid floor and ceiling effects.

4 Results and Analysis

We recruited participants for the study by making a webpage available on Reddit, posting to subreddits r/takemysurvey and r/samplesize. We additionally contacted friends and colleagues. No incentives were offered for participation. In total, we had 29 participants complete the study. The average amount of time spent on trials was 25.5 minutes, while the average real-world time (breaks, directions, feedback, etc. included) was 46.2 minutes. Of our participants, about 80% were men, 12% were woman, and the remaining 8% preferred not to say. 85% of participants were between the ages of 18-35 and the remaining in the ages of 36-55. Familiarity with network visualizations ranged from 15% either completely new or have seen in passing, 45% had discussed in coursework, 20% use them for their work, and 20% experts.

4.1 Hypotheses

To answer the research questions posed in the Introduction, we developed the following hypotheses:

- **H1a** Graphs which have lower distortion in Euclidean space will be significantly more effective in the Euclidean visualization condition rather than the hyperbolic or spherical conditions.
- **H1b** Graphs which have lower distortion in hyperbolic space will be significantly more effective in the hyperbolic visualization condition rather than the Euclidean or spherical conditions.
- **H1c** Graphs which have lower distortion in spherical space will be significantly more effective in the spherical visualization condition rather than the Euclidean or hyperbolic conditions.
- **H2** The Euclidean visualization condition will be more effective than either the hyperbolic or spherical condition overall.

We had no concrete hypotheses for **RQ3**, but analyze the data by task and discuss the results.

4.2 Quantitative analysis

The majority of data collected was not normally distributed, and so non-parametric tests were performed. Unless otherwise specified, the significance level of all tests was done at $p < 0.05$.

Effectiveness was measured in three ways: accuracy, response time, and mental effort [2, 13]. Accuracy is a binary measure: 0 if a participant answered incorrectly and 1 if a participant answered correctly. Response time is the amount of time in seconds from when the visualization loads until the participant submits their answer so this includes reading the question, navigating the visualization, counting, etc. Effort is measured indirectly, by adding together all of the pan, zoom, and double-click interactions. We considered including the hover interaction in the effort measure but decided against it as hovering was used almost equally in all three conditions; see Fig. 8(d).

Figures such as Fig. 7 show 95% confidence intervals (CI), a statistical measure which shows where the true mean of the collection lies with 95% probability. This statistic is computed based on the number of samples and the variance of those samples [8].

H1 Results To test H1 we first grouped user responses by visualization condition (**E**, **H**, **S**) and by distortion values (e, h, s) (see the blocks in Table 1) to partition our 27 graph-drawing pairs into 9 groups: e(**E**), e(**H**), e(**S**), h(**E**), h(**H**), h(**S**), s(**E**), s(**H**), s(**S**). Here, the group h(**E**) refers to the group of graphs which have lower distortion in hyperbolic space (the middle three graphs in Table 1) and are drawn in a Euclidean style (condition **E**).

We then aggregated accuracy over all tasks for each of these groups per participant and performed a two-way non-parametric Wilcoxon Rank Sum test i.e. for H1a, we test e(**E**) against e(**H**) and e(**E**) against e(**S**). Only one significant difference was found in accuracy: between e(**E**) and e(**S**). We next measured the response times and effort. The timing data indicate two significant results: between e(**E**) and e(**H**), with e(**H**) requiring more time and between s(**S**) and s(**H**), with s(**H**) requiring more time. The effort data indicates statistically significant differences for every comparison. The results are visualized in Fig. 7 and p-values can be found in Table 3.

