



Kent Academic Repository

Wallinger, Markus, Chapman, Peter, Nöllenburg, Martin, Rodgers, Peter and Blake, Andrew (2026) *Combined network and set visualization with hoop and linear diagrams*. In: Diagrammatic Representation and Inference. Lecture Notes in Computer Science . pp. 350-365. Springer

Downloaded from

<https://kar.kent.ac.uk/115555/> The University of Kent's Academic Repository KAR

The version of record is available from

https://doi.org/10.1007/978-3-032-34178-5_35

This document version

Author's Accepted Manuscript

DOI for this version

Licence for this version

CC BY (Attribution)

Additional information

For the purpose of open access, the author(s) has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

Versions of research works

Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in **Title of Journal** , Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries

If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our [Take Down policy](https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies) (available from <https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies>).

Combined Network and Set Visualization with Hoop and Linear Diagrams

Markus Wallinger¹[0000-0002-2191-4413], Peter Chapman²[0000-0002-5524-5780],
Martin Nöllenburg³[0000-0003-0454-3937], Peter Rodgers⁴[0000-0002-4100-3596],
and Andrew Blake⁵[0000-0001-5856-4544]

¹ Technical University of Munich, Munich, Germany

² Edinburgh Napier University, Edinburgh, United Kingdom

³ TU Wien, Vienna, Austria

⁴ University of Kent, Kent, United Kingdom

⁵ University of Brighton, Brighton, United Kingdom

Abstract. Set and network data co-occur frequently, with the sets representing node groupings and the network representing connections between node pairs. This paper investigates two different designs for jointly visualizing set and network data: HoopNet and LinearNet. HoopNet juxtaposes a hoop diagram with a circular network layout, whereas LinearNet juxtaposes a linear diagram with a linear network layout. We investigate two variations of each: a set-first and a network-first layout. Our human-subject studies show there is no difference between set-first and network-first optimization. However, participants achieved significantly higher accuracy with HoopNet, indicating it is a more effective visualization for set and network data than LinearNet.

Keywords: Linear Diagram · Hoop Diagram · Node-link Diagram

1 Introduction

Networks express relationships between pairs of objects, while set systems capture groupings of objects. Many real-world datasets combine these structures, including biochemical reaction networks [34] and social networks [16]. Such datasets require visualizations that support joint reasoning about both pairwise connections and overlapping sets. Although network and set visualization have each been studied extensively, their integration remains challenging [33].

In this paper, we investigate two designs for integrated set and network visualization. *HoopNet* combines hoop diagrams [27] with circular node-link layouts [3], while *LinearNet* combines linear diagrams [29] with arc-based network layouts (see Figure 1). Both combine set and network aspects, revealing set membership and connectivity. These designs are grounded in prior empirical work showing that linear diagrams outperform some other set visualization designs [7,15]. They align naturally with circular and linear network layouts, respectively. Whilst related designs appear in prior work [26,32], we develop layout optimization methods and evaluate their comparative effectiveness through controlled studies.

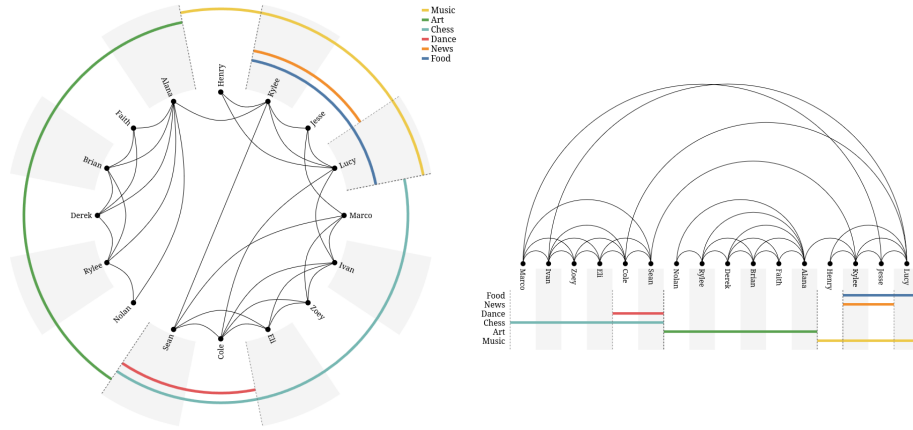


Fig. 1: Both HoopNet (left) and LinearNet (right) visualize networks and sets defined over nodes. HoopNet juxtaposes a hoop diagram with a circular network layout. LinearNet juxtaposes a linear diagram with a linear network layout. Both are set-first variants minimizing the number of gaps in the set lines.

We investigate the following research question: *How do radial and linear spatial organizations influence human accuracy and efficiency when reasoning about overlapping sets and network connectivity?*

To enable valid comparisons, we developed two exact optimization strategies based on integer linear programming: a *set-first* strategy that minimizes gaps in set lines, and a *network-first* strategy that minimizes edge crossings in the network layout. These strategies yield controlled layouts for studying how spatial organization affects task performance. Based on these layouts, we conducted three user studies focusing on accuracy and task completion time. The first two studies compared set-first and network-first optimization within each design. The third and main study directly compared HoopNet and LinearNet, using their set-first variants. Our results indicate that the radial design of HoopNet better supports comprehension in integrated set and network visualizations.

Our contributions include:

- **Diagram Design:** We detail the design of two closely related diagram forms: HoopNet and LinearNet and discuss how the radial spatial organization of HoopNet better supports integrated reasoning about sets and networks.
- **Optimization Framework:** We introduce two exact integer linear programming (ILP) strategies, *set-first* and *network-first* for the layout of HoopNet and LinearNet.
- **Empirical Evaluation:** Through three crowdsourced studies ($N \approx 510$), we systematically compare HoopNet and LinearNet in terms of accuracy and completion time. HoopNet achieved significantly higher accuracy than LinearNet ($\sim 10\%$ overall, $\sim 13\%$ for larger networks).

