Sticky Links: Encoding Quantitative Data of Graph Edges

Min Lu, Xiangfang Zeng, Joel Lanir, Xiaoqin Sun, Guozheng Li, Daniel Cohen-Or, and Hui Huang

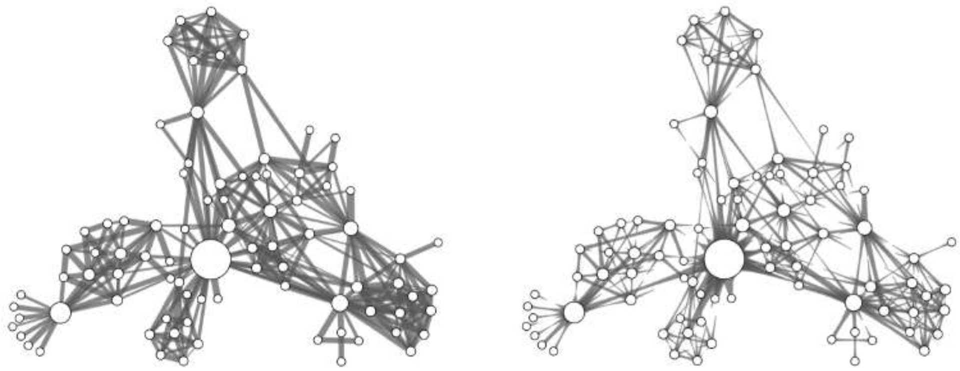

Fig. 1. Thickness vs. Stickiness for Encoding Quantitative Information of Graph Edges: compared to conventional visual encoding using thickness (left), sticky links (right) support a more readable encoding of quantitative information and connectivity, with significantly less visual clutter.

Abstract—Visually encoding quantitative information associated with graph links is an important problem in graph visualization. A conventional approach is to vary the *thickness* of lines to encode the strength of connections in node-link diagrams. In this paper, we present *Sticky Links*, a novel visual encoding method that draws graph links with *stickiness*. Taking the metaphor of links with glues, sticky links represent connection strength using spiky shapes, ranging from two broken spikes for weak connections to connected lines for strong connections. We conducted a controlled user study to compare the efficiency and aesthetic appeal of stickiness with conventional thickness encoding. Our results show that stickiness enables more effective and expressive quantitative encoding while maintaining the perception of node connectivity. Participants also found sticky links to be more aesthetic and less visually cluttering than conventional thickness encoding. Overall, our findings suggest that sticky links offer a promising alternative to conventional methods for encoding quantitative information in graphs.

Index Terms—Graph Visualization, Edge Drawing, Quantitative Encoding

1 Introduction

Graph drawing is at the very core of information visualization, and a large variety of graph visualization methods have been proposed to better reveal and understand the patterns of connectivity in network data [26, 62]. Node-link diagram [2] is the most widely-used graph visualization method, where the connections among entities are simply drawn as lines (called *links*). In many cases, links include a quantitative attribute that conveys the strength of the connections. For example, links may wish to convey the number of collaborations between entities in a social network or the volume of traffic between cities in a transportation network.

The conventional way to encode quantitative information of links in a node-link diagram is through the thickness of the lines. As shown in Figure 1(left), thicker lines are drawn to convey larger values, and thinner are drawn to convey smaller ones. However, since the width of a line is often limited in its span, the expressiveness of thickness is limited too. Also, this intuitive visual encoding of graph edges does not scale well. Visual clutter becomes a severe problem as the number of nodes and links increase, particularly in small regions with many links. There are many approaches to tackle this overdrawing problem, such as graph layout algorithms that modify the node positions to reduce the amount of edge crossing [3, 43], bundling of the edges [30, 33], or simplifying the graph by drawing the meta structure of the graph [20]. These works all perform some sort of manipulation on the graph's structure or layout, but do not change the visual encoding of a given graph and in particular its links.

This work introduces a novel method for graph visualization that enables the encoding of quantitative attributes of graph links. As depicted in the right side of Figure 1, we propose the use of *Sticky Links* instead of the traditional thickness-based approach in node-link diagrams. Sticky links are lines between nodes that are injected with *stickiness*, which encodes the quantitative strength of connections. The shape of sticky links is manipulated by the stickiness parameter, which ranges from a pair of separated strokes to a connected band, taking inspiration from the metaphor of sticky spikes. The key idea behind sticky links is to leverage the power of shapes by varying them spatially along with the length of the link rather than its width. With combined changing of length and shape, the difference between link weights can be expressed more precisely and easily.

The perception of the connection of nodes with sticky links is based on the *Gestalt Principle of Continuity* [66], which leverages the human perceptual tendency to continue a flow of visual elements, and thus perceive linked nodes from separate spikes, aligning with the notion of Partial Edge Drawing [4, 53]. Using sticky spikes rather than continuous links, the visual encoding of edges is relaxed from continuous

- *Min Lu, Xiangfang Zeng, Xiaoqin Sun and Hui Huang are with Shenzhen University. E-mail: {lumin.vis, xiangfangzeng15, ssharonqin, hhzhiyan}@gmail.com.*
- *Joal Lanir is with The University of Haifa. E-mail: ylanir@is.haifa.il.*
- *Guozheng Li is with Beijing Institute of Technology. E-mail: guozheng.li@bit.edu.cn.*
- *Daniel Cohen-Or is with Tel Aviv Univeristy. E-mail: cohenor@gmail.com.*
- *Hui Huang is the corresponding author of this paper.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

lines, which effectively reduces the visual clutter, while preserving the perception of quantitative information as well as connectivity.

To demonstrate the usefulness of sticky links in graph drawing, we first show its effectiveness in reducing visual clutter by measuring the edge crossings and ink ratio [60]. Then, we perform a controlled user study showing that links with stickiness provide significantly better visual perception than those using thickness in attribute-related graph tasks. Also, sticky links are found to be as good as lines at conveying graph connectivity, but with much less visual clutter. Furthermore, participants preferred sticky links, stating it is more aesthetic and visually clear.