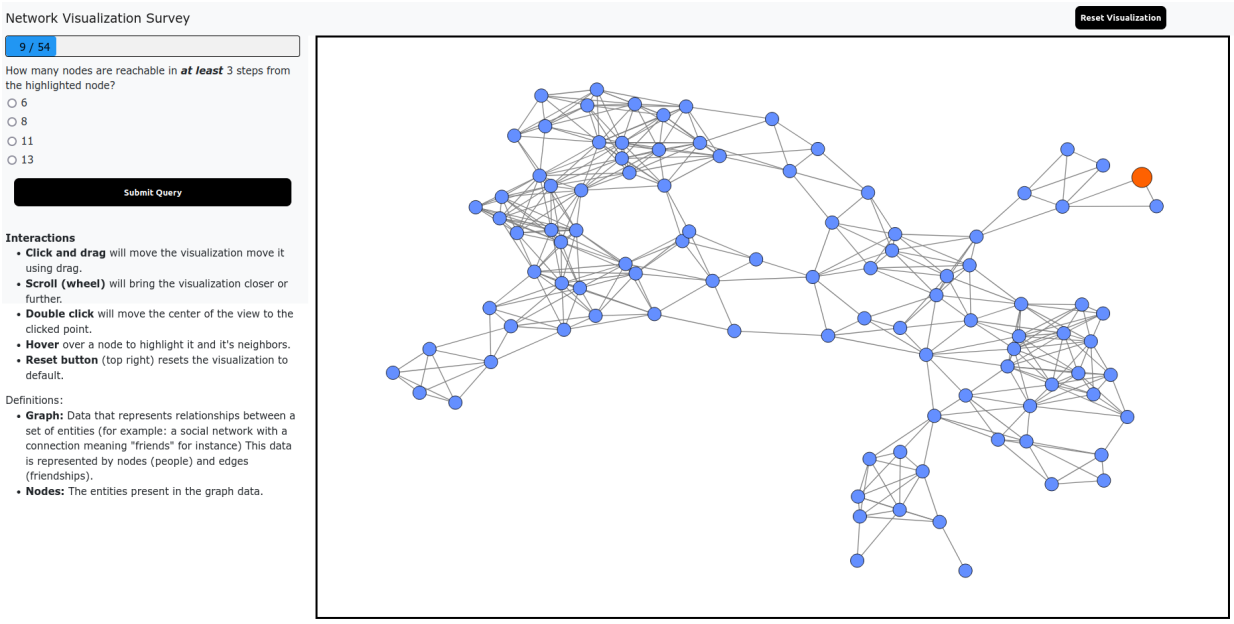
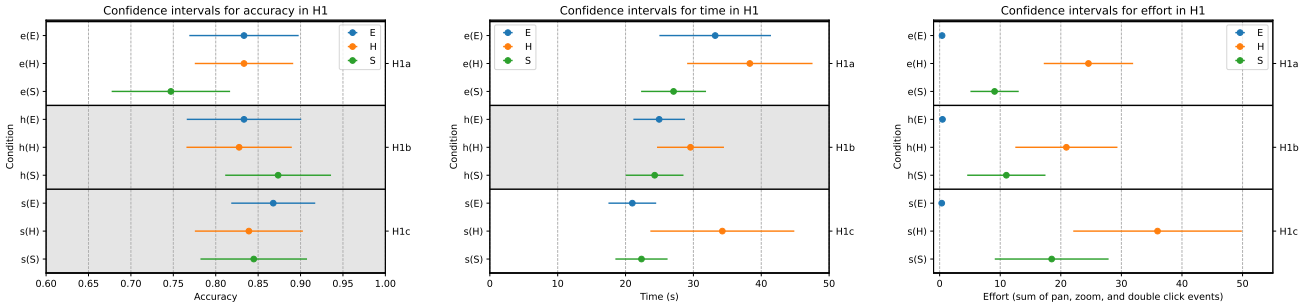


Fig. 6: An example question page from the experiment. Participants were shown a graph drawing (center window) with a question about the graph asked in the left side panel. Interaction descriptions and terminology definitions were in a scrollable window underneath the question and responses. The reset visualization button is shown in the top right.



(a) (b) (c)

Fig. 7: Mean values for (a) accuracy, (b) time, and (c) effort means including 95% Confidence Intervals (CI) for H1. Grey cells indicate no significant differences were found between the three conditions, while white cells indicate at least one difference. Exact pairs of differences are shown in Table 3.

	accuracy	Time	effort
e(**E**) – e(**H**)	0.695	**0.031**	< **0.001**
e(**E**) – e(**S**)	**0.014**	0.619	< **0.001**
h(**H**) – h(**E**)	0.449	0.143	< **0.001**
h(**H**) – h(**S**)	0.118	0.069	**0.001**
s(**S**) – s(**E**)	0.767	0.394	< **0.001**
s(**S**) – s(**H**)	0.847	**0.033**	**0.001**

Table 3: P-values obtained by Wilcoxon tests for H1. Bolded values indicate significant results, the CI of which are visualized in Fig. 7.