2 Background and Related Work

The focus of this paper is on visualizing group structures in networks. A recent survey is by Vehlow et al. [33]. Early work in this area overlaid Euler diagrams with networks [22]. Here, we focus on designs that juxtapose the network and set aspects. Some techniques use arc diagrams to show the network while indicating group structure with trees [24] or treemaps [23]. TreeNetViz [16] uses a circular layout for the network while juxtaposing a tree hierarchy outside the circle. Work on visualizing gene interaction in immune dysregulation [32] is similar to HoopNet, but does not discuss design approaches. Also similar is a proposed radial hypergraph visualization [18] which places nodes in a circle, with binary edges in the centre and hyperedges around the outside.

Our work relies on combining set and network visualizations. Regarding set visualization, we use linear diagrams and hoop diagrams. A survey by Alsallakh et al. [2] categorizes linear diagrams as an aggregation-based set visualization technique, conceptually similar to matrix-based approaches. The core visual metaphor is a matrix in which each set is represented by a horizontal line in its corresponding row, and each element by a column. A horizontal line segment is drawn within a cell if the corresponding element belongs to the set. Unlike matrix-based techniques, linear diagrams merge consecutive elements of the same set into a single continuous line segment that spans all relevant cells. An empirical study [29] identified key design guidelines for linear diagrams, highlighting that minimizing the number of set line gaps significantly improves task performance. Furthermore, perceived visual clutter has been shown to correlate with the number of gaps [1]. However, minimizing the total number of gaps is an NP-hard problem [6,9]. One of the most widely used visualizations for general set systems are Euler diagrams. However, the scalability of Euler diagrams is restricted, and studies have demonstrated that linear diagrams outperform Euler diagrams as a more effective set visualization [7,15]. Recently, hoop diagrams [27] have been introduced as a variant of linear diagrams, where visual elements are arranged in a circular rather than linear layout. A comparative study found no significant difference in user performance between hoop and linear diagrams, suggesting both are effective for set visualization. Our work builds on these foundations. We follow validated design guidelines from the above prior studies when integrating linear and hoop diagrams with network visualization in HoopNet and LinearNet.

Regarding network visualization, node-link diagrams are a common design, where nodes are represented by small disks and edges are represented by curves connecting the respective nodes. A recent survey of surveys on network visualization [11] summarizes the current state of research for different network dataset types. For this paper, the most relevant types of node-link layouts are the circular layout [10,30] and the arc diagram [36]. Circular layouts arrange all nodes on a circle, while linear arc layouts arrange all nodes on a horizontal or vertical line. One of the key advantages of these layouts is that they are fully defined by a cyclic or linear order over the nodes. Combinatorially, circular layouts are considered equivalent to linear layouts [19] as all nodes are in a convex position

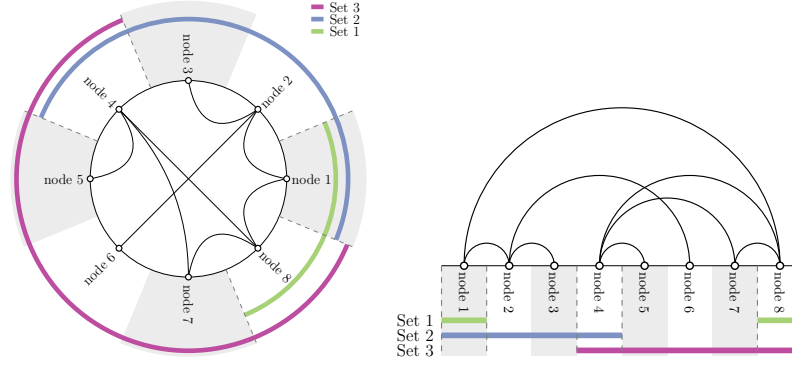


Fig. 2: HoopNet (left) and LinearNet (right) with the same instance and node ordering.

and induce the same number of edge crossings. Optimizing the layout to minimize crossings is considered to increase the readability of networks [3]. Most of the available algorithms for circular layouts are heuristics, but optimal ILP formulations have also been proposed [14].

3 Design Description

We combine hoop set diagrams with circular network layouts to create **HoopNet** diagrams. Similarly, we combine linear set diagrams with linear network layouts to obtain **LinearNet** diagrams. We juxtapose and attach the set visualization to the network visualization so that our designs consider both aspects in a coherent hybrid visualization that embodies two contrasting spatial metaphors, radial vs. linear, that we hypothesize support different cognitive strategies. The concepts of our design description are illustrated in Figure 2.

Set Visualization. HoopNet and LinearNet diagrams are based on hoop diagrams [27] and on linear diagrams [29], respectively. Hoop diagrams are structurally similar to linear diagrams, as hoop diagrams can be considered linear diagrams wrapped around a circle by attaching the end of the linear diagram to its beginning. As there are no experimental results for hoop diagrams, we use design principles empirically validated for linear diagrams [29] and assume they also apply to hoop diagrams. These design principles are:

- **DP1:** minimize the number of gaps in set lines.
- **DP2:** use visual guide lines at the beginning and end of line segments.
- **DP3:** use thin set lines.

Network Visualization. We represent the network as a node-link diagram guided by the following principles. These have been validated by studies [25] in the case of **DP4** and by following standard linear and circle designs in the case of **DP5**, as curved edges considerably improve crossing angles:

- **DP4**: minimize edge crossings.
- **DP5**: draw nodes equidistantly and edges as arcs above the line or inside the circle.