2 Related Work

We first discuss works related to node-link diagrams. We then review the general topic of graph readability, followed by a review of the topic of partial edge drawing in graphs.

2.1 Node-link Diagrams

Over the years, many graph visualization methods were developed [28] including node-link diagrams, such as NodeXL [58], or matrix representations for graphs, like in Matrix Zoom [1], MatrixExplorer [25] or Nodetrix [26] (which combines both types of representations). Due to its intuitive representation of connections, node-link diagram is clearly the most common way to visualize graphs and has attracted a large graph drawing community exploring the geometric representation and the layouts of graphs for years.

Various layout methods have been proposed over the last decades [2]. Reingold and Tilford [50] were the first to present tree layout for graphs with intrinsic hierarchy. Along this line, numerous hierarchical layouts were proposed, such as the radial graph layout [35]. By modeling nodes and links as physical bodies tied with springs, Kamada and Kawai [37] first presented the Force-Directed layout algorithm in graph drawing. Before that, Eades [21] shared a similar algorithm that also used the concept of physical forces to arrange nodes. Thereafter various spring-based models were inspired, like in [15] or [22]. Some other works considered graph layout as an embedding problem [23]. For example, Kruiger et al. [38] proposed a modified t-SNE method to preserve node neighborhoods. Beyond the automatic determination of graph layout, some algorithms allow integrating local editing by users [70], or transferring exemplar-based local editing to other similar sub-structures in the graph [47].

Another research direction for node-link diagrams involves improving the visual representation of nodes and links. Numerous studies have focused on developing new edge drawings that can better encode information about edges, such as their directionality and weight. For example, Riche et al. [51] proposed link curvature as a design variable to show the edge direction. Holten and Wijk [31] studied six edge drawings hinting at edge direction, including color, hue, and shape, etc. Later, they [29] extended the study of direction identification to five edge representations, including animated, textured links, taper links, etc. In the study, they used a visual representation known as the taper link, which shares similarities to our sticky link. The taper link employs a triangle-like shape to indicate the direction of connections between nodes, whereas our sticky link indicates the strength of connections.

Unlike the layout-related works above, our work does not deal with the determination of the placement of graph nodes and links, but with the drawing of a given graph of a layout that is already determined. We propose a new visual encoding of the edges in the graph. Our proposed method can be applied to visualize any node-link diagram with any kind of layout.

2.2 Multivariate Network Visualization

When the networks incorporate rich edge attributes beyond topology, effectively visualizing these multivariate graphs becomes critical. We explored the influential studies and the current directions in this field.

Prior works have explored different visual variables for multivariate graph edges [36]. For example, Schmauder et al. [55] introduced timelines to link in dynamic weighted digraphs where the splitting of each link into segments like the time steps. Later, Didimo et al. [18] displayed timeline edges with color intensity, for example, red edges have the highest intensity than other colors. To encode the uncertainty associated with graph edges, Schwank et al. [57] studied four edge drawings (i.e., dashed, blurred, wavy and stripped lines) and found dashed lines are the most efficient encoding way to perceive the uncertainty values. Guo et al. [24] explored different visual variables for encoding uncertainty on edges including light, grain, and fuzziness saturation on two types of tasks, visual search and comparison. Their results demonstrated that lightness can modify the overall visibility of the mark as well as maintain robustness against changes in the secondary visual variable. Considering node-link diagrams visualization in Augmented Reality, Wolfgang et al. [13] carried out a user study to evaluate the appropriateness of various variants for edge encoding. Bludau et al. [7] proposed "Unfolding edge" as an explicit example to facilitate the visualization of multivariate edge attributes and extended interaction such as zoom-in and rotation to this novel edge design. Through unfolding edge, multiple source types, involvement of uncertainty like dash and temporal dimension can be displayed of a link. Sebastin et al. [56] developed two small embedded visualization approaches for edge multivariate attributes, that is Partially filled Bars and Bars of Vary Height.

To encode quantitative information for edges except for timeline and uncertainty, visual variables such as opacity, thickness, and color, are usually used. For example, Wattenberg et al. [65] presented an arrow-like edge design, which uses thickness to encode the link weight. Dong et al. [19] studied color-gradient encoding (from light to dark) for geospatial networks and found it has significantly improved the effectiveness and efficiency compared to the use of thickness. Von et al. [64] used arcs to encode data attributes in a more complex form. Rather than relying on static variables, some researchers have explored the use of animated textures to encode data information. For example, Buschmann et al. [14] investigated the use of animated textures on edges to convey information about changing data over time. Romat et al. [52] proposed animated edge textures, using dynamic particles flowing along the edge to encode the data attributes.

Unlike the aforementioned studies, our research emphasizes encoding single-edge attribute and weight to significantly reduce cluster in a novel way, not considering visualizing multiple attributes on one edge which increase the cluster to some extent.

2.3 Graph Readability

The readability of node-link diagrams is an important research problem, especially for overly dense or too large graphs. Graph readability is closely related to the aesthetics of the graph. The fundamental work by Purchase [48] studied and quantified five aesthetic criteria that impact human understanding of graphs. The number of edge crossings in a graph is the most agreed upon aesthetic criterion. Bennett et al. [5] provided a survey of the aesthetic heuristics used in graph visualization, including the design heuristics of node placement, overall layout, etc. Wong et al. [68] studied the use of Gestalt principles in the perceptual grouping and segregation of graphs. They showed, for example, that nodes with a common feature appear to be grouped based on the similarity principle.

Based on the above study of aesthetic criteria and design principles in graph readability, some researchers presented means to improve the readability of graphs. A direct approach is to optimize the graph layout towards one or more graph drawing aesthetic criteria. For example, Matuszewski et al. [43] proposed an edge crossing minimization layout algorithm. Baur et al. [3] presented a crossing minimization algorithm in a circular graph layout.