H1 Discussion The fact that there is only one significant result when comparing accuracy seems to indicate that participants were able to effectively use all tools to complete tasks, regardless of whether or not we expect the graph to be easier to navigate in a given geometry. We are not able to confirm H1 for accuracy, however we can conclude that the non-Euclidean geometries are ***no worse*** in terms of accuracy than Euclidean. The one significant accuracy result shows that Euclidean visualization of Euclidean graphs is preferable to the sphere, possibly due to the sphere "hiding" portions of the visualized graphs.

The time results tell an interesting story for the **H** condition. Response time is significantly worse in **H** for graphs with lowest distortion in Euclidean and spherical drawn in the **E** and **S** condition respectively. However, for graphs with lowest distortion in hyperbolic space, there are no significant differences between the three conditions, indicating that the **H** condition may be more effective on these graphs than on others.

In terms of effort (measured by total number of pan, zoom, and double-click events), differences are significant for all tests. However, as can be seen in Fig. 7(c), the order of the means is always the same. **E** had the fewest interactions, **S** had the next most, and **H** always had the largest number of interactions. Regardless of distortion values, the effort required was nearly always the same. Therefore, while the non-Euclidean geometries are no worse in accuracy, they required more effort from participants than Euclidean visualization. These results are further confirmed by looking at the average total interactions for each geometry in Fig. 8(d).

If we take a closer look at the CI plots in Fig. 7(c), we see that the **E** condition required very few interactions on average to arrive at the correct answer. More effort was required on average for graphs with lower distortion in **S**, but this has a plausible explanation: the class

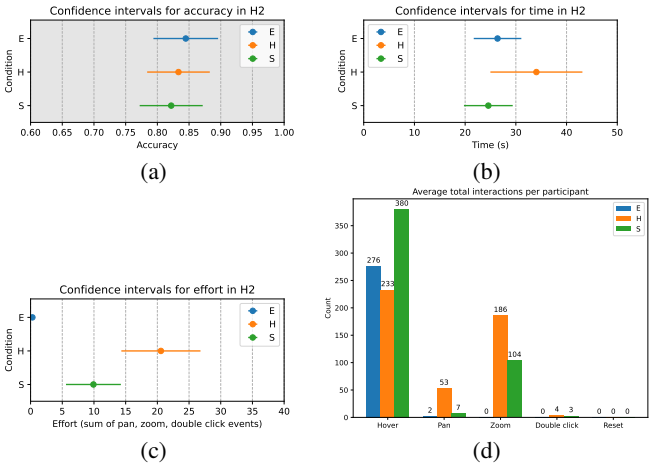


Fig. 8: Mean values for (a) accuracy, (b) time, and (c) effort means including 95% Confidence Intervals (CI) for H2. Average number of interactions (d) show that while hover was used in all geometries, pan and zoom were utilized more in the **H** and **S** conditions.

of graphs with low distortion in **S** should be exactly the graphs that make full use of the spherical space, meaning roughly half the graph is obscured in an orthographic projection.

Note that the mean values for **H** are lowest in the group h(**H**) when compared to e(H) and s(H), indicating some support for H1b. This holds for accuracy, time and effort, but none of these differences are statistically significant.

H2 Results The testing of H2 was more straightforward; we are interested in how participants behaved in each visualization condition overall (regardless of graph type). For each condition, we aggregated participant accuracy, response times, and effort, then performed two-way non-parametric Wilcoxon Rank Sum tests to test **E** against **H** and **E** against **S**. We found no significant difference in either test; see Fig. 8(a-c) and Table 4. While not stated in our hypothesis, we also compare condition **H** to **S**, and find significance in response time. The result for effort is largely the same as for H1.

	Accuracy	Time	Effort
E – H	0.557	0.052	$<$ **0.001**
E – S	0.097	0.435	$<$ **0.001**
H – S	0.373	**0.004**	$<$ **0.001**

Table 4: P-value results from Wilcoxon tests for H2 analysis. Bold values indicate statistical significance.

H2 Discussion It is clear from the results that we are unable to verify H2, which indicates effectiveness for non-Euclidean visualizations is comparable to that of Euclidean visualizations. The key takeaway here is that non-Euclidean visualizations are ***no worse*** than Euclidean in terms of accuracy and response time. This partially agrees with the results from Du et al. [9] who find few significant differences in time and error between **E, H** or **S**.

Results By Task Here we break up participant responses by visualization condition and by task. While we did not formulate task-specific hypotheses, we expected to find some differences between conditions. The focus+context of the **H** condition should support adjacency tasks such as **T1** and **T5**. The response times should generally follow the order of **E, H, S**, from shortest to longest. For the overview task, **S** will likely be worse than the rest, since half of the drawing area is hidden.

