Paper:
Wang, Y., Shen, Q., Archambault, D., Zhou, Z., Zhu, M., Yang, S. & Qu, H. (2016). AmbiguityVis: Visualization of Ambiguity in Graph Layouts. *IEEE Transactions on Visualization and Computer Graphics (InfoVis 2015), 22*(1), 359-368.

http://dx.doi.org/10.1109/TVCG.2015.2467691

AmbiguityVis: Visualization of Ambiguity in Graph Layouts

Yong Wang, Qiaomu Shen, Daniel Archambault, Zhiguang Zhou, Min Zhu, Sixiao Yang, Huamin Qu

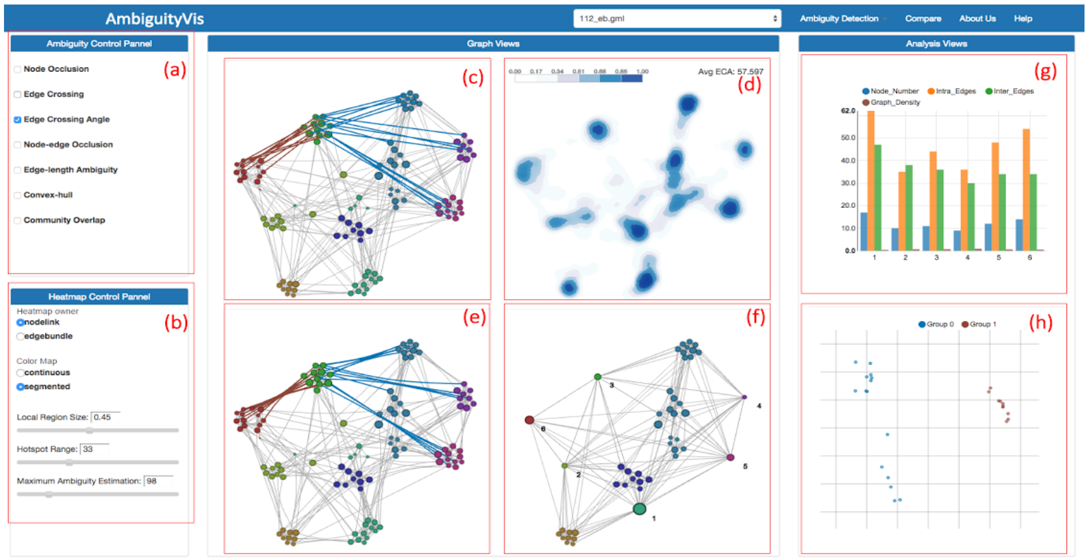

Fig. 1: User interface of AmbiguityVis showing the ambiguities existing in graph layouts. It consists of eight parts: (a) Ambiguity control panel allowing users to select the type of detected ambiguity. (b) Heatmap control panel enabling users to adapt the parameters of the heatmap such as hotspot range and local region size used in entropy and autocorrelation. (c) The node-link diagram layout. (d) Heatmap view for the selected type of ambiguity in the graph layout. (e) Edge bundling view. (f) Node aggregation view. (g) Bar chart view used to inform users of the underlying information in each metanode. (h) Scatterplot view generated by MDS to show the consistency of the bundled edges.

Abstract—Node-link diagrams provide an intuitive way to explore networks and have inspired a large number of automated graph layout strategies that optimize aesthetic criteria. However, any particular drawing approach cannot fully satisfy all these criteria simultaneously, producing drawings with visual ambiguities that can impede the understanding of network structure. To bring attention to these potentially problematic areas present in the drawing, this paper presents a technique that highlights common types of visual ambiguities: ambiguous spatial relationships between nodes and edges, visual overlap between community structures, and ambiguity in edge bundling and metanodes. Metrics, including newly proposed metrics for abnormal edge lengths, visual overlap in community structures and node/edge aggregation, are proposed to quantify areas of ambiguity in the drawing. These metrics and others are then displayed using a heatmap-based visualization that provides visual feedback to developers of graph drawing and visualization approaches, allowing them to quickly identify misleading areas. The novel metrics and the heatmap-based visualization allow a user to explore ambiguities in graph layouts from multiple perspectives in order to make reasonable graph layout choices. The effectiveness of the technique is demonstrated through case studies and expert reviews.

Index Terms—Visual Ambiguity, Visualization, Node-link diagram, Graph layout, Graph visualization

1 Introduction

Node-link diagrams are commonly used to visualize networks in many areas including: social networks, financial transaction networks, biological networks, and others. In order to present the underlying structure of the network visually, the development of automated graph drawing algorithms has been an active area of research for decades, and many approaches have been proposed [4, 22, 18]. Generally, graph drawing approaches optimize heuristics such as readability metrics that approximate the visual properties desired in graph drawings. For example, force-directed approaches reduce node occlusions and edge crossings via physical analogies. Multi-dimensional scaling (MDS) based approaches optimize the pairwise Euclidean distance between nodes to better reflect graph-theoretic distance. However, no graph layout algorithm can satisfy all the desired metrics simultaneously.

When drawing a graph, finding an optimal solution for many of these readability metrics is NP-hard [4]. Therefore, trade-offs need to be made. In this case, ambiguities or misleading information can occur in the drawing and such ambiguities can impede the comprehension of network structure. For instance, node occlusions and edge crossings make it difficult for users to understand the relationships in a graph. In addition, understanding which nodes belong to which community

- *Y. Wang, Q. Shen and H. Qu are with the Hong Kong University of Science and Technology. E-mail: {ywangct, shenqiaomu, huamin}@ust.hk*
- *D. Archambault is with Swansea University. E-mail: d.w.archambault@swansea.ac.uk*
- *M. Zhu is with Sichuan University and is the corresponding author. E-mail: zhumin@scu.edu.cn*
- *Z. Zhou is with Zhejiang University of Finance and Economics. E-mail: zhgzhou1983@zufe.edu.cn*
- *S. Yang is with Huawei Co. Ltd. E-mail: yangsixiao@huawei.com*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

can be difficult when community structures in a graph drawing visually overlap. When the graph to be drawn becomes very large, severe visual clutter might occur no matter which graph drawing algorithm is used. Aggregation methods, such as edge bundling [16, 25, 23] and simplification through metanodes [1, 2, 42], are often used to partially reduce the visual clutter. However, graph details are hidden and visual ambiguities can occur. In the case of edge bundling, consider Fig. 2: once the edges are bundled (Fig. 2(a)), it is difficult to identify whether the graph structure is Fig. 2(b) or Fig. 2(c). Similarly, when parts of the graph are replaced by metanodes in a compound graph visualization, the content of the metanodes is hidden from the viewer, reducing visual complexity. However, two metanodes can have the same visual appearance but represent subgraphs of completely different structures. Such ambiguities in graph drawings might cause problems for both developers and analysts. Developers of graph visualization systems often need to select a suitable layout from many possible options. They should be made aware of potential ambiguities and misleading information present in a visualization. A technique to reveal these ambiguities will help developers make appropriate choices to better support the visualization requirements of end users. Of equal importance is the need to warn analysts who are trying to read the drawings of potential problem areas in graph layouts.

However, the problem of visualizing ambiguities in graph layouts has received little attention and very few solutions are available. To visualize ambiguities, we first need to identify different kinds of ambiguity cases and then develop methods to quantify them. Some metrics are available, for example, Dunne *et al.* [9, 10] presented a summary of *global readability metrics* and introduced new *node and edge readability metrics*. However, these metrics focus on ambiguities resulting from the spatial relationships between nodes and edges in graph drawing. Devising metrics to quantify ambiguities in community structures and node/edge aggregation methods both remain open problems.

In this paper, we present AmbiguityVis, a new visual analytics technique to reveal ambiguities in graph drawings. AmbiguityVis targets three common classes of ambiguities: ambiguities in the spatial relationship between nodes and edges, visual overlap between community structures, and visual ambiguities resulting from node/edge aggregation. We propose a set of metrics to quantify these ambiguities and then design heatmap-based visualizations of these metrics to highlight these areas of ambiguity. The technique can help developers of graph visualization systems in designing and selecting appropriate graph drawing and visualization approaches and can help analysts better understand the drawings. We test our technique with two case studies and also conduct expert reviews to collect feedback.