Labeling. Lastly, we also need to consider the position and style of labels.

- **DP6**: align node labels.
- **DP7**: allow easy differentiation between adjacent nodes.

While **DP1** and **DP4** depend on node ordering and thus require algorithmic optimization, the remaining principles affect rendering; see Figure 2 for examples.

To support **DP2**, we add dashed gray guide lines at the start and end of set line segments, indicating when two consecutive nodes differ in set membership.

For **DP5**, we render edges differently in the two designs. In HoopNet, edges are drawn as cubic splines with three control points: the two endpoints and a midpoint. The midpoint is obtained by first computing the straight-line midpoint, then projecting a perpendicular toward the circle’s center, and placing the control point halfway along this line. In LinearNet, edges are drawn as circular arcs centered at the midpoint between the endpoints.

Finally, to address **DP7**, we alternate background shading to distinguish between sectors of consecutive nodes. For even-sized networks, we alternate white and light gray; for odd-sized networks, we cycle among white, very light gray, and light gray to increase visual saliency.

Set-First and Network-First Layouts. To support a controlled comparison between HoopNet and LinearNet, we compute optimal node orderings that minimize either gaps (**DP1**) or edge crossings (**DP4**). These two objectives are often in direct conflict: a gap-minimal order can incur many edge crossings, and vice versa; see Figure 3. Both problems are NP-hard [6,9], and we solve them exactly with integer linear programs; full formulations are in the supplemental material on OSF.

Crossing minimization follows the formulation of [14]. Binary variables $x_{u,v}$ encode whether node u precedes v in the order, with transitivity constraints enforcing a valid linear (equivalently [19], cyclic) order. For each pair of independent edges, an indicator variable is active whenever their endpoints interleave in the order, and the objective sums these indicators.

Gap minimization reduces to the Travelling Salesperson Problem [9] on the complete auxiliary graph $G^* = (V, E^*)$ whose edge weight $w_{u,v}$ is the Hamming distance between rows u and v of the set–node incidence matrix. Placing two nodes adjacently induces one gap for every set whose membership differs between them, so a minimum-weight Hamiltonian cycle corresponds to a gap-minimal cyclic order. For LinearNet, we add a zero-weight dummy node to mark the start and end of the linear order.

Two-stage combination. In the set-first variant, we solve the gap ILP first for a gap-optimal tour. Consecutive nodes with identical set memberships may be freely swapped without creating new gaps, so we then run the crossing ILP with the relative order of all other node pairs fixed and select the layout with the

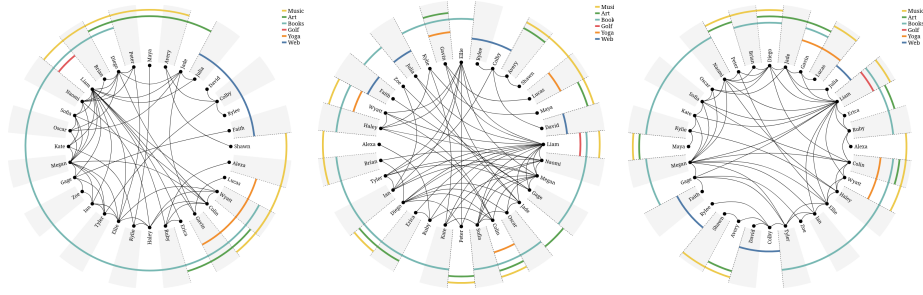


Fig. 3: Instance with 32 nodes and 6 sets. Ordering nodes randomly (middle) induces 439 crossings and 29 gaps. Optimizing gaps (left) results in 139 crossings and 8 gaps. Optimizing crossings (right) results in 82 crossings and 24 gaps.

fewest crossings. The network-first variant inverts this: We enumerate multiple crossing-optimal orders and pick the one with the fewest gaps. See Figure 3 for the resulting layout differences.

Although computational performance was not our primary focus, we assessed HoopNet and LinearNet in terms of runtime. Detailed numerical results are provided in the supplemental material. Runtimes for both strategies quickly became impractical for larger instances (32 nodes and 5 sets), exceeding three hours for network-first. Thus, although optimal solutions provide reliable benchmarks, exact optimization may not be suitable for practical use in all cases.

4 Empirical Study Process

Our three empirical studies were designed to help address our research question.

We have two alternative layout approaches for both of these diagram types: set-first and network-first. We followed the tournament-based experimental methodology in [29]. Hence, in the first two studies, we examined which of set-first and network-first is the better design for HoopNet and LinearNet. For the final study, we selected the best design for both and compared HoopNet with LinearNet to determine whether a radial or linear spatial organization should be preferred. This led to the following six null hypotheses:

- **H0:A1** There is no difference in accuracy for HoopNet between optimizing network-first or optimizing set-first.
- **H0:B1** There is no difference in task completion time on HoopNet between optimizing network-first or optimizing set-first.
- **H0:A2** There is no difference in accuracy for LinearNet between optimizing network-first or optimizing set-first.
- **H0:B2** There is no difference in task completion time on LinearNet between optimizing network-first or optimizing set-first.
- **H0:A3** There is no difference in accuracy between HoopNet or LinearNet.