We aggregate participant responses and perform a three way non-parametric Friedman test to detect differences between geometries in accuracy and time. Confidence intervals can be found in Fig. 9. We

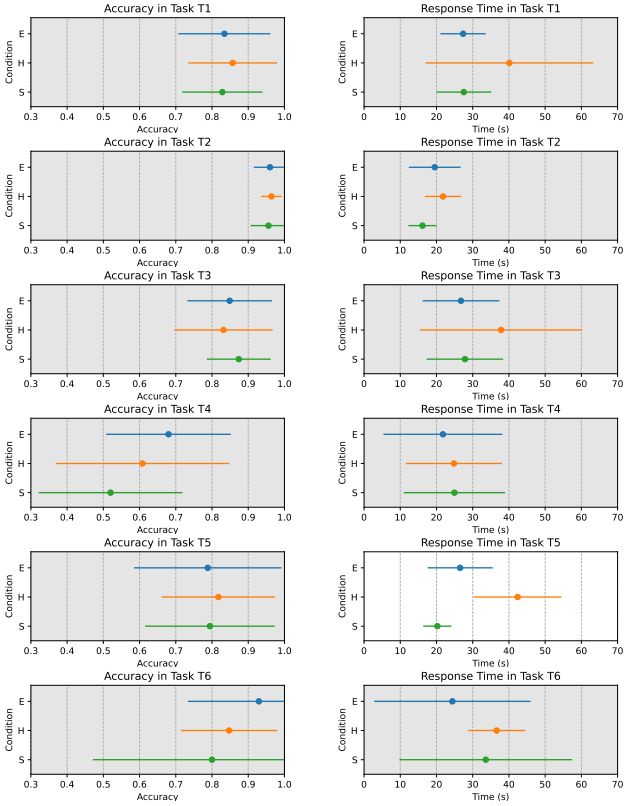


Fig. 9: Means and 95% CI by task for each visualization condition in (left) accuracy and (right) time. Grey cells indicate no statistical significance where white cells indicate at least one pairwise significance.

were unable to detect any differences in accuracy, which agrees with our overall results from H1.

For time, we found one significant result in **T5**. We then conducted a pairwise non-parametric Wilcoxon test at significance level $p < \frac{0.05}{3} \approx 0.017$ which showed a difference in means between **H** and **S**. Interestingly, **S** took around half as long on average as **H** for **T5** with no loss in accuracy which we can confirm visually in Fig. 9.

If we look at the data more closely, there are a few results that while not statistically significant, are surprising nonetheless. Based on prior studies, we expect the accuracy and time to follow the order of **E**, **S** and **H**. In **T1** and **T5** for instance the **H** condition slightly outperforms the others in accuracy at the cost of more time. In both instances, the variance in time for **H** is quite large comparatively which might indicate further training is required to use the condition more effectively. In **T4**, the overview task, the mean accuracy for **S** is much worse. This makes sense as much of the graph may be hidden in the **S** condition.

While there are no statistically significant results for our task-based expectations, we can look at the order of the means. Tasks **T1** and **T5** are instances of adjacency tasks that we expected to perform well in hyperbolic space, and the **H** condition does show slightly better accuracy. We were surprised that response times do not always conform to the expected order: the one significant result shows that **S** required less time than **H** but the mean is also lower than **E** (though difference between **E** – **S** was not significant). A similar pattern can be seen for **T2**. Finally, the overview task **T4** indeed shows lower accuracy in the **S** condition, though there is high variance for all three conditions.

4.3 Subjective Quantitative Feedback

We collected participant feedback using a Likert-scale at the end of each of the 3 blocks (corresponding to the three conditions). We asked for two types of feedback: interaction-related questions and aesthetics-related questions. The first type can help us understand the perceived utility of the provided interactions in the given condition; we can also

see if this is correlated to actual use of the interactions. The second type of questions can help see whether personal preferences match the accuracy/time/effort data. All questions were presented in the same way with the 5 options “Strongly Disagree", “Disagree”, “Neutral”, “Agree”, “Strongly Agree”.

4.3.1 Interaction-Related Questions

We collected feedback from participants to express how useful a given interaction was for each condition. The questions were:

- The hover interaction assisted answering questions.
- The click-and-drag interaction assisted answering questions.
- The scroll interaction assisted answering questions.
- The double-click interaction assisted answering questions.
- The reset button interaction assisted answering questions.