The major contributions of this paper are as follows:

- New metrics to quantify visual ambiguities resulting from abnormal edge lengths, visual overlap between communities, and node/edge aggregation.
- A method for creating heatmap-based visualizations of these new ambiguity metrics.
- Case studies and expert reviews that provide support for the effectiveness of the technique.

2 Related Work

We divide related work into three sections: graph drawing and aggregation methods, graph drawing readability metrics, and metrics for the separability of communities.

2.1 Graph Drawing and Aggregation

Graph drawing, which aims at designing layout algorithms for networks, has a history of more than fifty years [48] with many books [4, 31, 46] and surveys [22, 52, 18] written on the topic. According to Gibson *et al.* [18], algorithms for automatic graph layout can be placed into three categories: force-directed, dimensionality reduction based, and computational improvements (e.g. multi-level techniques). Force-directed approaches [14, 26, 29] model the graph as a physical system in order to optimize graph drawing aesthetics. Dimensionality reduction based methods often mirror multi-dimensional scaling [30] and other linear dimensionality reduction methods [5]. Computational improvement approaches create a hierarchy of coarse graphs and use this hierarchy to speed up the drawing of the graph (e.g. [20, 21]).

When a network becomes large, aggregation methods, mainly edge bundling [23, 25] and compound graph visualizations [42, 1, 2, 3], are used to reduce its visual complexity. Edge bundling aggregates edges with similar properties into bundles, whereas compound graph visualizations abstract subgraphs into hierarchies of metanodes to support exploration. Dunne and Shneiderman proposed a motif simplification where subgraphs are replaced with compact and informative glyphs [11].

However, as mentioned previously, optimal layouts for all metrics rarely exist and all of the above techniques can produce visual ambiguities. Therefore, a technique that helps in the design of graph layout algorithms and helps users understand the visual ambiguities present in specific graph drawings and visualizations would be helpful to the community.

Techniques have been developed to visualize node density in very large graphs through splatting [50]. By converting the nodes in the drawing into a continuous field, GraphSplatting is able to aggregate dense areas of large graphs. Although the heatmaps in GraphSplatting share some visual similarities with our work, our heatmaps serve a different purpose (i.e., revealing ambiguities in graph layouts) and are created differently (i.e., based on readability metrics).

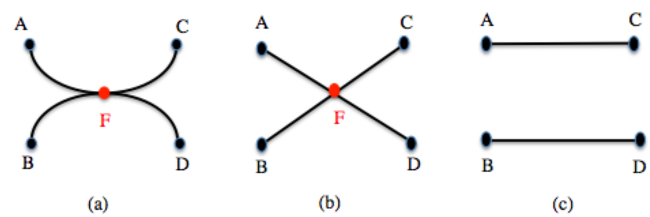

Fig. 2: Ambiguity resulting from edge bundling: after the edges are bundled as in (a), it is difficult to tell if the graph structure is (b) or (c).

2.2 Graph Drawing Readability Metrics

Similar to Dunne and Shneiderman [10], this paper uses "readability metrics" instead of "aesthetic criteria" to represent how well the drawing reflects the graph structure. Obviously, ambiguity or misleading information occurs when the graph has low readability. Previous work has evaluated readability from two perspectives: human-centered evaluations and metric evaluations.

Human-centered evaluations test the readability of graphs on specific tasks with users. Huang [27] performed eye tracking studies to investigate the influence of crossing angles and the geometric path tendency in graph layouts. Pohl *et al.* [38] compared the readability of force-directed, orthogonal, and hierarchical methods through a user study involving eye tracking and task-oriented analysis. Marriott *et al.* [35] performed a re-drawing experiment, comparing the cognitive impact of layout features such as symmetry, alignment of nodes, and collinearity on human recall of graph layouts. Some studies on graph drawing aesthetics have asked participants to create layouts that best reflect the network structure [41, 49]. Holten *et al.* [24] compared the readability of various directed edge encodings for node-link diagrams.

Quantitative metrics provide numerical scores that approximate graph readability and are usually validated through human-centered evaluations. Purchase [39] introduced seven global layout metrics based on graph drawing aesthetics. Ellis and Dix [12] defined three metrics for clutter and density based occlusion of plotted points. Dunne *et al.* [10, 9] investigated both global and local readability metrics, focusing on the following:

- Node occlusion: proportion of pairwise node overlap.
- Edge crossings: number of pairwise edge crossings.
- Edge crossing angle: deviations from an ideal crossing of 70°.
- Edge tunnel: number of edges that pass under an unrelated node.
- Group overlap: number of nodes that are not part of a node group A but lie inside the convex hull of A (global only).

Building off the work of Dunne *et al.* [9, 10], this paper focuses on new ambiguities in graph layouts such as ambiguities resulting from abnormal edge lengths, visual overlap between community structures and node/edge aggregation, which have not been fully investigated before. We also develop novel techniques for ambiguity measurement such as entropy and autocorrelation based methods and introduce new visualization approaches for showing ambiguities in the layout.

2.3 Measuring the Visual Separability of Communities

Communities, or clusters, are highly connected groups of nodes. Effective layouts of communities can convey higher level structures in the network. When nodes from multiple communities visually interleave in the same area of the drawing, it is difficult to read these clusters. Thus, drawings that can separate out these communities help clarify graph structure.

In order to evaluate how well clusters are separated in low dimensional representations of high dimensional data, measures of cluster quality have been developed. Sips *et al.* [45] proposed two class consistency measures: one is based on the distance from the cluster center of gravity and the other is based on the entropy present in the spatial distribution of the nodes from different classes. Tatu *et al.* [47] automated the ranking of scatterplots for both classified and unclassified data by taking the correlation and cluster separation into account. The CH measure [6] and the Silhouette measure [43] can also be used, and the effectiveness of both measures has been supported by human-centered evaluations [33]. Dunne *et al.* [9] proposed a quantitative measure based on the convex hulls of clusters to evaluate cluster separability.

In this paper, we extend these entropy-based measures [45, 47], originally designed for scatterplots, and propose new metrics to measure the ambiguities in community structures. When compared with the convex hull method [9], our metrics can reveal more details of the ambiguity present in graph layouts.

3 Ambiguity Detection

AmbiguityVis aims at detecting and visualizing potential ambiguities in a graph layout. Different types of ambiguities can be present, and some of them have been considered in previous work [9, 10]. However, as mentioned in the last section, ambiguities resulting from abnormal edge lengths, visual overlap between community structures and node/edge aggregation have not been fully investigated. These ambiguities will prevent users from interpreting the drawings correctly and performing various analytical tasks, especially group-level tasks [44]. In this section, we introduce these ambiguities and present novel metrics to quantify them. AmbiguityVis can then use these metrics, as well as existing ones, to detect and visualize potential ambiguities in graphs.

3.1 Abnormal Edge Length Ambiguity

When two nodes are placed close to each other in a drawing, humans tend to believe that there is a close relationship between the nodes, whether this relationship actually exists or not [36]. Conversely, if two nodes are closely related in a graph-theoretic sense, but are placed distant from each other in a Euclidean sense, users tend to believe that the two nodes are distant in the network [13]. Therefore, long edges are misleading and can introduce ambiguities. Despite this evidence from human-centered evaluations, few metrics have been introduced to quantify this type of ambiguity. Davis and Hu [7] compared the length of an edge to the average edge length in the layout. However, this metric does not closely approximate either ambiguity described above.