Another branch of research focuses on graph simplification of massive graphs, which otherwise, would be infeasible to draw in a single screen [41]. For example, van Ham and van Wijk [63] proposed to use hierarchical clustering on networks and inspect local clusters, while maintaining a global overview of the entire structure. Another means of graph simplification is to design a summarised visualization for graphs that only shows meta structures [46]. For example, with the knowledge of motifs in graphs [44], Dunne and Shneiderman [20] visually summarized frequently occurring motifs with glyphs, including fan, connector,

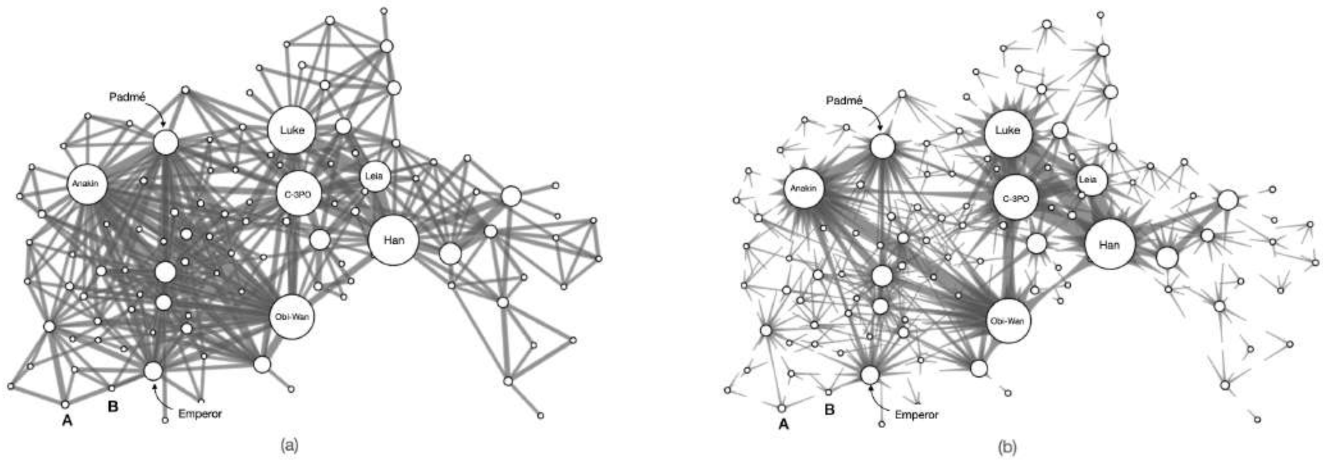

Fig. 2. Character Network of *Star War*: (a) thickness of lines encodes the link weight which is the number of common scenes between two characters; (b) sticky links draw the same graph with much less visual clutter region, by which it is more readable and more noticeable to compare the weight difference between links.

and cliques, etc. Some other works reduce the number of nodes drawn in graphs by filtering or sampling them. For example, the Skeletal images [27] highlight high-metric nodes and draw filtered trees with triangles. Furthermore, fundamental AI models can be used to extract graph content, such as influential users and sub-communities [69].

All the above works improve graph readability either by transforming the graph data or optimizing their layouts. Our work takes a different approach of solving the same issues in improving graph readability and aesthetics by introducing sticky links as a new visual design of links. Without the need to change the graph or its layout, sticky links significantly alleviate visual clutter via a smarter and lighter distribution of ink along links.

2.4 Partial Edge Drawing

The idea of conveying connections using partially drawn links goes in line with the research direction of partial edge drawing (i.e., PED) in the graph drawing community.

One of the earlier works in this direction was done by Becker et al. [4], separating the link into halves, and using the thickness of the halves to encode information. Several research works were performed along this research line. Rusu et al. [53] proposed asymmetric breaks in edges, in which one edge is fully connected while the other one has a jagged break at their crossing. Burch et al. [12] presented another asymmetric partial method, drawing the links starting at the start node and pointing to the end node. Later, Burch et al. [11] developed interactive partial links that allow for applying only to regions of interest. Collins et al. [17] drew connections in Parallel Tag Clouds as 'stubs' that connect to each endpoint text in the parallel coordinate and fade to transparency between the words, creating a visual effect of partial edge drawing. Streit et al. [59] applied the same idea of stubs and fading effects in the edge drawing of bicluster visualization.

Other design variations of partial edge drawing have been proposed later. For example, Schmauder et al. [55] used partial edges in dynamic graphs, by splitting each link as segments as time steps to encode the temporal information. Misue and Akasaka [45] proposed Morphing Edge Drawing (MED), in which links are morphed between partial and complete drawings. Other research focuses on the mathematical modeling of partial links. Bruckdorfer and Kaufmann [9] mathematically formalized PED and suggested several variants with symmetry and homogeneity in the breaks. Bruckdorfer et al. [8] studied the symmetric and σ-homogeneous where lengths to be a given fraction σ of the edge lengths PED (SHPED), and explore the bounds of SHPED of graphs. Finally, Hummel et al. [32] proposed efficient algorithms to solve the maximum-ink PEDs problem, to maximize the total length of PEDs while keeping the graph crossing-free.

Research efforts are also drawn into the empirical studies to learn the performance of partial edges compared with conventional links. Binucci et al. [6] performed a user study and found that homogeneity is more important than fewer crossings and more inks in partial edge drawing. Sathiyanarayanan and Priozzi [54] reported there is no significant difference in performance between complete edges and partial edges, but people prefer partial draw edges for visual aesthetics. Burch [10] studied how the link length and direction influence the judgment of the target node.

Although the concept of partial links has been studied for several years, research has mainly emphasized its benefits in enhancing connectivity perception. When representing the quantitative attribute of graph links, the visual encoding remains the thickness of the partial edges. In this work, we introduce sticky links, which unify fully connected lines and partial edge drawing into a single form using stickiness. The key difference is that by changing stickiness, the shape and length of links are utilized to encode link attribute values, which have not been explored in partial edge drawing before.

3 Stickiness vs. Thickness

Although network data often includes quantitative attributes for edges, very few visual forms of link encoding are used in node-link diagrams other than edge width (i.e., thickness), opacity, brightness, etc [61,65]. The reason behind this could be that the available visual channels are quite limited, considering that the position of the links (i.e., length, orientation, etc.) is largely determined by the position of the nodes. Also, the space available for visualizing links is also very limited as the line between two nodes. Since visual encodings such as opacity, brightness, or saturation, can be orthogonally layered over links, in this work, we mainly compare sticky links with the conventional link using thickness.