We summarize the results of this feedback in Fig. 10(a); we can clearly see a trend in the usefulness of each interaction. The most useful was the hover interaction, followed by click-and-drag (pan), then by scroll (zoom), then by double-click (recenter) and finally reset. Looking at each geometry, we can see that participants reported interactions were more useful in the **H** condition. Conversely, the **E** condition was helped less through interactions.

To verify these visual results, we conduct a statistical analysis of the responses to detect any significant difference. We conduct a three way Friedman test to find which questions contain any differences, followed by pairwise two way Wilcoxon tests at significance level $p < 0.01$. Of these, only the double-click question had significant difference in mean with **E** against **H** being different at $p = 0.003$ as well as **E** against **S** being different at $p = 0.005$. This result is in agreement with our interaction counts and qualitative feedback; the double-click feature was rarely used in **E**. The remaining results were not significant, but can be found in supplemental material.

4.3.2 Aesthetics-Related Questions

We collected feedback from participants on their subjective thoughts and preferences for each condition. The questions were:

- The provided interactions overall supported answering the questions asked.
- This style of visualization was easy to read and understand.
- The visualizations were aesthetically pleasing.
- The visualization style supported answering the questions asked.
- It was enjoyable utilizing the interactions.

We summarize this feedback in Fig. 10(b) and the results favor Euclidean geometry. For all five questions **E** wins or ties for first place. The non-Euclidean geometries have little difference between each other except on the question "This style of visualization was easy to read and understand", where **H** is lower than both **E** and **S**.

We again conduct a statistical analysis in the same form as before. Only one test resulted in significance: between **E** and **H** ($p = 0.003$) in the question “The visualization style was easy to read and understand." In contrast to the subjective feedback from Du et al. [9] where **S** was consistently close to **E** in preference and in task support, our results indicate that **S**, **H** and **E** are comparable and **H** does not conspicuously lag behind the other two.

4.4 Qualitative Feedback

We collected comments and feedback from participants after they completed the survey. They were encouraged to respond with free text to the following prompts:

- Please describe the strategy you used to answer the questions.
- Did you have any problems taking this study?
- Please leave any other comments or observations below.

4.4.1 Strategy

We asked question regarding the participants’ strategy in order to understand how they are making use of given interactions to answer the questions. Most participants reported that the **Hover** interaction followed by **Pan** interaction were the most helpful for solving the tasks provided to them. One participant reported,

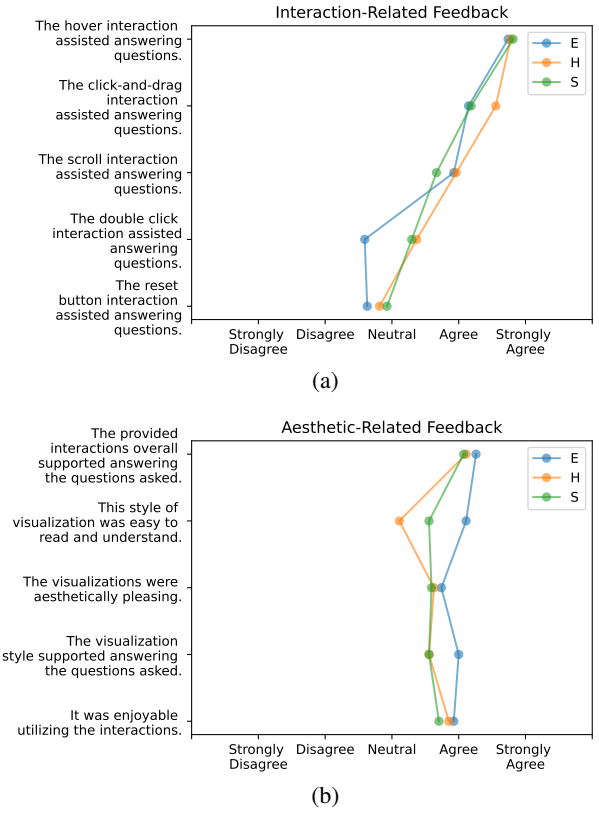


Fig. 10: Mean values from all participants regarding interaction-related feedback (a) and aesthetics-related feedback (b). Exact wording of questions is shown along the y-axis, with responses on the x-axis.