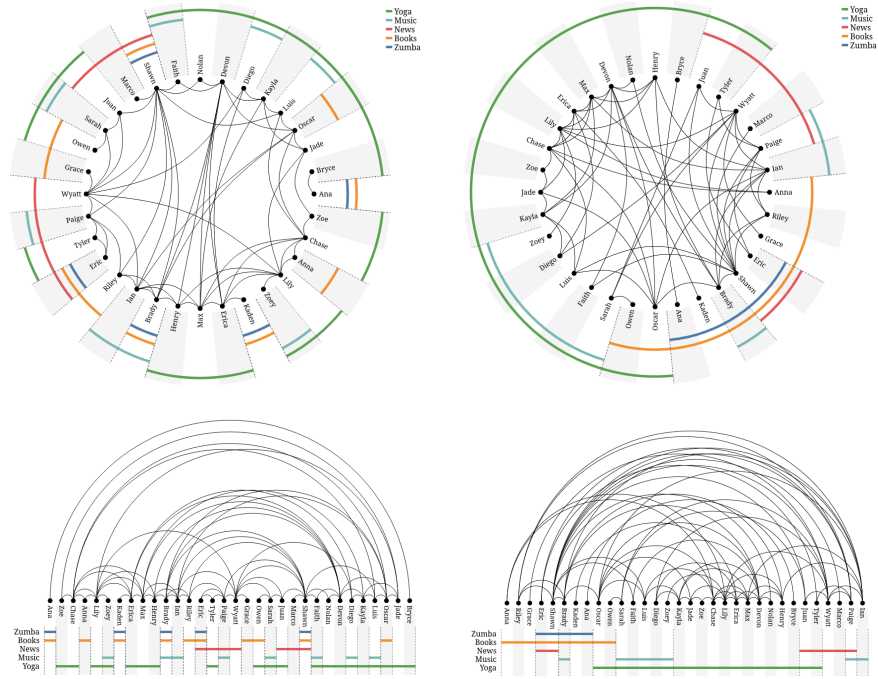


Fig. 4: One large ($|V| = 32$, $\mathcal{S} = 5$) instance used in our study visualized as HoopNet (top) and LinearNet (bottom). The left column shows network-first optimization while the right column shows set-first optimization.

- **H0:B3** There is no difference in task completion time between HoopNet or LinearNet.

We expected each of these accuracy and time hypothesis pairs to be answered with three separate studies. We developed an online study platform and chose the crowdsourcing platform Prolific (www.prolific.com) to recruit participants. Crowdsourcing is a widely used and reliable way of finding participants for information visualization studies [4]. Prolific is specifically designed for academic research, providing suitable filters and features.

4.1 Methodology

Data Preparation. Our data are drawn from the SNAP dataset [21], in particular, Twitter follows and circles. Twitter had connections between users, where one user followed the tweets of others. Twitter circles were smaller, and tweets could be restricted to selected user groups. To ease understanding for a wider audience, we reinterpreted this data as undirected friendship links rather than follows, and as subject interest groups rather than circles. Singleton nodes and

non-overlapping sets were discarded, and further automatic filtering was applied to the simplest components to meet the size requirements listed below. The nodes were then renamed using a list of common first names, avoiding names that started with the same letter when possible. We used common interests for the sets, again avoiding those sharing a starting letter.

Diagram Size. To provide variety, we used *small* and *large* diagrams for the data collection tasks, with sizes typical for network visualization user studies [5]. Small diagrams had 16 nodes and 3 or 4 sets. Large diagrams had 32 nodes and 5 or 6 sets. Small and large are used to differentiate between our two datasets in this context. Diagrams were chosen at random as long as they could provide unambiguous, correct answers of sufficient complexity and enough potential alternative incorrect answers for the multiple-choice selection. The incorrect alternatives were chosen at random from the potential alternatives.

Tasks. To ensure variety in tasks, we developed four task types (T1–T4), with examples listed below. They were divided into *set-oriented* (T1, T2) and *network-oriented* tasks (T3, T4). All tasks required studying both the set and network information in order to complete successfully, however, we considered the set-oriented tasks to require study of the set information first, followed by the network information. Conversely, we considered network-oriented tasks to require studying network information first, followed by the set information.

In terms of developing tasks, we accessed existing task classifications for sets [2] and networks [20]. In all cases, the set part of the tasks falls into the “Tasks related to elements” classification because the shared information between the set visualization and the network visualization is through the elements (or nodes). The network part of the tasks again involves nodes, and the tasks are classified as “Adjacency” (both counting adjacency and finding adjacent nodes) and “Common connection”. We calibrated the level of difficulty of both the set and network parts of tasks in the knowledge that non-expert users would need to complete all parts of the multi-section tasks to be successful.

Study Design. We chose a between-subject design, so each subject saw the same conditions throughout. This was because we thought a within-subject design would lead to too much confusion when participants were presented with both HoopNet and LinearNet diagrams.

From our experience with previous studies, we estimated that 12 data collection tasks were a suitable number for participants to complete in a reasonable time. Whilst a smaller study with 8 tasks would have been possible with the combination of 4 task types and 2 data sizes, we performed more tests to allow for more data collection in the time we felt was reasonable. Of the four task types, half had two small and one large diagram (T1, T3), whilst the others had one small and two large diagrams (T2, T4). We chose the details of each task (such as number, person, or set intersections) required for the tasks at random. However, we added restrictions on answer uniqueness and to ensure that the questions would be at the right level of complexity for participants to engage with the diagrams. For instance, for T4, the friendship count was restricted to