To address this issue, we propose a novel metric that better approximates the ambiguity resulting from abnormal edge lengths. Given a network with N nodes and an edge with endpoints i and j, let us denote the degree of node i as d_i and the set of nodes directly connected to i as $D(i)$. The same quantities for node j are defined as d_j and $D(j)$ respectively. Suppose node j is the r_{ij}-th nearest neighbor of node i in terms of Euclidean distance. If $r_{ij} < d_i$, we regard edge ij as not ambiguous with respect to node i. Otherwise, we say that it is ambiguous. A similar analysis can be performed from the perspective of node j. Therefore, the local metric for edge length ambiguity at each edge is defined as follows:

$$h = 0.5 * u(r_{ij} - d_i)\frac{r_{ij} - d_i}{N - d_i} + 0.5 * u(r_{ji} - d_j)\frac{r_{ji} - d_j}{N - d_j} \tag{1}$$

where $i, j = 1, 2, ..., N$, $i \neq j$. The local metric for edge length ambiguity at each node is defined as:

$$h = \frac{\sum_{j \in D(i)} u(r_{ij} - d_i) * \frac{(r_{ij} - d_i)}{N - d_i}}{d_i} \tag{2}$$

where $i = 1, 2, ..., N$, and $u(t)$ is the Heaviside step function:

$$u(t) = \begin{cases} 1 & t > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The ambiguity caused by abnormal edge lengths is a special case of ambiguous spatial relationships. The possible ambiguities in spatial relationships between nodes and edges also include: **node occlusion**, **edge crossing**, **edge crossing angle**, and **node-edge occlusion**. These low-level ambiguities are generally accepted to have an effect on the readability of node-link diagrams [27, 40, 28]. The metrics to quantify these ambiguities have been proposed before [9, 39]. In AmbiguityVis, we also use these metrics:

$$h = \begin{cases} o & \text{for node occlusion} \\ c & \text{for edge crossing} \\ |70 - \theta| & \text{for edge crossing angle} \\ n_e & \text{for node-edge occlusion} \end{cases} \tag{4}$$

where o is the number of nodes that have partial or complete overlap with a particular node, c represents the number of edge crossings with a particular edge, θ represents the acute angle crossing with a particular edge, and n_e represents the number of node-edge occlusions (both *edge tunneling* and *edge bridge*) with a particular node. The node occlusion metric does not differentiate between partial and total node occlusion as in Dunne *et al.* [9]. This simplification speeds up the calculation and is precise enough to localize this ambiguity as supported by the expert review and case studies in Section 6.

For the corresponding global metrics of the above ambiguities, each global metric is defined as the average ambiguity value among all the nodes or points of edge crossings in AmbiguityVis.

3.2 Ambiguity in Community Structure

Detecting community structures in networks has received considerable attention [32]. Partitioning a network into different communities can help improve the readability of a graph drawing. User experimentation has shown that a clear, visual separation of community structures is preferred, even if it means sacrificing uniform edge lengths in the drawing [49]. However, for large and dense networks, it is difficult to clearly separate all communities and visual overlap between different communities can occur. To inform users of the visual overlap between communities, Dunne *et al.* [9] proposed a metric based on convex hulls. Intuitively, we can also overlay these convex hulls on the communities to visually show the ambiguity. These methods can offer a rough overview about the visual overlap between different communities, but details about the degree and distribution of the ambiguity cannot be precisely represented. For example, the communities in Fig. 3(a) and Fig. 3(b) have similar convex hulls, but the degree of the visual overlap between the red and blue communities differs substantially. Users want to know the variation of the overlap for the whole joint-region of the two convex hulls (e.g, some subregions might not have any overlapping at all).

To address this issue, we propose two metrics, an entropy-based metric and an autocorrelation-based metric, to detect and quantify the visual overlap between communities.

3.2.1 Entropy-based Metric

Entropy is a widely-used measurement that quantifies the distribution of random variables in information theory and has been used to analyze the separability of clusters in a scatterplot. Inspired by these works [45, 47], we design an entropy-based metric to measure the degree of visual overlap between communities, as defined in Eq. (5):

$$H = -\sum_{c=1}^{M} p(c)\log_2 p(c) \tag{5}$$

where M is the number of communities, $p(c)$ indicates the percentage of nodes from community c. The area considered is a square of configurable width centered at a specific node. If this region contains nodes from only one community, the entropy value would be 0, meaning that the community structure is represented perfectly. If the region contains an equal mixture of all communities, the entropy reaches a maximum value, meaning that the community structure is not represented at all.

This entropy value is calculated for each node of the graph to measure the visual overlap between communities. The corresponding global metric value is computed by calculating the average value of local metrics at all the nodes.

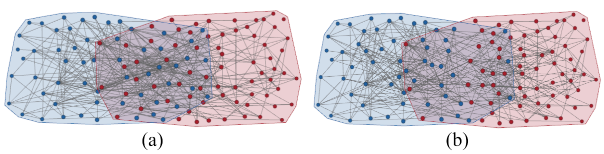

Fig. 3: Convex hulls do not offer precise details about the degree and distribution of ambiguities present in graph layouts.

3.2.2 Autocorrelation-based Metric

Local entropy, as defined above, can quantify the amount of visual overlap to some extent. However, it does not take the community label of its central node into account when comparing this node to others in the region. Spatial autocorrelation can be used to analyze the node distribution in a graph layout and can quantify the degree to which observations of the same phenomenon are correlated. Geary's C [17] is a widely used spatial autocorrelation measure that has been applied to many areas such as geographical data analysis.

Geary's C cannot be immediately applied to analyze the distributions of communities in a graph layout because it cannot handle categorical data such as the community classification of a node. Therefore, we adapt this metric to measure the community distribution in a region surrounding node i (Eq. (6)):

$$C_i = \frac{\sum_{j=1}^{N}(1-w_{ij})(X_i - X_j)}{\sum_{j=1}^{N}(1-w_{ij})} \tag{6}$$

where N is the number of nodes present in the region centered at node i, X_i and X_j are the respective community identifiers of node i and node j. If they are from the same community, $X_i - X_j$ is set to 0. Otherwise, $X_i - X_j$ is set to 1. The spatial coefficient w_{ij} is the normalized distance between node i and node j. When all the nodes surrounding node i come from the same community as node i, then $C = 0$, which indicates a perfect preservation of community structure. However, when nodes surrounding node i come from different communities, the closer the nodes are to i, in terms of Euclidean distance, the more community structure degrades from the perspective of i.

Similar to the entropy-based metric, the value of the autocorrelation-based metric is calculated for each node of the graph and visualized as will be described in Section 4. The global version of this metric is defined as the average value of all the nodes in the graph.

Finally, the two metrics defined above require a community label for each node of the graph. Therefore, they need a community assignment as input. If the assignment of communities to nodes in the graph is completely divorced from graph structure, the graph layout will not reflect these assigned communities that well. Thus, misleading visual feedback for the community structure ambiguity may occur. In this paper, we use modularity maximization [37] to detect communities based on graph connectivity and to label each node of the graph accordingly.

3.3 Ambiguity in Node/Edge Aggregation

In this section, we present novel metrics for evaluating the ambiguity present in edge bundling and metanode-based aggregation.

3.3.1 Edge Bundling

Many types of edge bundling algorithms have been proposed to deal with visual clutter. These algorithms can show the overall structure of the graph, but may not always accurately display the relationship between node pairs [23, 34], which can affect human performance on a number of graph-based tasks such as path tracing. Therefore, quantifying the ambiguity in edge bundling is an important and unsolved problem. We consider two questions when examining the ambiguity of edge bundling: (1) **How consistent are the edges in the bundle?** (2) **How clearly are the edges bundled together?**

For the first question, we consider similar factors such as distance, edge length similarity and parallelism used in previous work to design edge bundling algorithms [25]. First, nearby edges instead of spatially distant ones should be bundled together. Second, edges of similar lengths instead of edges of different lengths should be bundled together. Finally, parallel edges are more suitable for bundling than perpendicular edges. Thus, nearby edges, edges of similar lengths, and parallel edges will cause less ambiguity in the layouts.

For the second question, when edges touch in a bundle, connectivity can become ambiguous. As shown in Fig. 2, it is difficult to disambiguate the two cases. Touching edges can be considered as a type of intersection. Another important factor is edge curvature. Xu *et al.* [55] conducted two studies to determine the impact of edge curvature on graph readability. The studies found that straight edges can clearly show the relationship between nodes in the graph, while for curved edges, the relationship becomes harder to understand with increased curvature.

Guided by the two questions above, we quantify the ambiguity present in edge bundling based on **the consistency of the bundled edges** and **the clarity of each bundle**. We propose three novel metrics to evaluate the consistency of the bundled edges based on distance, edge length similarity and parallelism.

Fig. 4: The disadvantage of parallel measurement by angle: case (a) and case (b) cannot be distinguished.

Distance: Suppose $Dist(P,AB) = min||PQ||, \forall Q \in AB$, where AB is a line segment. We define the distance between edge AB and CD as follows:

$$D = \max\{Dist(M,AB), Dist(N,CD)\} \tag{7}$$

where M is any point on the edge CD and N is any point on edge AB. The corresponding measure defined by Holten and van Wijk [25] is based on the distance between the midpoints of two edges. In contrast, the metric defined above takes all the points of both edges into consideration and can better reflect this distance.