In this work, we introduce a novel way for quantitative encoding of node-link diagrams using *stickiness*. Using the metaphor of nodes bonded with glue, *stickiness* is defined as the connection strength between these nodes that can vary, resembling the varying degrees of adhesion associated with the nodes. The key idea of stickiness is to integrate both the *length* and *shape* of links into a new compound visual encoding that associates the strength of the links with a visual feeling of stickiness from one node to another. With the metaphor of glued sticks between nodes, we manage to redirect the shape of the links to encode quantitative information. By changing the amounts of glue, the shape and length are changed simultaneously, so that the graph link varies from small spikes with little glue to strongly connected lines with much glue, which allows a wider range of visual expressiveness in quantitative encoding.

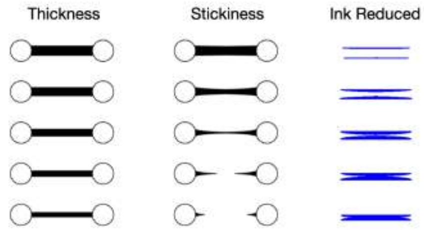

Fig. 3. Sampled pairs of links using thickness and stickiness, and their ink differences.

Figure 3 lists some pairs of links with thicknesses and stickiness ranging from large values to small ones. To enable the comparison between stickiness and thickness, given the same attribute value, sticky links are drawn with the same width as conventional lines at the end of the two nodes but shrunk in-between (design details are provided in Section 4.2). A clear advantage of using sticky links to visually encode the connections is that they effectively encode the strength and connectivity between nodes using a small amount of ink, where we refer to 'ink' as the pixels drawn to represent the link.

Figure 2 presents an example of the character networks in the movie series *Star War*, with 109 nodes and 398 links[1]. Each link has a weight that counts the number of scenes in which a pair of characters appear together. The size of each node represents the number of scenes the character appeared in. Figure 2(a) shows the classic node-link diagram where the thickness of the link encodes the weight. Figure 2(b) shows the sticky version of the same graph. Compared to the even-width distribution of ink in the thickness version, there are several advantages of unevenly distributing ink in *stickiness*, as elaborated below.

More Expressive Quantitative Encoding Quantitative information of links is more clearly and precisely visualized by sticky links, compared to the lines using thickness. In Figure 2(b), links among main characters stand out more and it is much easier to compare their weights in (b) than in (a). For example in (b), it is easy to tell that 'C-3PO' appears in more scenes with 'Obi-Wan' than with 'Anakin', 'Luke' has more with 'Han' than with 'Leia'. It is also very intuitive to identify the links with max-(min-) weights. For instance, 'Leia' has the most common scenes with 'Han'. 'Obi-Wan' has the most common scenes with 'Anakin'.

Sticky links leverage the power of both shape and length in visual discrimination, which makes it feasible to tell the subtle differences in weights. Taking the node 'Padmé' for example, it is easy to notice the differences of her links to others with stickiness in (b), while their differences are hardly noticeable in (a). Also, for the tiny links of 'Emperor', it is difficult to perceive the thickness difference between links, but it is much easier to perceive their stickiness difference.

Less Visual Clutter Sticky links significantly alleviate visual clutter compared to thickness. When links are drawn as two spikes with a small level of stickiness, the ink between links is largely reduced. Furthermore, given the same value, the middle part of the sticky link is more narrow than when using thickness, thus, less ink is consumed.

As can be seen in Figure 2(a), with an even distribution of 'ink' along the link, conventional lines can easily induce visual clutter when coming across others, especially when links travel in long distances. In Figure 2(b), sticky links draw graphs with much less ink, which makes the graph more readable compared to the thickness version. For example in the middle region of the graph, links between the main characters (i.e., the big nodes in the graph), such as 'Anakin', 'C-3PO', and 'Obi-Wan', which are hard to see in (a), can be easily perceived in (b). Also, the small nodes which are almost buried by overly crossed lines in (a), are feasible to recognize their interactions with others in (b). As can be seen, stickiness effectively reduces the visual clutter without damaging the delivery of quantitative information and the connectivity of links.

More Hierarchical Perception of Connectivity The use of sticky links in hinting connectivity is based on the Gestalt Continuity principle [66], which states that the human eye follows the line and prefers to see a continuous flow of visual elements. Varying stickiness can create a diverse illusion of connectivity. When the spikes are short and separated, such as links between small dots shown in Figure 2(b), connectivity can be softly implied. When stickiness is large, such as the connection between main characters, our brain strongly perceives the lines as one continuous entity. Compared to plain lines which flatten links on a plate, sticky links intrinsically provide a hierarchical perception of connectivity by stickiness. For example in Figure 2(b), it is very intuitive to learn the hierarchies of the interactions among characters, such as very sticky and thick links among the main characters, secondary sticky links of supporting characters, and small spikes of minor characters.

Another benefit of such link design is the localization of the connection, in that sticky links can effectively resolve the ambiguity in connectivity perception that could be caused by links of co-linear nodes. As can be seen on the left bottom of graphs in Figure 2, the two points noted A and B are col-linearly aligned with the node 'Emperor', and it is ambiguous to tell whether A and B are connected or not. This is clearer with sticky links in (b) that A has no connection with B, but with a connection with the 'Emperor' node.

4 STICKY LINKS

In this section, we present the design of sticky links. The open-source code and an online demonstration of sticky links can be found in the following link https://github.com/SZUVIZ/StickyLinks.

4.1 Design Rationale

Connections between nodes are visually represented by sticky links using the 'stickiness' metaphor. Proposed as a new way to visually encode links, the shape of the links is designed with two rationales.

DR1. Faithful Encoding: The design of sticky links as a visual encoding should follow the principle of *Faithful Encoding* [67], that the shape of sticky links should accurately reflect the underlying quantitative attribute. Spikes are drawn short and thin for small stickiness, and long and thick for large stickiness. It should be easy for users to interpret the magnitude of the underlying attributes by simply looking at the shape of the links.