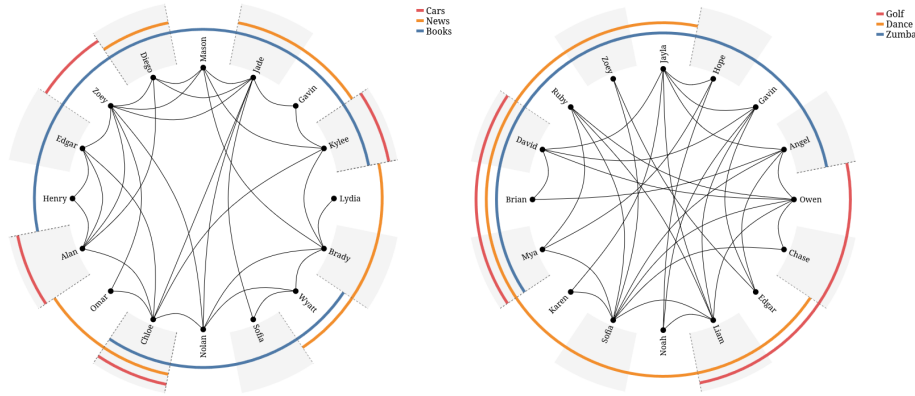


Fig. 5: One small ($|V| = 16$, $\mathcal{S} = 3$) instance used in our study visualized as HoopNet. In (left), we asked the participants: “What are the interests of the person who is friends with both Kylee and Edgar?” (T3). For (right), we asked the participants “Who is interested in Golf, Dance, and Zumba and is also friends with Jayla?” (T1).

a random number with exactly one correct answer in the diagram and between 3 and 8. Below, we list one example question for each task type.

- T1. *Who is interested in Golf, Dance, and Zumba and is also friends with Jayla?* Here, the set intersection and person vary. This is related to the set sub-task of finding set intersections, followed by the network sub-task of finding named nodes and discovering their links. An alternative, less efficient, network-first strategy is to find a named node, examine its connecting nodes, and then find all set intersections among them. We mitigate against this alternative strategy by putting the set intersection first in the question. This is for Figure 5 (right), the first data collection task, a small diagram. Answer options: *David, Zoey, Edgar, Maya*, with *David* being correct.
- T2. *How many friendship links are there joining people who have interests in News only to people who have interests in Music only?* Here, the two set intersections vary. This relates to the set sub-task of finding named set intersections, followed by the network sub-tasks of identifying the nodes in the set intersections and discovering the node links. We could not identify a sensible alternative network-first strategy. This is for Figure 4, the tenth data collection task, a large diagram. Answer options: *0, 1, 2, 3*, with *1* being correct.
- T3. *What are the interests of the person who is friends with both Kylee and Edgar?* Here, the two people vary. This relates to the network sub-tasks of finding two named nodes and discovering network connectivity, followed by the set sub-task of identifying the node set intersections. We could not identify a sensible alternative set-first strategy. This is for Figure 5 (left), the

third data collection task, a small diagram. Answer options: (a) *Cars, News, and Books*, (b) *Cars only*, (c) *News, Books, and nothing else*, (d) *Books only*, with (a) *Cars, News, and Books* being correct.

- T4. *What are the interests of the person who is friends with exactly 6 other people?* Here, the number varies. This is related to the network sub-task of discovering well-connected nodes, followed by the set sub-task of discovering the node intersections. We could not identify a sensible alternative set-first strategy. Figure 1 shows the fourth training task, a small diagram. Answer options: (a) *Art only*, (b) *Music, Food and nothing else*, (c) *Chess only*, (d) *Music, News, Food and nothing else*, with (a) *Art only* being correct.

Procedure and Conditions. We produced 18 tasks, each with separate diagrams for each of the four conditions: *HoopNet set-first*, *HoopNet network-first*, *LinearNet set-first*, *LinearNet network-first*. The same questions, response options, and underlying data were used for all conditions. Only the visualizations changed. The first four tasks presented were training tasks, with diagrams of increasing size, and the last training diagram was the same size as a small diagram in the study. In the case of training diagrams, after the user selected an answer, a response with the correct answer and an explanation was given. The training questions were one of each task type.

Regarding the remaining 14 tasks, twelve were for data collection, with task types and diagram sizes as described above. The final two tasks were attention checks that showed a standard question image, with three options appearing as standard options (without referencing any label in the diagram) and a fourth option labeled “Choose this option”.

The training questions were always presented in the same order at the start of the study, straight after the initial information page and consent to start the study. The twelve data collection questions and two attention-check questions were presented in random order, with the answer options also randomized. The study was finished by presenting a demographics collection form and, finally, a completion link back to Prolific.

4.2 Study Execution

Overview. We conducted three studies to test our six hypotheses, each following a similar format.

- **Study 1:** HoopNet set-first vs. HoopNet network-first.
- **Study 2:** LinearNet set-first vs. LinearNet network-first.
- **Study 3:** HoopNet set-first vs. LinearNet set-first.

Each study would collect *accuracy* and *time* when answering questions. Accuracy is the frequency of the single correct answer out of four options presented. Time is the time from when the task first appears to the user until the user selects an option and presses a button to move on to the next task. We note that throughout this paper, we consider accuracy to be the more important indicator than time.

Participants. Our participant selection method used Prolific, which selected participants based on predefined filters we set and passed them to our study vehicle. We requested an even gender distribution from Prolific. The three studies comprised participation counts of: LinearNet 164 (network-first 83, set-first 81); HoopNet 164 (network-first 83, set-first 81); and HoopNet vs LinearNet 167 (HoopNet 87, LinearNet 80). Inclusion criteria were: fluent English, Prolific approval rate greater than 95% and no prior participation in an earlier study in the sequence. More details on participant filters, pilot studies, and data collection can be found online; see the link below.

Condition Selection for Study 3. The conditions of Study 3 were determined by the results of Studies 1 and 2. The analysis and discussion for Study 1 and Study 2 have been moved online to save space (see link to OSF below). There were no significant results between set-first and network-first in Study 1 and Study 2. The means for accuracy were better for set-first in both of these studies and so we used HoopNet set-first and LinearNet set-first in Study 3.

Study materials, including questions and diagrams, anonymized study results data and the discussion of Study 1 and Study 2, are available online on OSF⁶. Additionally, we also preregistered⁷ our study design before collecting responses.