> *I rely heavily on the hovering tool where the nodes and the edges are highlighted when I hover my mouse over the nodes. I use this to find linked nodes when a small region has a lot of nodes together. I also use it to find the neighbors of the highlighted node and move my cursor back and forth between the target node and its neighbors to count edges. I also use the double-click feature often center the target node so I can easily check neighbors and edges.*

A few participants also mentioned the zoom and recenter interactions in their responses. Notably, no participant mentioned the reset visualization as a part of their strategy which is consistent with our data; see Fig. 8(d). The task group participants were placed in heavily influenced their strategy. An instance of a comment left by a participant who was placed in the task group with **T3** and **T5** is,

> *Find the point, then use the hover to determine neighbors. For the shortest path found the most direct path. For the neighbors, just hovered to find the neighbors then went through each neighbor, hovered and counted the degree.*

The feedback left by participants reconfirms our results regarding the use of interactions with different visualizations.

4.4.2 Problems

By understanding what problems participants faced while taking our study, we can highlight the limitations and challenges of the study. One of the main problems highlighted by different participants was that the question node (node in orange in Fig.3) did not change color on being hovered and if it was the neighbor of a hovered node. One user mentioned,

> *Already highlighted nodes (orange) don’t change color*

when they are neighbors of a hover node, making it difficult to see if they are connected to it or not

This was an intentional design decision to always have question nodes be visually consistent, but proved frustrating to several participants. Another limitation we encountered through participants' comments was node-edge or edge-edge overlap. A few comments mentioned it was difficult for the participants to identify if there are connections between nodes because of overlapping and made it difficult for them to solve the tasks. One participant left a comment,

Some nodes were positioned on top of edges so it was sometimes difficult to see if two nodes were connected or if the edge went through to a different node.

While the hover and zoom interactions were designed to counter this issue, it could not fully solve overlap for the more dense graphs. Another common problem faced by participants was the zoom and pan in the **H** and **S** conditions. Several users reported it was "un-intuitive" or that it "felt weird" to operate. A user reported,

The click-and-drag for elliptic surfaces felt a little off for how it moved the graph at times, when dragging near the edges of the circle.

This suggests that more training might be needed to allow participants to get more comfortable with the non-standard graph visualization conditions.

4.4.3 General Comments

The comments for the last question acted to solicit any other opinions or issues the participants faced that they felt was not apt to mention for the other two questions. Several participants mentioned their opinions on what they liked and what they preferred while some expressed their dissatisfaction with one of the conditions. The **H** condition was especially polarizing with some describing it as "interesting" or "visually fun" and others detailing their frustration. One participant wrote

I love the aesthetics of the (hyperbolic style).

while another said

I hated the (hyperbolic) style. The globe and flat styles were comparable, but I would never want to use the (hyperbolic) one.

Several participants also noted they were only asked two types of questions throughout the experiment,

Asking the same question again and again did not make the experience pleasant.

Frustrated and/or bored participants is a limitation of the study (participants were asked to answer the same two questions 54 times).

5 Discussions, Limitations, and Future Work

The results overall seem to indicate that with appropriate interactions, non-Euclidean representations of graphs can be competitive in terms of time and accuracy to standard Euclidean representations. However, there seems to be no additional benefit gained from a lower distortion value. This may potentially be due to the relatively small sizes of graphs and that much larger (1000s of nodes and edges) may produce different results. Additionally, while we generated graph embeddings whose distortion matched our hypothesis (e.g. Euclidean graphs with Euclidean embeddings whose distortion is lower than a hyperbolic and spherical embedding), we only required that it was lowest in the specified geometry. For the graph E_rand_small, the Euclidean distortion was 0.10 while the hyperbolic distortion was 0.13. Without knowing the distribution of distortion scores it is hard to say whether or not the differences in distortion are large enough. We leave a thorough exploration of the distortion metric as future work.

As our experiment was within-subjects but between tasks, each participant worked with only 2 of the 6 tasks. Further, the total number of participants per task was low (9 or 10). More participants would likely have resulted in more solid data.

The relatively large variance for time (see Fig. 7(b)) in the **H** condition might indicate that participants did not receive enough training to be comfortable working in that condition. While it is believable that hyperbolic takes longer on average, we expect with proper training the mean can be made significantly faster.

As we can observe from the demographic data, our participants are a biased sample with the majority being young men with some graph experience. A more diverse set of participants might help with the generalizability of the results.