DR2. Smooth Transition: The shape of sticky links should be consistently controlled by the *stickiness* (denoted with s). As stickiness increases, spikes should be smoothly and continuously transitioned from short and separate spikes to connected lines. Therefore sticky links should be modeled as a general visual representation of graph links that cover a wide range, from partial drawing edges [4] to fully connected lines.

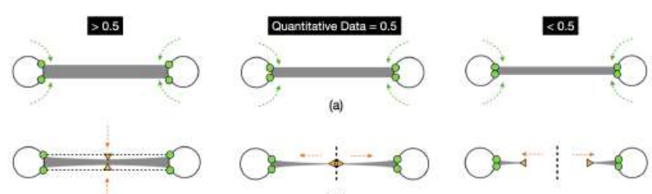

Fig. 4. Edges Drawn with Thickness and Stickiness: (a) to draw a line with thick, four control points (green dots) are linearly interpolated according to the attribute value. (b) taking the same four control points, sticky links change shapes and length (orange dots) by a piece-wise interpolation of the attribute value.

[1] https://www.kaggle.com/datasets/ruchi798/star-wars

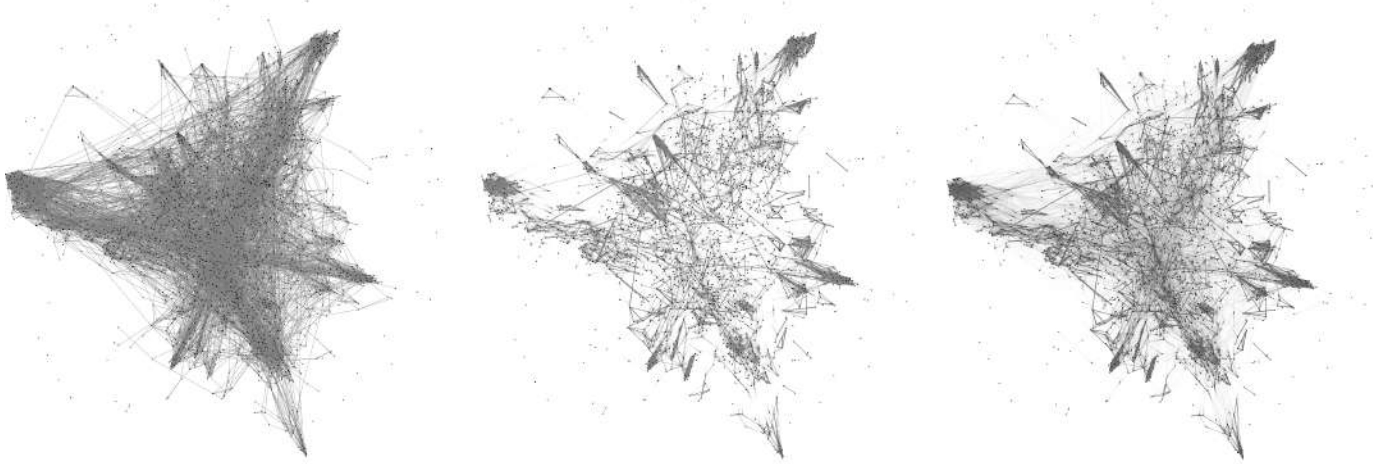

Fig. 5. Graph Visualizations on the Wiki Dataset using (left) conventional edges (middle) sticky links and (right) a design variation of sticky links in which dashed lines are drawn between the pair of spikes.

4.2 Design Details

To formulate the implementation, the magnitude of the underlying quantitative attribute q is normalized to the range of $[0,1]$. As shown in Figure 4(a), for the traditional thickness mapping, each link is defined by four control points (i.e., green dots in the figure) attached to the pair of nodes, moving along the arc of the node by a linear interpolation of q. As q decreases, the thickness of the link narrows down.

The design of sticky links is derived from conventional lines. To achieve *Faithful Encoding* (DR1), given q, sticky links take the same four control points as the thickness drawing, so that the bigger the q is, the thicker the sticky links are. But in sticky links, the shape of the link is also changed to enhance the quantitative encoding. Taking $q = 0.5$ as the middle point, two types of link drawing are applied (Figure 4(b)). When q is above 0.5, the link is curved by the other two control points (the orange triangles in the figure) that push the link to the middle of the band, with a force linear interpolated by q. As q decreases from 1 to 0.5, the link changes from a straight band to the sticky link as shown in the middle of Figure 4(b). When q is below 0.5, the connection breaks into a pair of spikes. In each spike, the orange control point shortens the spike by moving toward the corresponding node with a linear interpolation of q.

Figure 6 gives a close-up view to illustrate the drawing of sticky links. The control point p_i of node i is computed by Equation 1. Note that when $s > 0.5$, the two nodes share p_i and p_j.Taking 0.5 as the middle point, the link can be smoothly changed from a tightly-straight band to small spikes, as the stickiness decreases, which meets up the design rationale of *Smooth Transition* (DR2).

$$p_i = \begin{cases} m + \overrightarrow{m,m_b} * (s-0.5)/0.5 & 0.5 < s \leq 1.0 \\ m + \overrightarrow{m,c_i} * (0.5-s)/0.5 & 0 < s \leq 0.5 \end{cases} \quad (1)$$

where c_i is the point where the connection intersects with node i, m is the middle point of connection, m_b is the middle point of the straight band.

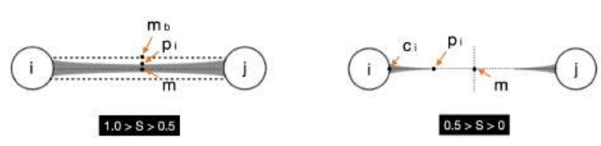

Fig. 6. Drawing of sticky links in two segments.

5 Examples of Use

In this section, we demonstrate three usage examples of sticky links in graphs by varying the quantitative information and their mappings to stickiness.