4.3 Layout Study Results

Two outcome variables were recorded: participants' accuracy (a binomial response variable) and the time taken to answer a question (in seconds). Since each participant answered twelve questions, they each supplied twelve accuracy responses and twelve timing responses. Each participant's responses are not independent, and thus traditional analyses, such as logistic regression (for the accuracy results) and ANOVA (for the time responses) might lead to incorrect conclusions. We use general estimating equations (GEE) [12], combined with generalized linear models, to account for these clustered data. Analysis was performed using the `geepack` package of R [17]. We can still calculate p -values and report them as for more standard techniques. The interpretation of a p -value is straightforward: it gives the probability of obtaining the data, assuming there is no difference between the populations' parameters. We consider results with a p -value less than 0.05 to be significant.

4.4 HoopNet vs. LinearNet Study Results

The descriptive statistics for Study 3, which compared HoopNet and LinearNet diagrams using set-first optimization, are shown in Table 1.

From Table 1, we see that the accuracy for HoopNet diagrams is significantly higher than for LinearNet diagrams. This result holds for the overall dataset and for the subsets based on diagram size. In percentage terms, there is around a 10% increase in accuracy of HoopNet over LinearNet. Thus, the accuracy rates when

⁶ <https://osf.io/6kr9g>

⁷ <https://osf.io/wha79>

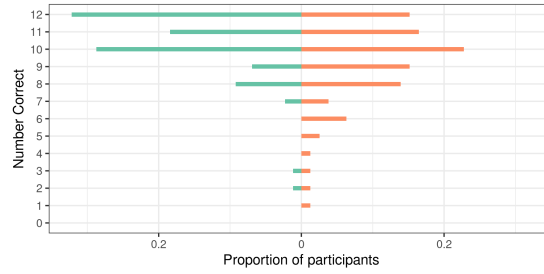


Fig. 6: Distribution of number of correct answers for HoopNet and LinearNet.

using HoopNet are significantly better than using LinearNet, and moreover, both use the same amount of time. We, therefore, have evidence to reject $H0:A3$.

For task completion time, there were no significant differences between participants using HoopNet and those using LinearNet. This result is the same regardless of diagram size. Whilst the *sample* mean time is quicker for HoopNet, the result is not significant. Hence, there is no evidence to reject $H0:B3$.

The distributions of correct answers are shown in Figure 6. We can see that the modal category for HoopNet diagrams is all questions answered correctly. In contrast, for LinearNet diagrams, the modal category is one in which two questions were answered incorrectly.

Regarding diagram size, for large diagrams HoopNet had a mean accuracy of 83% compared to 71% for LinearNet. For small diagrams, the difference was lower as HoopNet had a mean accuracy of 89% compared to 83% for LinearNet.

The significant accuracy result provides evidence that *HoopNet is a more effective visualization than LinearNet*.

5 Discussion of Results

Set-first vs. Network-first. We conjecture that the lack of significant results in Study 1 and Study 2 on optimization strategy is due to the sub-tasks that involve solving both network and set sub-tasks. Hence, whether the optimization is for either network layout or set layout, there is always one part of the full task advantaged and one part disadvantaged.

Table 1: Descriptive statistics for HoopNet and LinearNet diagrams

Treatment	Mean Accuracy			Mean Time (s)		
	Overall	Small	Large	Overall	Small	Large
HoopNet	0.8611	0.893	0.830	47.16	39.37	54.95
LinearNet	0.7646	0.835	0.707	50.22	41.80	58.84
<i>p-value</i>	0.00041	0.026	0.00039	0.360	0.510	0.410

Linear vs. Radial. Our main finding is that HoopNet consistently supported higher accuracy than LinearNet, particularly for larger networks, without increasing task completion time. Our findings challenge established results, as linear designs tend to outperform radial designs on accuracy and completion time [8,13], with recent work confirming this for bar charts [35] and matrix layouts [31]. We conjecture that a combination of two effects is responsible for our results. Firstly, the curves representing edges in HoopNet are easier to follow than in LinearNet. In the latter, the curves turn 180° by design. In HoopNet, the shorter edges turn more while the longer edges are straighter. Secondly, radial layouts form a continuous loop, which makes it easier to follow sets around the circle and see how they overlap. In contrast, linear layouts rely on left-to-right scanning, which can make it harder to keep track of memberships, particularly when nodes in the same set are widely separated.

The large-diagram accuracy gap suggests LinearNet is nearing its understandability limit, whereas HoopNet could scale further.

6 Conclusion

We have presented two designs for visualizing set and network data: HoopNet and LinearNet. HoopNet juxtaposes hoop set diagrams with circular network diagrams, whereas LinearNet juxtaposes linear set diagrams with linear network diagrams. We have implemented two alternative ILP algorithms for exact optimization of our designs. For set-based tasks, we can minimize gaps in set lines using our set-first method. For network-based tasks, we can minimize edge crossings with our network-first method. Our empirical human-subject studies yielded statistically significant results: the radial design of HoopNet is a more effective visualization than the linear design of LinearNet.

Limitations. As with any empirical study, our results are valid within the constraints imposed by the study. In our case, the results are valid for the task types undertaken by our pool of participants drawn from the general population, although we mitigated this by introducing variety into the source data and tasks through randomization and introducing several different task types. Our studies avoided many use cases for these visualizations, including interactions and embedding them in a web page with additional subject context. This was done to minimize the number of variables present in the experimentation. The studies were limited in scale, meaning we chose data sets of a size that allowed the questions to be of a suitable difficulty and to be gathered in a reasonable amount of time from crowd-sourced participants. We expect our designs will scale beyond the tested size of 32 nodes, with an informal upper bound around 100 nodes and 12 sets, typical of small-to-medium sizes in graph visualization user studies [5]. This range exceeds the computability of our ILP and would require heuristic layout methods. For larger instances, we conjecture that interaction techniques can be a suitable way to handle visual complexity with detailed views on demand.