Edge length similarity: the edge length similarity between edge AB and edge CD is defined as:

$$L_{sim} = \frac{\min(len(AB), len(CD))}{\max(len(AB), len(CD))} \tag{8}$$

where $len()$ represents the function that returns the length of the edge.

Parallelism: To evaluate how parallel the pairs of edges in the bundle are, we could calculate the angles between the edges [25]. However, this measure cannot effectively distinguish between the two configurations shown in Fig. 4. As an alternative, the projection is used to compute how parallel AB is to CD:

$$P = \frac{len(A'B')}{len(CD)} \tag{9}$$

where $A'B'$ is the projection of AB on CD. We quantify how parallel AB is to CD as the minimum of the projection from AB to CD and the projection from CD to AB.

We apply multidimensional scaling to integrate all the three metrics and produce a 2D scatterplot (Fig. 9(b)(e)) to visually show the consistency of bundled edges. The aggregated "consistency" between any pair of edges is defined as:

$$con = \sqrt{\alpha * D^2 + \beta * (1 - L_{sim})^2 + \gamma * (1 - P)^2} \tag{10}$$

where α, β and γ are weights for the three metrics defined above with $\alpha + \beta + \gamma = 1$. These weights can be configured by users in our prototype implementation of AmbiguityVis, and the default values are set to $\frac{1}{3}$. D is normalized to $[0, 1]$ based on the maximum of the distances in the selected bundles.

Clarity of the bundle. To evaluate the clarity of the bundled edges, both curvature and intersection are considered. We define curvature as the amount of deviation from a straight line. As an estimate of curvature, we compute the ratio between the length of the curved edge and the corresponding straight line between the endpoints of the edge. For example, suppose the relationship of the bundled edges in Fig. 2(a) is as depicted in Fig. 2(c). The curvature for the edge $\widehat{\mathrm{AC}}$ would be estimated as follows:

$$C_l = \frac{len(\widehat{\mathrm{AC}}) - len(AC)}{len(AC)}. \tag{11}$$

For the intersection between bundled edges, we can compute all points of intersection as defined above, but this process can be complex and inefficient. Therefore, we do not explicitly compute these intersections but use the accumulation in the heatmap, based on kernel density estimation, to visualize them.

We combine edge curvature and intersection to show the ambiguity present in edge bundling. Every curved edge in a bundle can be approximated by sampling a series of points on that edge. Eq. (11) computes the curvature at each sample point on the edge. These values will be mapped to different colors (see Section 4). When the edges intersect or dense edges are bundled together, the ambiguity will naturally accumulate and the color of this region will increase in saturation, indicating higher severity. Therefore, the result is able to show these ambiguities (Fig. 10(b)).

3.3.2 Node Aggregation

Node aggregation replaces subgraphs with metanodes in order to simplify graph structure [1, 2, 42]. According to the taxonomy of Vehlow *et al.* [51], node aggregation is one of the main techniques to visualize group structures in graphs. Saket *et al.* [44] provided a classification for group-level tasks in graph visualization: *group-only, group-node, group-link* and *group-network tasks*. Node aggregation is effective in reducing the amount of visual clutter in a network and providing an overview of its structure. Thus, node aggregation can be useful for *group-only tasks*. However, metanodes hide subgraph details from the user and can exhibit the same visual appearance regardless of its contents. Therefore, the ambiguity associated with metanodes can affect the user's performance on the remaining three tasks.

Motif simplification [11] partially mitigates this problem by providing a meaningful glyph to indicate the content of a metanode. However, the glyphs are limited to a small number of specific motifs, and this approach can have limitations in terms of scalability. For very large graphs, metanodes with similar circular shapes are often used, which cause ambiguities, because visually similar metanodes could represent drastically different subgraphs. Thus, in this work, we provide a way to visualize some basic information about the subgraph represented by a particular metanode, informing users of possible ambiguities resulting from the hidden details.

Our solution is to provide statistics about the subgraph contained inside a metanode including the number of nodes, intra-cluster edges, and inter-cluster edges. We also consider graph density, which is calculated by comparing the number of edges in the subgraph to the number of edges in a complete graph with the same number of nodes. These four measures are visualized using bar charts as shown in Figs. 11(b)(d) with each measure encoded in bars of a different color and the x-axis representing the identifier of each metanode. Users can interactively select the statistics displayed. This interaction provides an overview of the contents of the metanodes or details about each metanode from the perspective of a particular statistic.

4 Ambiguity Visualization

The main purpose of our visualization technique is to highlight areas of the graph layout that have more serious problems in terms of the visual ambiguities that we quantify. Dunne and Shneiderman [10] provided a solution by highlighting nodes with a particular ambiguity in red. However, in large graphs, node and edge occlusion in the layout limit the scalability of this approach. In AmbiguityVis, we employ a continuous, rather than discrete, approach to localize areas of ambiguity through a heatmap. Heatmaps can display how a particular attribute is distributed in spatial regions and have been used to visualize gaze distributions [8, 38], recommendation information [15] and node density in a node-link diagram [50]. A possible alternative to the heatmap would be a view similar to GraphScape [54], but GraphScape has the disadvantage of surface occlusion. Thus, a heatmap visualization was finally chosen for AmbiguityVis. In AmbiguityVis, we use a segmented white-to-blue color map as shown in Fig. 5. Saturated blue indicates areas with serious ambiguities according to a particular metric while white indicates little to no ambiguity. Given a value of a particular metric computed on the nodes/edges of the graph, we compute the heatmap as follows:

Hot Spot Position: We assign a hot spot to the point where the specific ambiguity is defined. Thus, the position of each hot spot is the center of the specific node or edge crossing, depending on the metric.

Hot Spot Range: This range defines how much a hot spot influences its surrounding area.

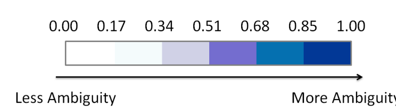

Fig. 5: Segmented white-to-blue color map used in the heatmap.

The ambiguity values of points around a hot spot are computed based on kernel density estimation (KDE). Thus, the ambiguity value at a point in the heatmap with a number of hotspots around it will accumulate, resulting in a relatively high value of ambiguity. All ambiguity values are normalized to $[0, 1]$ using a maximum ambiguity value set by the user. After normalization, the range of ambiguity values is segmented into K sequential levels ($K = 6$) to denote the levels of severity as shown in Fig. 5.

The hot spot range and maximum ambiguity value used by the normalization procedure can be interactively modified by the user. Using a wide range provides a general overview of ambiguous areas while using a small range can help localize areas with specific levels of severity. The heatmap is only used for local ambiguities defined at nodes or points of crossing. For ambiguities in edge length, the metric can be visualized on both the endpoints of the edge and the edge itself. Visualizing the edge length ambiguity on the endpoints of an edge can inform users that there are spatially distant nodes from a particular node but directly connected, while visualizing this ambiguity on the edge itself can highlight edges with severe ambiguity directly. When the metric is displayed on the edges of the graph, the edge color ranges from white to black where white indicates no ambiguity and black indicates serious ambiguity.

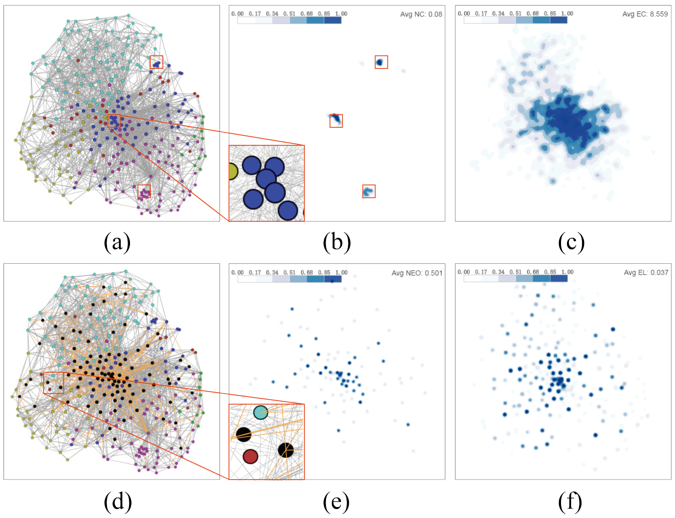

Fig. 6: Visualization feedback of ambiguities resulting from spatial relationship among nodes and edges in the layout of neural network data set. (a) Original graph layout. (b) Heatmap showing the areas with node occlusion. (c) Heatmap view showing the distribution of edge crossings in (a). (d) Original graph layout in which the nodes with node-edge occlusion are highlighted in black and edges in yellow. (e) Heatmap view showing the regions with node-edge occlusion ambiguity. (f) Heatmap view showing the edge length ambiguity.