5.1 Real-world Network Visualization

In this example, we show a real-world network dataset *Wiki* [42] drawn with spiky links. Originating from Wikipedia, an online encyclopedia collaboratively created and edited by volunteers globally, the dataset represents a word co-occurrence network, constructed from the entirety of English Wikipedia pages, and comprises 2405 nodes and 12761 edges. In the preprocessing, we computed the Cosine similarity between node features and normalized them to fall within the range [0, 1], which then are taken as the weight of edges. Figure 5 shows the graph visualization drawn with thickness and stickiness. As can be seen, pairs of nodes with large similarities can be clearly seen in the graph with sticky links (middle), which are highly occluded in the conventional drawing (left). With the basic idea of sticky links, design variations can be easily extended, for example, shown in Figure 5(right), drawing thin dashed links between sticky links can make the connectivity more visually pronounced compared to basic sticky links.

5.2 Focus+Context Exploration

The mapping from the quantitative attributes to stickiness can be customized according to the goal of exploration. For instance, links that need to be emphasized or explored can be assigned with large stickiness, while others can be drawn as small spikes to just provide the context. In this way, sticky links can serve as a ”Focus+Context” tool for graph exploration [16].

Figure 7 depicts an example of the popular television series *Game of Thrones*. The network contains 109 characters and 353 links among those characters. The weight of every link is the number of interactions that the characters have had over the shows. Figure 7 show the graph drawings with stickiness on the left column and those with thickness on the right. By giving different mapping functions when mapping weight to stickiness, different kinds of links can be explored. For example, in the top of Figure 7, weight is positively linearly mapped to stickiness, therefore links with the strongest weight are highlighted. In the middle of Figure 7, the mapping is reversed (a negative linear mapping) to highlight the infrequent interactions. At the bottom of Figure 7, the focus is on link weights within a certain range. In comparison to sticky links, the focused links in the conventional graph are less noticeable, especially the one heavy clutter in the middle.

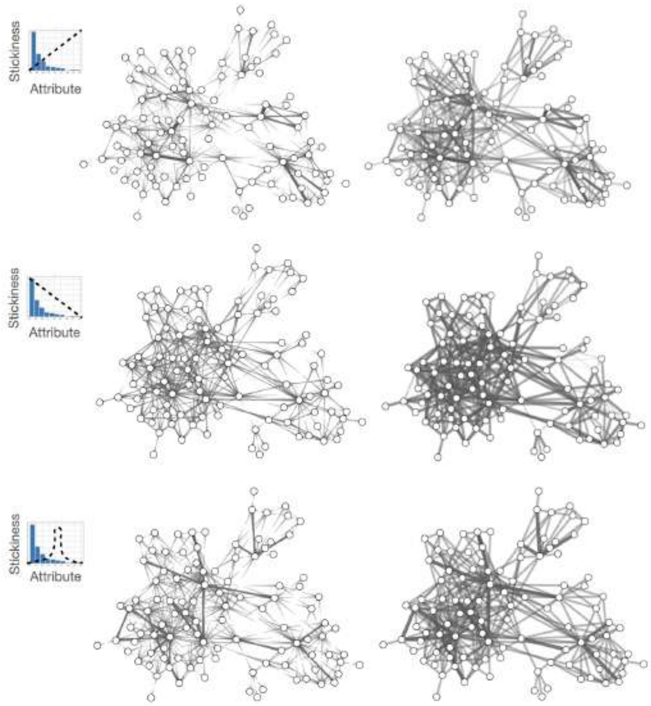

Fig. 7. Different Focus on Edge Links in *Game of Thrones* Network: Different connections can be visually emphasized via different mappings from attribute to stickiness, e.g., positive linear mapping (top), negative linear mapping (middle), and a V-shape mapping (bottom).

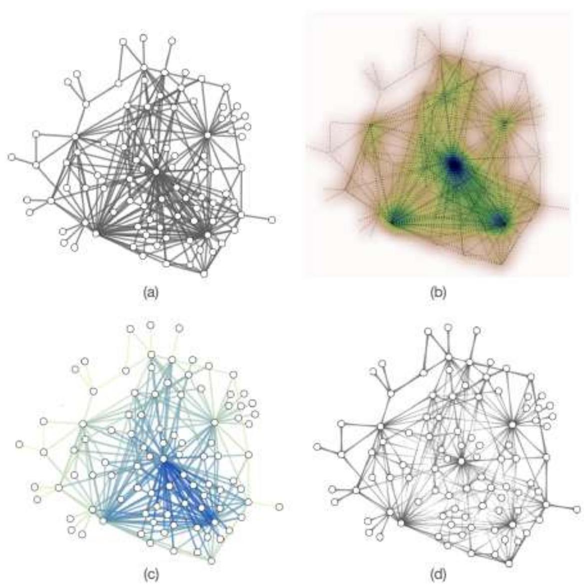

Fig. 8. Density-adapted Stickiness Example: (a) conventional node-link diagram; (b) estimated density field of links; (c) stickiness calculated as the reverse of the average density of all the points along the link; (d) density-adapted sticky links.

5.3 Density-adapted Stickiness

With sticky links, not only the quantitative attribute that links originally have can be encoded, but also the quantitative features of links that are derived from the graph. In the following example, we showcase the definition of a density-related link weight and use it as stickiness to make the link explicitly connected in spare regions, while links are drawn with small stickiness in dense regions, to alleviate visual clutter.

The calculation of density-adapted stickiness is shown in Figure 8. Links are first discretized into a sequence of points along the line, from which kernel density distribution is estimated. Regions with lots of line crossings are estimated high dense, and regions with fewer crossings are low dense. Then for each link, we compute its link weight q as the reverse of the average density of all the points along the link (Figure 8(c)). That is, links are drawn as little spikes with small stickiness if the connections come across in high-dense regions, and connected lines with large stickiness in low-dense regions. Compared to the traditional graph visualization (Figure 8(a)), sticky links with adapted stickiness provide a better trade-off between connectivity and clutter (Figure 8(d)).