Our study did not look at designs without node ordering vs. the two orderings that were chosen to be examined. Given the lack of significance in the first

two studies, the choice of optimization seems to have little effect, but the final significant result is only valid with regard to set-first optimization, and other optimization (or no optimization) combinations might differ in results.

Future Work. More empirical human-subject evaluation is necessary to examine the scalability of HoopNet diagrams, which may not yet have reached their maximum size in the current study (see Section 5). Other empirical work is also possible, comparing HoopNet against alternative set network visualization techniques, including embedded methods, such as Euler diagrams with networks and linesets on networks [28]. We found no statistical difference between set-first and network-first optimization for HoopNet or LinearNet, hence we can use either, depending on context. This might form the basis of an interesting interactive tool where a user can apply a set-first optimization to tasks that are wholly or mainly set-oriented, and similarly a network-first optimization to network-oriented tasks. ILP models can be slow to optimize, and it is algorithmically interesting to develop and implement fast optimization heuristics for HoopNet and LinearNet. Lastly, user interface interaction techniques could be used to handle the visual complexity and allow the designs to scale to larger instances. Whilst we consider that data presented in a book or on a static website should not greatly exceed the large data sets we tested in our study, for specialists and within interactive systems, larger datasets may be feasible.

References

1. Alqadah, M., Stapleton, G., Howse, J., Chapman, P.: The perception of clutter in linear diagrams. In: Proc. Diagrammatic Representation and Inference (Diagrams’16). LNCS, vol. 9781, pp. 250–257. Springer (2016). https://doi.org/10.1007/978-3-319-42333-3_20
2. Alsallakh, B., Micallef, L., Aigner, W., Hauser, H., Miksch, S., Rodgers, P.J.: The state-of-the-art of set visualization. *Comput. Graph. Forum* **35**(1), 234–260 (2016). <https://doi.org/10.1111/CGF.12722>
3. Baur, M., Brandes, U.: Crossing reduction in circular layouts. In: Proc. Graph-Theoretic Concepts in Computer Science (WG’04). LNCS, vol. 3353, pp. 332–343. Springer (2004). https://doi.org/10.1007/978-3-540-30559-0_28
4. Borgo, R., Micallef, L., Bach, B., McGee, F., Lee, B.: Information visualization evaluation using crowdsourcing. In: *Computer Graphics Forum*. vol. 37, pp. 573–595. Wiley (2018). <https://doi.org/10.1111/cgf.13444>
5. Burch, M., Huang, W., Wakefield, M., Purchase, H.C., Weiskopf, D., Hua, J.: The state of the art in empirical user evaluation of graph visualizations. *IEEE Access* **9**, 4173–4198 (2021). <https://doi.org/10.1109/ACCESS.2020.3047616>, <https://doi.org/10.1109/ACCESS.2020.3047616>
6. Chapman, P., Sim, K., Chen, H.H.: Minimising line segments in linear diagrams is NP-hard. *J. Comput. Lang.* **71**, 101136 (2022). <https://doi.org/10.1016/J.COLA.2022.101136>
7. Chapman, P., Stapleton, G., Rodgers, P., Micallef, L., Blake, A.: Visualizing sets: An empirical comparison of diagram types. In: Proc. Diagrammatic Representation and Inference (Diagrams’14). LNCS, vol. 8578, pp. 146–160. Springer (2014). https://doi.org/10.1007/978-3-662-44043-8_18

8. Diehl, S., Beck, F., Burch, M.: Uncovering strengths and weaknesses of radial visualizations—an empirical approach. *IEEE Trans. Vis. Comput. Graph.* **16**(6), 935–942 (2010). <https://doi.org/10.1109/TVCG.2010.209>
9. Dobler, A., Nöllenburg, M.: On computing optimal linear diagrams. In: *Proc. Diagrammatic Representation and Inference (Diagrams'22)*. LNCS, vol. 13462, pp. 20–36. Springer (2022). https://doi.org/10.1007/978-3-031-15146-0_2
10. Dogrusöz, U., Madden, B., Madden, P.: Circular layout in the graph layout toolkit. In: *Proc. Graph Drawing (GD'96)*. LNCS, vol. 1190, pp. 92–100. Springer (1996). https://doi.org/10.1007/3-540-62495-3_40
11. Filipov, V.A., Arleo, A., Miksch, S.: Are we there yet? A roadmap of network visualization from surveys to task taxonomies. *Comput. Graph. Forum* **42**(6) (2023). <https://doi.org/10.1111/CGF.14794>
12. Galbraith, S., Daniel, J.A., Vissel, B.: A study of clustered data and approaches to its analysis. *Journal of Neuroscience* **30**(32), 10601–10608 (2010). <https://doi.org/10.1523/JNEUROSCI.0362-10.2010>
13. Goldberg, J.H., Helfman, J.I.: Eye tracking for visualization evaluation: Reading values on linear versus radial graphs. *Inf. Vis.* **10**(3), 182–195 (2011). <https://doi.org/10.1177/1473871611406623>
14. Goodarzi, A.T.: Crossing Reduction in Circular Layouts under Grouping Constraints. Bachelor's thesis, Julius-Maximilians-Universität Würzburg, Würzburg, Germany (2022), <https://www1.pub.informatik.uni-wuerzburg.de/pub/theses/2022-torabi-goodarzi-bachelorarbeit.pdf>
15. Gottfried, B.: A comparative study of linear and region based diagrams. *J. Spatial Information Science* (10), 3–20 (2015). <https://doi.org/10.5311/JOSIS.2015.10.187>
16. Gou, L., Zhang, X.L.: TreeNetViz: Revealing patterns of networks over tree structures. *IEEE Trans. Vis. Comput. Graph.* **17**(12), 2449–2458 (2011). <https://doi.org/10.1109/TVCG.2011.247>
17. Halekoh, U., Højsgaard, S., Yan, J.: The R package geepack for generalized estimating equations. *Journal of Statistical Software* **15**(2), 1–11 (2006). <https://doi.org/10.18637/jss.v015.i02>
18. Kerren, A., Jusufi, I.: A novel radial visualization approach for undirected hypergraphs. In: *Proc. EuroVis 2013 – Short Papers*. Eurographics Association (2013). <https://doi.org/10.2312/PE.EUROVISSHORT.EUROVISSHORT2013.025-029>
19. Keys, C.D.: Graphs critical for maximal bookthickness. *Pi Mu Epsilon Journal* **6**(2), 79–84 (1975), <https://www.jstor.org/stable/24345165>
20. Lee, B., Plaisant, C., Parr, C.S., Fekete, J., Henry, N.: Task taxonomy for graph visualization. In: *Proc. BEyond time and errors: novel evaluation methods for information visualization, BELIV 2006*. pp. 1–5. ACM Press (2006). <https://doi.org/10.1145/1168149.1168168>
21. Leskovec, J., Sosič, R.: Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)* **8**(1), 1–20 (2016). <https://doi.org/10.1145/2898361>
22. Mutton, P., Rodgers, P., Flower, J.: Drawing graphs in Euler diagrams. In: *Proc. Diagrammatic Representation and Inference. (Diagrams'04)*. LNCS, vol. 2980, pp. 182–196. Springer (2004). https://doi.org/10.1007/978-3-540-25931-2_9
23. Neumann, P., Schlechtweg, S., Carpendale, S.: Arctrees: Visualizing relations in hierarchical data. In: *7th Joint Eurographics - IEEE VGTC Symposium on Visualization*. pp. 53–60. Eurographics Association (2005). <https://doi.org/10.2312/VISSYM/EUROVIS05/053-060>