5 Implementation

The ambiguity detection and visualization methods discussed above have been implemented in a prototype system. It takes the graph structure, layout, and a community partition as input. In our case studies, we mainly use graph layouts generated by *Gephi*. Fig. 1 shows the user interface. Figs. 1(a)(b) show the control panel where a user can select the type of ambiguity to be shown. Users can set parameters such as the local region size used in the entropy and autocorrelation based metrics or the hotspot range and maximum ambiguity estimation used in the heatmap. Figs. 1(c)(e)(f) show the original node-link diagram, the layout with edge bundling, and the layout with node aggregation respectively. The heatmap view (Fig. 1(d)) is the central view and displays the severity of the ambiguities present in the graph layout. The value of each global ambiguity metric is shown in the top-right corner of heatmap view to provide users with a global average of the ambiguity in the graph layout. The bar chart view (Fig. 1(g)) is used to inform users of the underlying information in each metanode. The scatterplot view (Fig. 1(h)), generated by MDS, shows the consistency of the bundled edges in the selection.

After the graph layout data has been loaded, users can explore the details of the ambiguity in the graph layout interactively. AmbiguityVis provides many interactive operations, including:

`Global Configuration` Users can select the type of ambiguity metrics that will be computed using the main menu and select which of these metrics will be shown in the heatmap view through the ambiguity control panel. Also, users can interactively set the parameters of the heatmap.

`Ambiguous Area Location` A region of interest can be selected by brushing a rectangle on the heatmap view (Fig. 1(d)). Linked highlighting indicates the corresponding regions of interest in Fig. 1(c) and Fig. 1(e). When the edge length ambiguity is highlighted on the endpoints of the edge, the user can brush a local region in the heatmap (Fig. 1(c)). The corresponding edges of abnormal length will be highlighted automatically in the node-link diagram (Fig. 1(e)). To explore the ambiguity details, users can pan and zoom in the node link view (Fig. 1(c)), the heatmap view (Fig. 1(d)), or edge bundling view (Fig. 1(e)). These interactions are synchronized in all the three views.

`Bundled Edge Selection` To observe the consistency of the bundled edges, AmbiguityVis allows users to select two edge clusters in the edge bundling view (Fig. 1(e)) by brushing on the curved edges, which are marked in brown and blue respectively. The corresponding edges in the node-link diagram view (Fig. 1(c)) are highlighted in the same color, and scatterplot view (Fig. 1(h)) is updated.

`Manual Layout Improvement` Users can interactively move nodes in the node-link diagram (Fig. 1(c)) to improve the layout. The change in the values of the ambiguity metrics is simultaneously updated in the heatmap (Fig. 1(d)).

`Interactive Metanode Generation` Users can load a graph with predefined metanodes, but they can also select subgraphs in the layout and generate new metanodes (Fig. 1(f)).

`Convenient Graph Layout Comparison` AmbiguityVis enables users to compare up to three different layouts of the same network side-by-side.

6 Evaluation

To demonstrate the effectiveness of AmbiguityVis, we conducted case studies and expert reviews which we describe in this section. The case studies are presented in two parts. The first considers both low-level ambiguities and community structure ambiguities while the second considers ambiguities in node/edge aggregation.

6.1 Case Studies for Low-level and Community Structure Ambiguities

For low-level and community structure ambiguities, AmbiguityVis helps in two cases: locating the problematic areas in single graph layout or selecting the best graph layout for a specific network from several layout algorithms. In the following section, we present the results of these two case studies.

6.1.1 Scanning Single Graph Layout

The first case study aims at demonstrating the effectiveness of AmbiguityVis by showing potential low-level and community structure ambiguities in single graph layout. The data set used in this case study is a neural network [53] consisting of 297 nodes and 2148 edges. We use ForceAtlas2 [29], which is a force-directed graph layout algorithm implemented in *Gephi* to generate the graph layout. We take the graph layout as input into AmbiguityVis. Similar to a computed tomography (CT) used by a doctor to scan a patient's body, users can use AmbiguityVis to "diagnose" the problems in a graph layout from different perspectives and easily recognize problematic areas.

The detailed results of AmbiguityVis are shown in Figs. 6 and 7. In the original graph layout (Fig. 6(a)), the color of each node represents its community type, and many nodes are distributed within limited screen space. In particular, we would like to examine node occlusion in the drawing. Manually searching the whole graph layout will be time-consuming, whereas brushing on the ambiguous parts indicated in the heatmap view (Fig. 6(b)) reveals the locations of significant node-occlusion in the drawing. Fig. 6(c) shows the distribution of edge crossings in the diagram, indicating that the center part of the view has a large number of crossings and thus more ambiguities. When considering node-edge occlusion, the affected edges are highlighted in yellow and the nodes are black, as shown in Fig. 6(d). Fig. 6(e) provides a clearer view of the degree and location of the node-edge occlusion ambiguity, which would otherwise be difficult to achieve without AmbiguityVis. With AmbiguityVis, we find that some nodes are even inappropriately placed on two or three edges. Fig. 6(f) highlights the nodes connecting edges with highly abnormal edge lengths, informing users that the edges connecting these nodes are generally too long and do not reflect the actual graph-theoretic distance between nodes.

The heatmap views (Figs. 7(c)(d)) clearly illustrate the details about the ambiguity in terms of visual overlap between different communities, while the convex hull view can offer an overview of the general structure and overlap of communities (Fig. 7(b)). The region marked as rectangle 1 in both Fig. 7(c) and Fig. 7(d) shows that the center part of the drawing has severe visual community overlap, which is confirmed in Fig. 7(a) since several communities interleave with each other to a high degree. However, there are other regions with definite visual overlap not highlighted in the heatmap based on the entropy metric. For example, nodes from the red community are interleaved with other communities (marked as rectangles 2 and 3), and similar cases can also be found in rectangles 4 and 5. In contrast, the heatmap based on autocorrelation (Fig. 7(d)) is able to highlight these regions. Thus, we advise users to use the autocorrelation-based method as it is more sensitive to the visual overlap between different communities. In summary, users can locate problems existing in a layout easily and clearly with the help of AmbiguityVis.

6.1.2 Selecting Suitable Graph Layouts

In this case study, we use AmbiguityVis to help us select the most suitable graph layout for a specific network. We use a football network data [19] representing the network of American football games between Division IA colleges during the regular season in 2000. The network contains 115 nodes and 613 edges. We use *Gephi* to generate three force-directed graph layouts: ForceAtlas2 with LinLog mode enabled (FA2+LinLog) [29], FR [14], and Hu's method (HU) [26] shown in Fig. 8. The same parameter settings are used for all three heatmaps. We can see that the layout of FA2+LinLog has advantages in preserving the community structures indicated by the autocorrelation-based metric and convex hull, as shown in the "ACA" row and the "CH" row of Fig. 8. This is not surprising, considering that LinLog is good at preserving cluster structure in graphs [29] and this property can also be easily seen in the original layout. The visual overlap ambiguity views based on autocorrelation further inform users that FR incurs more visual overlaps between communities than the other force-directed approaches. Although FA2+LinLog has the least visual overlap between communities, the heatmap view for the autocorrelation-based metric clearly shows that several nodes are near the regions of other communities. This may affect user's perception of these communities, but the problem cannot be easily identified in the original graph layout. The node occlusion views show that all three layouts have little to no node occlusion and thus are not presented in Fig. 8. When examining edge crossings, FA2+LinLog has relatively less ambiguity overall when compared with FR and HU. This finding is supported by both the global metric value and the heatmap. However, the crossings can be serious in local regions, impeding a users understanding the connectivity between nodes. When considering the edge length ambiguity metric, it is not easy to conclude which layout has the most/least of this ambiguity by using only the heatmap views. The value of the global metric indicates that the three layouts have similar values for this ambiguity (FA2+LinLog: 0.026, FR: 0.024, HU: 0.02), whereas FA2+LinLog has a relatively high value. The heatmap views for node-edge occlusion show that FR has the least amount of node-edge occlusion in the drawing.