6 Quantification of Decluttering

To demonstrate the effectiveness of sticky links, this section quantifies their ability of reducing visual clutter in a graph. In the following section, we report on a controlled user study conducted to assess the impact of sticky links on both quantitative perception and connectivity perception in node-link diagrams.

We use two metrics to measure the clutter in graphs. (1) *Number of Edge Crossings (NEC)*: the readability of graphs closely relates to the aesthetics of graph drawing, among which minimizing edge crossing is considered the top aesthetic criterion [49]. Thus, we chose the *number of edge crossings* as an important metric to assess the decluttering performance. In practice, an edge crossing is defined as a point where two edges intersect. (2) *Consumed Ink Ratio (CIR)*: we adopt the concept of *data ink* by Tufte [60] and compute the ratio of ink consumed to draw the graph. In the context of node-link diagrams, the consumed ink ratio is technically calculated as the proportion of non-white pixels which represent data in a raster image with a fixed size of $1200 * 1000$.

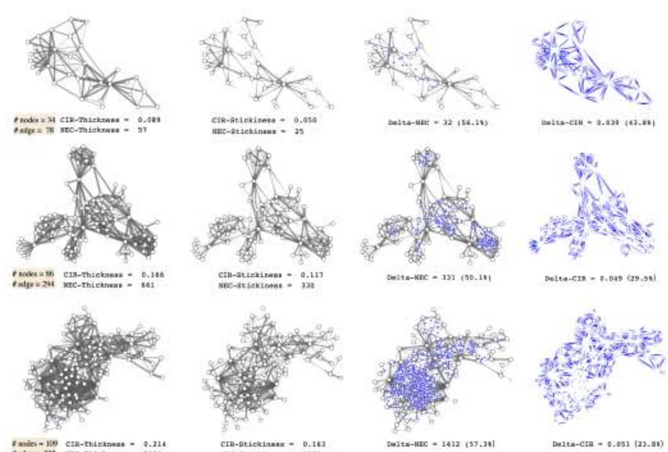

Fig. 9. Clutter Reduced by Sticky Links: compared to links using thickness, sticky links largely reduce the number of edge crossings (the third column) and ink ratio (the forth column).

To fairly compare between thickness and stickiness, given the same attribute value, sticky links are drawn with the same width at the base of the nodes as the thickness lines are, i.e., the four green points in Figure 4 are the same between the two edge drawings of thickness and stickiness. Figure 9 displays the reduction in edge crossings and ink ratio of sticky links (compared to thickness) for three small to large graph datasets (i.e., Zachary's karate club network, modified Les Misérables Character Network, and Character Network of Star War). The third column shows that more than half of the edge crossings in the graph are untangled in the sticky version. The forth column shows the distribution of ink reduced by sticky links, where blue pixels are the non-white pixels that appear only in conventional links but not in sticky links. As can be seen, sticky links significantly reduce the use of ink between nodes, effectively decluttering the diagram.

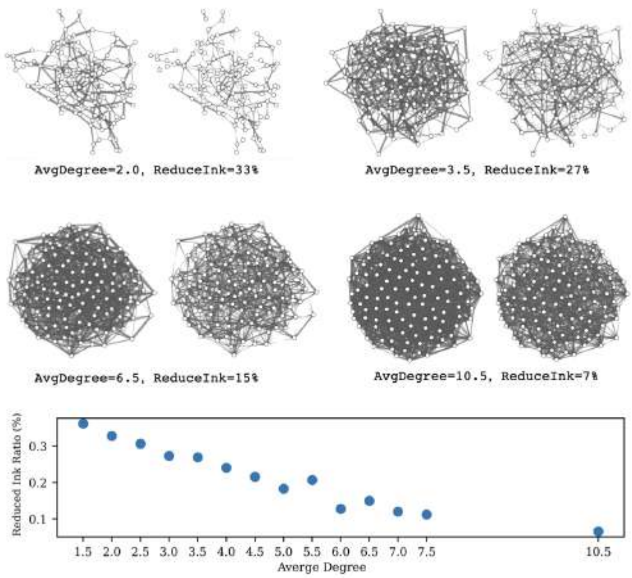

Fig. 10. Scalability of Sticky Links in Decluttering: sticky links alleviate the visual cluttering for graphs at low and medium average degree, by reducing the ink 10% or more. When a graph is highly dense (e.g., 10.5), it becomes also infeasible to read the graph with sticky links.

The performance of sticky links in decluttering is further examined by testing graphs with various connectivity. Specifically, a series of graphs with 120 nodes are synthesized by varying *Average Degree* (i.e., the ratio between the number of edges and the number of vertices) from 1.5 to 10.5. In each graph, links are randomly added or removed to the graph to achieve a certain average degree, and the assigned weights of links are randomly generated from 0 to 1. As depicted in Figure 10, a graph with an average degree of 2 can achieve a reduced ink ratio of up to 35%. The reduced ink ratio decreases close to linearly as the average degree increases. When the graph has low to medium connectivity, such as 2.0 to 4.5, the reduced ink ratio remains at 20% or higher. As the graph becomes denser, for instance, at 6.5, sticky links declutter the graph in a manner that allows people to perceive some connections that are likely occluded in traditional drawings. When the graph is densely connected, such as at 10.5, graphs drawing of both thickness and stickiness result in a tangled and messy ‘hairball’ appearance.

7 EVALUATION

We performed a controlled within-subject user study to compare the performance of *stickiness* and *thickness* in link representation. Following graph tasks summarized by Lee et al. [40], we focus on the evaluation of attribute-based tasks and topology-based tasks. Specifically, two basic tasks related to quantitative encoding and connectivity perception are chosen and designed: (1) *Task I:* given two links, to compare and find which one has a bigger attribute. (2) *Task II:* to identify whether two given nodes are connected or not. We had two hypotheses for the study: (1) *Hypothesis I*: Stickiness performs better in the attribute-based task (Task 1) than thickness. (2) *Hypothesis II*: Stickiness performs better in topology-based task (Task 2) than thickness.