Several feedback comments mentioned using a trackpad, while the systems were designed with a mouse in mind to navigate. Everything was tested to work with both, but many design decisions (speed of zoom, pan, etc.) were calibrated with a mouse.

6 Conclusion

We aimed to extend prior non-Euclidean graph visualization human-subjects studies such as [9] to understand the role of such tools in the broader graph visualization field. The trials of our experiment can be found online (anonymized for submission): https://anonymous.4open.science/w/riemann-study-web-3AE6/. Our results are clear: non-Euclidean visualizations with sufficient interaction support are no worse in terms of time and accuracy than Euclidean visualizations of graphs. These tools require the use of these interactions (and thus, effort) to be useful, whereas Euclidean visualizations can be read statically for the sizes of graphs we considered. The effect of the distortion metric has little effect on the time and accuracy in the geometry.

REFERENCES

[1] R. A. Amar, J. Eagan, and J. T. Stasko. Low-level components of analytic activity in information visualization. In J. T. Stasko and M. O. Ward, eds., *IEEE Symposium on Information Visualization (InfoVis 2005), 23-25 October 2005, Minneapolis, MN, USA*, pp. 111–117. IEEE Computer Society, 2005. doi: 10.1109/INFVIS.2005.1532136 2, 4

[2] M. Burch, W. Huang, M. Wakefield, H. C. Purchase, D. Weiskopf, and J. Hua. The state of the art in empirical user evaluation of graph visualizations. *IEEE Access*, 9:4173–4198, 2021. doi: 10.1109/ACCESS.2020.3047616 5

[3] K. Chen, T. Dwyer, B. Bach, and K. Marriott. It's a wrap: Toroidal wrapping of network visualisations supports cluster understanding tasks. In *CHI '21: CHI Conference on Human Factors in Computing Systems*. ACM, 2021. 1

[4] K. Chen, T. Dwyer, Y. Yang, B. Bach, and K. Marriott. Gan'sda wrap: Geographic and network structured data on surfaces that wrap around. In S. D. J. Barbosa, C. Lampe, C. Appert, D. A. Shamma, S. M. Drucker, J. R. Williamson, and K. Yatani, eds., *CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022 - 5 May 2022*, pp. 135:1–135:16. ACM, 2022. doi: 10.1145/3491102.3501928 1, 2

[5] K.-T. CHEN. It's a Wrap! Visualisations that Wrap Around Cylindrical, Toroidal, or Spherical Topologies. 8 2022. doi: 10.26180/20723092.v1 2

[6] K.-T. Chen, T. Dwyer, B. Bach, and K. 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*, pp. 1–12, 2021. 2

[7] K.-T. Chen, T. Dwyer, K. Marriott, and B. Bach. Doughnets: Visualising networks using torus wrapping. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2020. 1

[8] F. M. Dekking. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science & Business Media, 2005. 5

[9] F. Du, N. Cao, Y. Lin, P. Xu, and H. Tong. iSphere: Focus+context sphere visualization for interactive large graph exploration. In G. Mark, S. R. Fussell, C. Lampe, m. c. schraefel, J. P. Hourcade, C. Appert, and D. Wigdor, eds., *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017*, pp. 2916–2927. ACM, 2017. doi: 10.1145/3025453.3025628 1, 2, 3, 4, 7, 8, 9

[10] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea. Toward a quantitative survey of dimension reduction techniques. *IEEE Trans. Vis. Comput. Graph.*, 27(3):2153–2173, 2021. doi: 10.1109/TVCG.2019.2944182 2

[11] E. R. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In *International Symposium on Graph Drawing*, pp. 239–250. Springer, 2004. 2

[12] M. H. Gross, T. C. Sprenger, and J. Finger. Visualizing information on a sphere. In *Proceedings of VIZ'97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*, pp. 11–16. IEEE, 1997. 1, 2

[13] W. Huang, P. Eades, and S. Hong. Measuring effectiveness of graph visualizations: A cognitive load perspective. *Inf. Vis.*, 8(3):139–152, 2009. doi: 10.1057/IVS.2009.10 5

[14] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, 1989. 2

[15] S. Kobourov and K. Wampler. Non-Euclidean spring embedders. *IEEE Trans. Vis. Comput. Graph.*, 11(6):757–767, 2005. 1, 2

[16] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010. 3

[17] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Human Factors in Computing Systems, CHI '95 Conference Proceedings*, pp. 401–408. ACM/Addison-Wesley, 1995. 1, 2