As each layout has its own advantages, AmbiguityVis informs users of the advantages and disadvantages of each drawing for a specific network. The choice of layout depends on the task of the user and which metrics need to be emphasized for that task.

6.2 Case Studies for Node/Edge Aggregation

Two case studies are presented to demonstrate the effectiveness of the ambiguity metrics and visualization feedback for edge bundling and metanode simplifications. The network considered by this case study is the same network used in the case study of Section 6.1.2.

6.2.1 Force-directed Edge Bundling

Force-directed edge bundling [25] is one of the most representative edge bundling methods [56]. Therefore, it is adopted for our case study to demonstrate the effectiveness of AmbiguityVis in showing ambiguities in edge bundling. An implementation of this edge bundling algorithm is included in AmbiguityVis, and users can choose to detect the edge bundling ambiguity after loading a graph layout.

When considering the consistency of bundled edges, users can select two bundles by brushing a line across them as shown in Fig. 9(a). Each edge present in the selection is mapped to a point in a scatterplot (Fig. 9(b)) using MDS and Equation (10). In this case study, linked highlighting is used to inform users of the corresponding edges in each view. Fig. 9(a) shows the result when users interactively select two bundles that have visual similarities. The MDS projection (Fig. 9(b)) indicates that the bundled edges in brown are more suitable for bundling than those in blue, as the brown points are clustered more tightly in the scatterplot. When viewing the original node-link diagram, shown in Fig. 9(c), it is clear that the brown edges are almost parallel to each other, and therefore are more suitable for bundling. Also, more crossings exist in the blue edges even though they have similar edge length and are close to each other. Users can select bundles of edges and include nearby edges. The MDS scatterplot can help users decide if edges should be added to or removed from the bundle. Fig. 9(e) shows that the bundled edges in brown in Fig. 9(d) are suitable for bundling, as they form a tight cluster in the scatterplot. However, none of their surrounding edges (marked in blue) can be added to this bundle, as none of the blue points (Fig. 9(e)) are close to the brown cluster. This finding is confirmed in the original node-link diagram (Fig. 9(f)). Thus, a scatterplot based on these measures and MDS can provide useful visual feedback.

The heatmap can be used to visually show the ambiguity in edge bundling. For example, Fig. 10(b) provides an overview of the ambiguities in the edge bundling and helps locate the most ambiguous parts through linked highlighting (the red rectangles). As we can see in Fig.

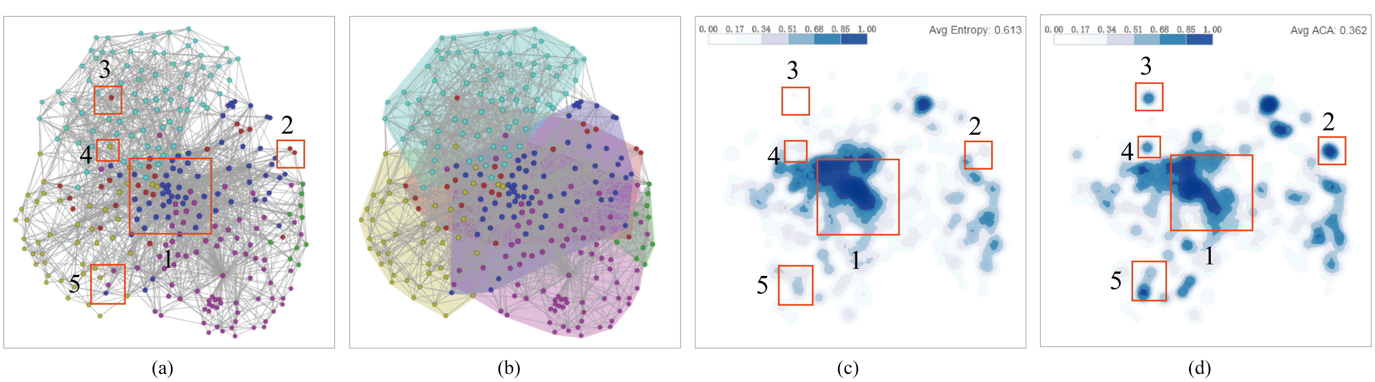

Fig. 7: Visualization feedback for community overlap in the layout of neural network data set. (a) Original graph layout in which the node color encodes the community type. (b) Convex hull. (c) Heatmap based on entropy. (d) Heatmap based on autocorrelation.

10(a), the three marked areas have more ambiguities due to crossings and relatively high edge curvature.

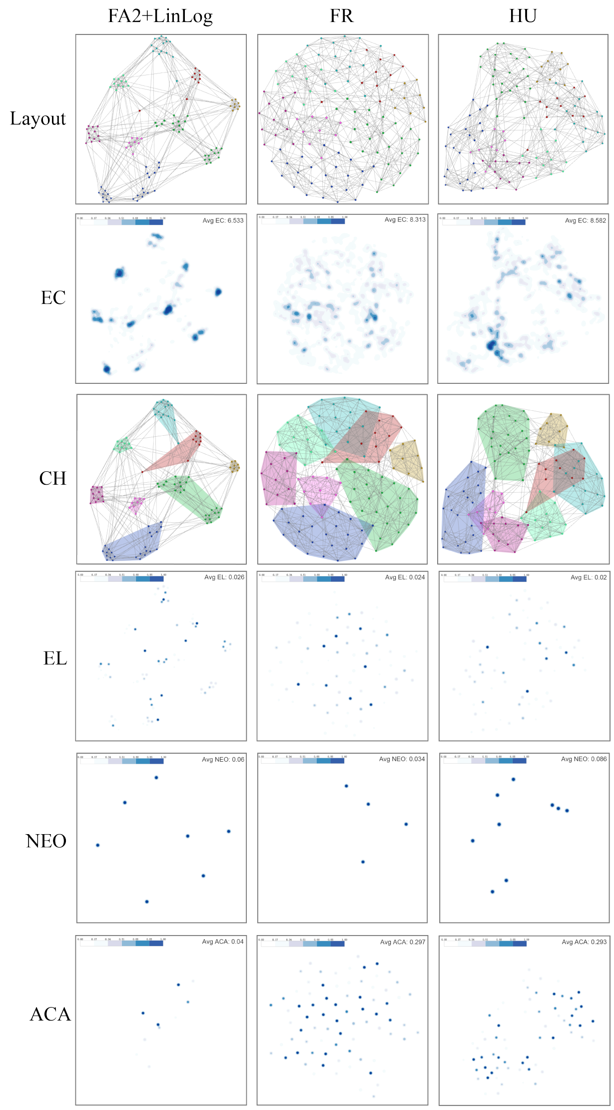

Fig. 8: Detailed comparison of three graph layouts for the same data [19]. The rows from top to bottom represent views of the original layout, edge crossing (EC), convex hull (CH), edge length ambiguity (EL), node-edge occlusion (NEO) and autocorrelation-based ambiguity (ACA) metrics.

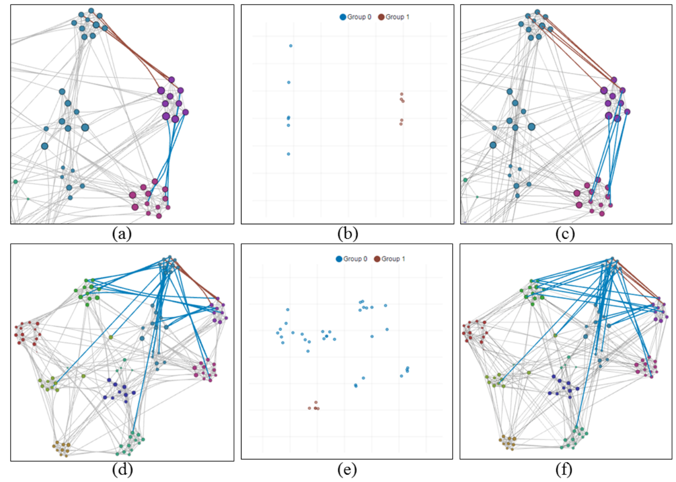

Fig. 9: Evaluation for the consistency of edges in the bundle. (a) Two bunches of bundled edges which look similar in terms of edge bundling and are selected by interaction. (b) Mapping the brown and blue edges in (a) onto scatterplot using MDS. (c) Original node-link diagram of (a). (d) A cluster of bundled edges (brown) and its surrounding edges (blue) are selected by interaction. (e) The brown and blue edges in (b) are mapped onto the scatterplot using MDS. (f) Original node-link diagram of (d).