24. Pretorius, A.J., van Wijk, J.J.: Visual analysis of multivariate state transition graphs. *IEEE Trans. Vis. Comput. Graph.* **12**(5), 685–692 (2006). <https://doi.org/10.1109/TVCG.2006.192>
25. Purchase, H.C., Carrington, D.A., Alder, J.: Empirical evaluation of aesthetics-based graph layout. *Empir. Softw. Eng.* **7**(3), 233–255 (2002). <https://doi.org/10.1023/A:1016344215610>
26. Ren, D., Lee, B., Brehmer, M.: Charticulator: Interactive construction of bespoke chart layouts. *IEEE Trans. Vis. Comput. Graph.* **25**(1), 789–799 (2019). <https://doi.org/10.1109/TVCG.2018.2865158>
27. Rodgers, P., Chapman, P., Blake, A., Nöllenburg, M., Wallinger, M., Dobler, A.: Hoop diagrams: A set visualization method. In: *Proc. Diagrammatic Representation and Inference (Diagrams'24)*. LNCS, vol. 14981, pp. 377–392. Springer (2024). https://doi.org/10.1007/978-3-031-71291-3_31
28. Rodgers, P.J., Stapleton, G., Alsallakh, B., Micallef, L., Baker, R., Thompson, S.J.: A task-based evaluation of combined set and network visualization. *Inf. Sci.* **367–368**, 58–79 (2016). <https://doi.org/10.1016/j.ins.2016.05.045>
29. Rodgers, P.J., Stapleton, G., Chapman, P.: Visualizing sets with linear diagrams. *ACM Trans. Comput. Hum. Interact.* **22**(6), 27:1–27:39 (2015). <https://doi.org/10.1145/2810012>
30. Six, J.M., Tollis, I.G.: Circular drawing algorithms. In: *Handbook of Graph Drawing and Visualization*, chap. 9, pp. 285–315. CRC Press (2013)
31. Ståhlbom, E., Molin, J., Ynnerman, A., Lundström, C.: Should I make it round? suitability of circular and linear layouts for comparative tasks with matrix and connective data. *Comput. Graph. Forum* **43**(3) (2024). <https://doi.org/10.1111/CGF.15102>
32. Tubbs, J.D., Ding, J., Baum, L., Sham, P.C.: Immune dysregulation in depression: Evidence from genome-wide association. *Brain, Behavior, & Immunity-Health* **7**, 100108 (2020). <https://doi.org/10.1016/j.bbih.2020.100108>
33. Vehlou, C., Beck, F., Weiskopf, D.: Visualizing group structures in graphs: A survey. *Comput. Graph. Forum* **36**(6), 201–225 (2017). <https://doi.org/10.1111/CGF.12872>
34. Vehlou, C., Hasenauer, J., Kramer, A., Raue, A., Hug, S., Timmer, J., Radde, N., Theis, F.J., Weiskopf, D.: ivun: interactive visualization of uncertain biochemical reaction networks. *BMC Bioinform.* **14**(S-19), S2 (2013). <https://doi.org/10.1186/1471-2105-14-S19-S2>
35. Waldner, M., Diehl, A., Gracanin, D., Splechtna, R., Delrieux, C., Matkovic, K.: A comparison of radial and linear charts for visualizing daily patterns. *IEEE Trans. Vis. Comput. Graph.* **26**(1), 1033–1042 (2020). <https://doi.org/10.1109/TVCG.2019.2934784>
36. Wattenberg, M.: Arc diagrams: visualizing structure in strings. In: *Information Visualization (INFOVIS'02)*. pp. 110–116. IEEE (2002). <https://doi.org/10.1109/INFVIS.2002.1173155>