[18] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. 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, 5 pages, p. 1–5. Association for Computing Machinery, New York, NY, USA, 2006. doi: 10.1145/1168149.1168168 2, 4

[19] J. Miller, V. Huroyan, and S. Kobourov. Spherical graph drawing by multi-dimensional scaling. In *Graph Drawing and Network Visualization: 30th International Symposium, GD 2022, Tokyo, Japan, September 13–16, 2022, Revised Selected Papers*, pp. 77–92. Springer, 2023. 1, 2, 3, 4

[20] J. Miller, S. G. Kobourov, and V. Huroyan. Browser-based hyperbolic visualization of graphs. In *15th IEEE Pacific Visualization Symposium, PacificVis 2022, Tsukuba, Japan, April 11-14, 2022*, pp. 71–80. IEEE, 2022. doi: 10.1109/PACIFICVIS53943.2022.00016 2, 3, 4

[21] T. Moscovich, F. Chevalier, N. Henry, E. Pietriga, and J. Fekete. Topology-aware navigation in large networks. In D. R. O. Jr., R. B. Arthur, K. Hinckley, M. R. Morris, S. E. Hudson, and S. Greenberg, eds., *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pp. 2319–2328. ACM, 2009. doi: 10.1145/1518701.1519056 3

[22] T. Munzner. H3: laying out large directed graphs in 3d hyperbolic space. In *IEEE Symposium on Information Visualization*, pp. 2–10. IEEE Computer Society, 1997. 1, 2

[23] T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3d hyperbolic space. In *Procedings of the 1995 Symposium on Virtual Reality Modeling Language*, pp. 33–38. ACM, 1995. 1, 2

[24] J. Pan, W. Chen, X. Zhao, S. Zhou, W. Zeng, M. Zhu, J. Chen, S. Fu, and Y. Wu. Exemplar-based layout fine-tuning for node-link diagrams. *IEEE Trans. Vis. Comput. Graph.*, 27(2):1655–1665, 2021. doi: 10.1109/TVCG.2020.3030393 3

[25] M. Penrose. *Random Geometric Graphs*. Oxford University Press, 05 2003. doi: 10.1093/acprof:oso/9780198506263.001.0001 3

[26] F. Sala, C. D. Sa, A. Gu, and C. Ré. Representation tradeoffs for hyperbolic embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 4457–4466. PMLR, 2018. 2

[27] A. P. Sangole and G. K. Knopf. Visualization of randomly ordered numeric data sets using spherical self-organizing feature maps. *Comput. Graph.*, 27(6):963–976, 2003. doi: 10.1016/J.CAG.2003.08.012 1, 2

[28] M. Sarkar and M. H. Brown. Graphical fisheye views. *Commun. ACM*, 37(12):73–84, 1994. doi: 10.1145/198366.198384 2

[29] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Trans. Comput. Hum. Interact.*, 3(2):162–188, 1996. doi: 10.1145/230562.230577 3

[30] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952. 2

[31] Y. Wang, Y. Wang, H. Zhang, Y. Sun, C. Fu, M. Sedlmair, B. Chen, and O. Deussen. Structure-aware fisheye views for efficient large graph exploration. *IEEE Trans. Vis. Comput. Graph.*, 25(1):566–575, 2019. 2, 3

[32] C. Ware and R. J. Bobrow. Supporting visual queries on medium-sized node-link diagrams. *Inf. Vis.*, 4(1):49–58, 2005. doi: 10.1057/PALGRAVE.IVS.9500090 3

[33] B. Wong. Points of view: Color blindness. *Nature methods*, 8:441, 06 2011. doi: 10.1038/nmeth.1618 4

[34] Y. Wu and M. Takatsuka. Visualizing multivariate network on the surface of a sphere. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation-Volume 60*, pp. 77–83, 2006. 2

[35] V. Yoghourdjian, D. Archambault, S. Diehl, T. Dwyer, K. Klein, H. C. Purchase, and H. Wu. Exploring the limits of complexity: A survey of empirical studies on graph visualisation. *Vis. Informatics*, 2(4):264–282, 2018. doi: 10.1016/J.VISINF.2018.12.006 2

[36] J. X. Zheng, S. Pawar, and D. F. M. Goodman. Graph drawing by stochastic gradient descent. *IEEE Trans. Vis. Comput. Graph.*, 25(9):2738–2748, 2019. doi: 10.1109/TVCG.2018.2859997 2, 4