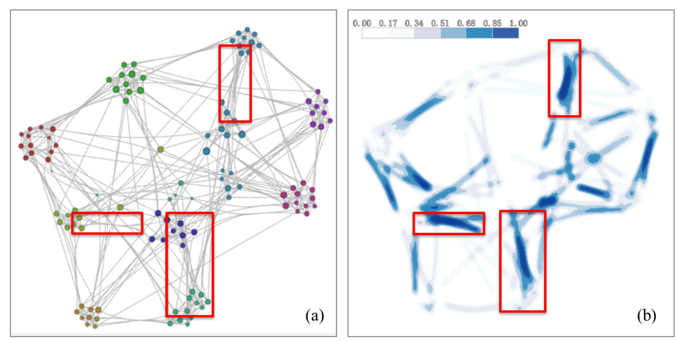

Fig. 10: Visualization of the ambiguity in edge bundling using heatmap. (a) Graph layout using force-directed edge bundling. (b) Heatmap showing the degree of ambiguity in bundled edges.

6.2.2 Node Aggregation Statistics

In the following case study, we interactively aggregate communities of the node-link diagram into circular metanodes. As is typical for most node aggregation methods, the size of the metanode is proportional to the number of nodes contained inside it, and the width of edge between two metanodes is proportional to the number of edges between the corresponding nodes. In Fig. 11(a), only the relative size of each community is encoded along with the number of the links between them. By using the statistics displayed about each community, users can see more information on these communities including the number of nodes, the number of inter and intra cluster edges, and the graph density. AmbiguityVis allows users to view one or multiple statistics for each metanode by interactive selection, as shown in Figs. 11(b)(d). In Fig. 11(b), all the four statistics are selected for display, but it is hard to compare the graph density of the metanodes. By selecting only the graph density, we can more easily compare the graph densities between metanodes (Fig. 11(d)). Based on these two views (Figs. 11(b)(d)), there are two interesting features in the data. First, metanode 1 has the most nodes (about 24) and the most intra-cluster edges (101 edges), but it is not very close to a complete graph, because its graph density is only about 0.36. Second, although metanodes 8 and 9 contain only a few nodes, they are actually complete graphs, since the graph densities of both metanodes are 1.0.

6.3 Expert Reviews

In order to further evaluate the technique presented in this paper, we conducted two expert interviews with domain experts: Researcher A has more than ten years of graph visualization research experience and is still very active in graph drawing field. Researcher B is a postdoctoral researcher interested in visual analytics and has published several top conference papers about graph visualization. In both interviews, the experts showed great interest in our technique.

Researcher A. After being briefly introduced to AmbiguityVis, he first asked several questions about the computation of the quantitative metrics for each ambiguity and the interactive capabilities of the prototype. He enjoyed using AmbiguityVis and, in particular, using the heatmap. This visualization technique was useful in informing users of the degree and position of each type of underlying ambiguity. Drawing on his research experience, he mentioned that coping with ambiguity is inevitable in graph drawing, and that it is important to understand the location and severity of these ambiguities in the layout. The expert also commented that visualizing the ambiguity in edge bundling and metanodes is important, since it is quite helpful in designing hierarchical graph visualizations. He had not seen much work done in this area before. Overall, the expert found that *"The proposed ambiguity metrics and the heatmap-based visual feedback would help me to understand how my layout algorithm behaves, and whether changing layout parameters would help to improve the drawing clarity. The prototype system based on the proposed techniques is also useful for comparing layouts from different algorithms side-by-side."* The expert provided important feedback about AmbiguityVis which we have already integrated into the prototype. For example, we now have an option to superimpose parts of the graph structure on top of the heatmap to better localize ambiguities. Also, the color map has been changed from a rainbow color map to a sequential color map. He suggested that when processing very large graph, the size of each view might need to be expanded, thus a better arrangement of each view is required because of limited screen space. The expert recommended some possible alternative methods, such as CH measure [6], which can be used to detect community structure ambiguities. These suggestions have been left for future work.

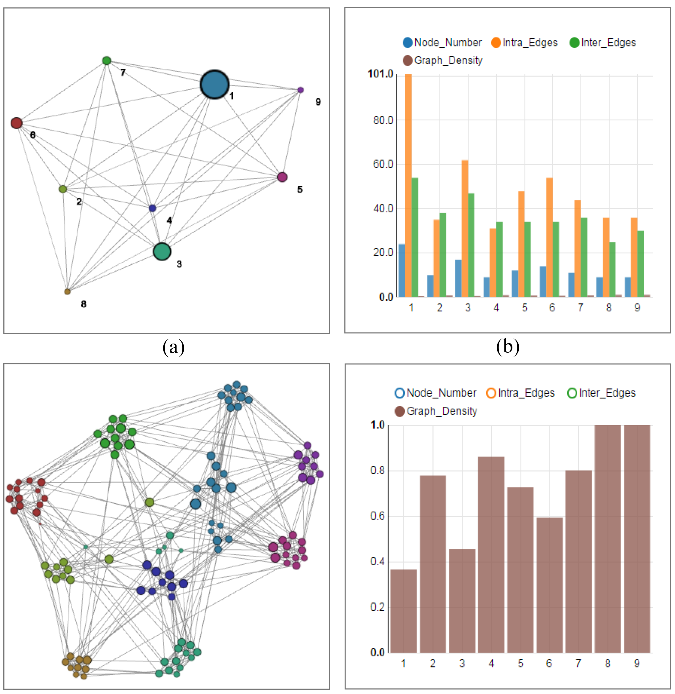

Fig. 11: Visual feedback for metanodes. (a) Metanode network structure generated by merging nodes in the same community based on modularity. (b) Bar chart showing the number of nodes, intra-cluster edges, inter-cluster edges of each metanode and graph density. The x-axis shows the identifier number of each community. (c) Node-link diagram showing the same network with different color encoding different communities. (d) Bar chart showing the graph density of each metanode.

Researcher B. We first briefly introduced the ambiguity detection and visualization methods and basic operations of the prototype system to the expert. Then we asked her to explore the ambiguity distribution and evaluate the ambiguity severity in the graph layouts without using AmbiguityVis. The data we used is the layouts of the neural network [53] generated by ForceAtlas2 [29], FR [14], and Hu's method (HU) [26]. After she finished exploring the ambiguities, we asked her to localize the ambiguity and evaluate the ambiguity level using AmbiguityVis. When finishing all these tasks, she said that AmbiguityVis not only quickly highlights the potential ambiguous regions that are consistent with her judgment, but also offers insight into problematic areas which otherwise may not be easily discovered. Overall, she appreciated the interaction and visual feedback. For example, she said that when she brushed over the curved edges to select them, the feedback from the scatterplot helped her judge how suitable the edges would be for bundling. Researcher B also mentioned that using the implemented prototype is not easy and recommended integrating popular layout algorithms into AmbiguityVis as direction for future work. This feature would enable users to load the network data directly and start exploring immediately without the need to compute a layout beforehand.

7 Discussion and Conclusion

In this paper, we present a technique that is capable of visualizing three classes of ambiguities in graph layouts: ambiguous spatial relationships between nodes and edges, visual overlap between different communities, and ambiguities that exist in node/edge aggregation methods. New metrics are proposed to quantify these ambiguities such as abnormal edge lengths, visual overlap between community structures, and node/edge aggregation. We present a visualization method that can provide both an overview and details of these ambiguities. These novel metrics, along with established metrics, are integrated into AmbiguityVis. Case studies and expert reviews demonstrate that the novel ambiguity metrics and the heatmap-based visualization can help developers and analysts quickly and accurately localize potential areas of visual ambiguity.

AmbiguityVis has some limitations. First, although AmbiguityVis will highlight areas of a graph drawing with high scores in a particular metric, the ambiguity it detects may not be important for all tasks. Therefore, more task-driven ambiguity visualization techniques would be interesting to pursue. Second, the effectiveness of our visualization is demonstrated through case studies and expert reviews. Human-centered evaluation is required to further support the effectiveness of the metrics in quantifying the level of ambiguity perceived by users in graph drawings. Finally, AmbiguityVis takes a graph and its drawing as input. Combining popular graph layout algorithms with the technique proposed in this paper would more conveniently support the comparison of these algorithms.

In the future, we plan to conduct human-centered experiments to provide further support for the proposed metrics. Also, we will further improve the scalability of AmbiguityVis, in terms of both its visual and computational scalability. This approach can be extended to show ambiguities in other visualization techniques such as parallel coordinates. In order for the graph drawing and visualization community to benefit from our software, we plan to release an application built off the *Gephi* libraries. This software will help developers in choosing appropriate layouts for particular tasks and designing new graph drawing approaches.

Acknowledgments

The authors wish to thank Yifan Hu, Panpan Xu, Youliang Yan, and Guowei Huang for offering many constructive suggestions. The authors also thank the anonymous reviewers for their valuable comments. This research was supported in part by HK RGC GRF 618313, a grant from Huawei Co. Ltd, and the National Natural Science Foundation of China (No. 61303133).

References

[1] J. Abello, F. Van Ham, and N. Krishnan. ASK-GraphView: A large scale graph visualization system. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):669–676, 2006.

[2] D. Archambault, T. Munzner, and D. Auber. GrouseFlocks: Steerable exploration of graph hierarchy space. *Visualization and Computer Graphics, IEEE Transactions on*, 14(4):900–913, 2008.

[3] D. Archambault, H. C. Purchase, and B. Pinaud. The readability of path-preserving clusterings of graphs. *Computer Graphics Forum (Proc. of EuroVis '10)*, 29(3):1173–1182, 2010.

[4] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall, 1998.

[5] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *Graph Drawing*, pages 42–53. Springer, 2007.

[6] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[7] T. A. Davis and Y. Hu. The university of florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1, 2011.

[8] A. T. Duchowski, M. M. Price, M. Meyer, and P. Orero. Aggregate gaze visualization with real-time heatmaps. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 13–20. ACM, 2012.

[9] C. Dunne, S. Ross, B. Shneiderman, and M. Martino. Readability metric feedback for aiding node-link visualization designers. *IBM Journal of Research and Development*, 59(2/3):14–1, 2015.

[10] C. Dunne and B. Shneiderman. Improving graph drawing readability by incorporating readability metrics: A software tool for network analysts. *University of Maryland, HCIL Tech Report HCIL-2009-13*, 2009.

[11] C. Dunne and B. Shneiderman. Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3247–3256. ACM, 2013.

[12] G. Ellis and A. Dix. The plot, the clutter, the sampling and its lens: occlusion measures for automatic clutter reduction. In *Proceedings of the working conference on Advanced visual interfaces*, pages 266–269. ACM, 2006.

[13] S. I. Fabrikant and D. R. Montello. The effect of instructions on distance and similarity judgements in information spatializations. *International Journal of Geographical Information Science*, 22(4):463–478, 2008.

[14] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[15] E. Gansner, Y. Hu, S. Kobourov, and C. Volinsky. Putting recommendations on the map: Visualizing clusters and relations. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 345–348, New York, NY, USA, 2009. ACM.

[16] E. R. Gansner, Y. Hu, S. North, and C. Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 187–194. IEEE, 2011.

[17] R. C. Geary. The contiguity ratio and statistical mapping. *The incorporated statistician*, 5(3):115–146, 1954.

[18] H. Gibson, J. Faith, and P. Vickers. A survey of two-dimensional graph layout techniques for information visualization. *Information visualization*, 12(3-4):324–357, 2013.

[19] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[20] S. Hachul and M. Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In *Proceedings of the 12th International Conference on Graph Drawing*, GD'04, pages 285–295, Berlin, Heidelberg, 2004. Springer-Verlag.

[21] D. Harel and Y. Koren. A fast multi-scale method for drawing large graphs. In *Graph drawing*, pages 183–196. Springer, 2001.

[22] I. Herman, G. Melancon, and M. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[23] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006.

[24] D. Holten, P. Isenberg, J. J. Van Wijk, and J.-D. Fekete. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 195–202. IEEE, 2011.

[25] D. Holten and J. J. van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009.

[26] Y. Hu. Efficient, high-quality force-directed graph drawing. *Mathematica Journal*, 10(1):37–71, 2005.

[27] W. Huang. Using eye tracking to investigate graph layout effects. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on*, pages 97–100. IEEE, 2007.

[28] W. Huang, P. Eades, and S.-H. Hong. Beyond time and error: a cognitive approach to the evaluation of graph drawings. In *Proceedings of the 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization*, page 3. ACM, 2008.

[29] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLoS ONE*, 9(6):e98679, 06 2014.

[30] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15, 1989.

[31] M. Kaufmann and D. Wagner. *Drawing graphs: methods and models*, volume 2025. Springer, 2003.

[32] A. Lancichinetti and S. Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[33] J. M. Lewis, M. Ackerman, and V. De Sa. Human cluster evaluation and formal quality measures: A comparative study. In *Proc. 34th Conf. of the Cognitive Science Society (CogSci)*, pages 1870–1875, 2012.

[34] S.-J. Luo, C.-L. Liu, B.-Y. Chen, and K.-L. Ma. Ambiguity-free edge-bundling for interactive graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 18(5):810–821, May 2012.

[35] K. Marriott, H. Purchase, M. Wybrow, and C. Goncu. Memorability of visual features in network diagrams. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2477–2485, 2012.

[36] C. McGrath, J. Blythe, and D. Krackhardt. Seeing groups in graph layouts. *Connections*, 19(2):22–29, 1996.

[37] M. E. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[38] M. Pohl, M. Schmitt, and S. Diehl. Comparing the readability of graph layouts using eyetracking and task-oriented analysis. In *Computational Aesthetics*, pages 49–56, 2009.

[39] H. C. Purchase. Metrics for graph drawing aesthetics. *Journal of Visual Languages & Computing*, 13(5):501–516, 2002.

[40] H. C. Purchase, D. Carrington, and J.-A. Allder. Empirical evaluation of aesthetics-based graph layout. *Empirical Software Engineering*, 7(3):233–255, 2002.

[41] H. C. Purchase, C. Pilcher, and B. Plimmer. Graph drawing aestheticscreated by users, not algorithms. *Visualization and Computer Graphics, IEEE Transactions on*, 18(1):81–92, 2012.

[42] A. Quigley and P. Eades. FADE: Graph drawing, clustering, and visual abstraction. In *Graph Drawing*, pages 197–210. Springer, 2001.

[43] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[44] B. Saket, P. Simonetto, and S. G. Kobourov. Group-level graph visualization taxonomy. In *EuroVis 2014 — Short Papers*, pages 85–89. Eurographics Association, June 2014.

[45] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28(3):831–838, 2009.

[46] R. Tamassia. *Handbook of graph drawing and visualization*. CRC press, 2013.

[47] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 59–66. IEEE, 2009.

[48] W. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 3(1):743–767, 1963.

[49] F. van Ham and B. E. Rogowitz. Perceptual organization in user-generated graph layouts. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1333–1339, 2008.

[50] R. van Liere and W. de Leeuw. Graphsplatting: visualizing graphs as continuous fields. *Visualization and Computer Graphics, IEEE Transactions on*, 9(2):206–212, April 2003.

[51] C. Vehlow, F. Beck, and D. Weiskopf. The State of the Art in Visualizing Group Structures in Graphs. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARs*, pages 21–40. The Eurographics Association, 2015.

[52] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J.-D. Fekete, and D. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011.

[53] D. J. Watts and S. H. Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

[54] K. Xu, A. Cunningham, S.-H. Hong, and B. H. Thomas. Graphscape: integrated multivariate network visualization. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on*, pages 33–40. IEEE, 2007.

[55] K. Xu, C. Rooney, P. Passmore, D.-H. Ham, and P. H. Nguyen. A user study on curved edges in graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2449–2456, 2012.

[56] H. Zhou, P. Xu, X. Yuan, and H. Qu. Edge bundling in information visualization. *Tsinghua Science and Technology*, 18(2):145–156, 2013.