7.1 Experiment Set-up

Our main study variable was type of edge drawing, i.e., thickness vs. stickiness. We tested the two tasks (T1 and T2) in three levels of *average degree* within two *graph scales*. For each graph scale, edges are randomly removed or added to the graph, to get three levels of densities (i.e., 2, 2.5, 3) [2]. The number of vertices (n) and edges (m) are reported in Table 1. For each testing graph, each link is assigned a randomly generated quantitative value in the range from 0 to 1.

[2]https://graph-tool.skewed.de/

Table 1. Six Graphs to Be Tested

Graph	**Vertices** (n)	**Edges** (m)	**Average Degree** (m/n)
Graph 1-1	60	120	2
Graph 1-2	60	150	2.5
Graph 1-3	60	180	3
Graph 2-1	120	240	2
Graph 2-2	120	300	2.5
Graph 2-3	120	360	3

ForceAltas2 algorithm was used to lay out all the graphs [34]. The diagram is rendered in black-and-white, to avoid the interference of color. As explained in Section 3, other visual encodings of links, such as opacity, can be orthogonally added to links, therefore we kept them constant (i.e., opacity = 0.5) for all links in all tests. Figure 11 shows the testing samples of three average degree levels (2, 2.5, 3) of graphs with 60 nodes using the two different edge drawing styles.

In both task 1 and task 2, for each condition of *graph scale* and *density level*, we generated 10 trials for each *edge drawing type*. Therefore, for each task, each participant was presented with 120 tests (2 graph scales * 3 clutter levels * 2 diagram types * 10 trials). The order of all tests was randomized, and for each test, we randomly rotated the graph to alleviate the learning effect.

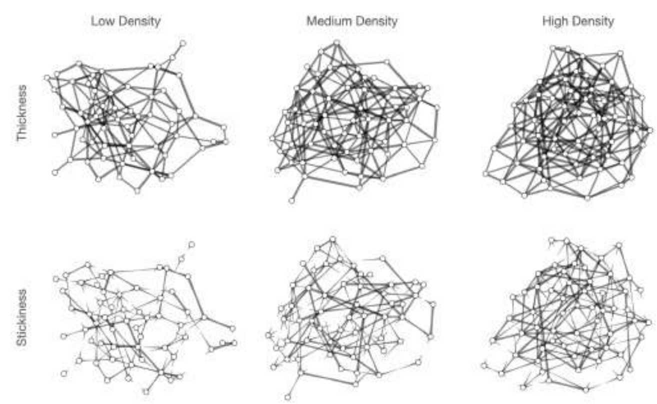

Fig. 11. Three average degree levels of graphs with 60 nodes, rendering in edge style of thickness and stickiness respectively.

The evaluation interface was implemented as a web-based application, that can be opened in a web browser. For task 1, two links are randomly selected from the graph. Links are highlighted by filling up their corresponding nodes - the pair of nodes of one link is colored in red and the other pair is colored in green. For task 2, two nodes are randomly selected from the graph and are highlighted by filling them in red. We balanced the selection of the two nodes to ensure that 50% of tests are of nodes that are connected and 50% of nodes that are not connected. For each test, the interface shows the node-link diagram at the same size. Below the diagram, a label asked the question. For task 1, it asked ’which link is with a larger attribute, the green pair or the red pair?’, and two buttons were presented titled ’green’ and ’red’. For task 2, it asked ’are the two highlighted nodes connected?’ and two buttons were presented ‘yes’ and ‘no’. The time cost from the moment the graph was presented till the time of the answer selection and the accuracy were measured for each test.

7.2 Procedure

We recruited 31 participants from a local university, consisting of 7 females and 24 males. All participants were science or engineering students. Out of the participants, 26 reported that they are familiar with node-link diagrams, with eight of them reporting that they have used node-link diagram before, and two reporting that they excel at node-link diagram. Five of the participants reported no prior knowledge of node-link diagrams.

The experiment was divided into two sessions, one for task 1 and the other for task 2, using the same group of participants. Participants were invited to sit in a quiet room in front of a 14” screen one participant at a time. At the beginning of the experiment, the experiment administrator gave a brief introduction of the experiment and tasks and also the use of the testing interface. Then the participant performed several practice tests with the administrator. Participants were notified that the time and accuracy of their tests will be recorded. They were encouraged to perform as accurately (most importantly) and quickly as possible. At the end of the experiment, participants were invited to fill in a questionnaire comparing the two methods and answer several open questions. On average, each experimental session took around 20 minutes. Each participant was compensated an equivalent of 5 USD for their participation in the study.

7.3 Analysis

For time cost, we first removed outliers of over 4 times standard deviation as they may indicate the participant has stopped during a trial and may thus skew the results. The distribution of each study variable was visually inspected in its Q-Q plot and was approximated normally distributed. We then conducted a 3-way repeated measures analysis of variance (ANOVA) on both accuracy and time. Edge type (sticky links, thickness), graph scale (60, 120) and average degree (2.0, 2.5, 3.0) were the independent variables. We reported statistical significance, effect size, and partial eta-squared η^2 [39]. Partial eta-squared quantifies the extent of an effect, which is independent of sample size. A value of 0.01 indicates a small effect, 0.06 suggests a medium effect, and 0.14 signifies a large effect.

7.4 Results

We first describe the accuracy results followed by the analysis of the time, as well as finally their preference results.

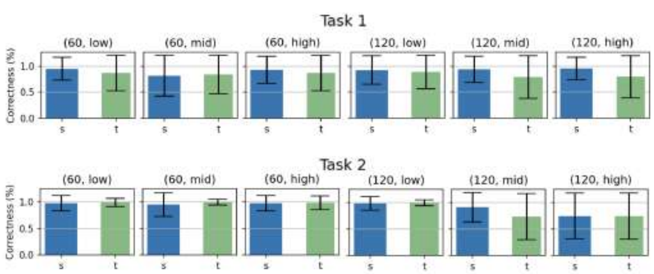

Fig. 12. Accuracy rate for the two tasks: sticky links achieve higher accuracy than conventional links in most conditions.

Accuracy Overall, in task 1 of comparing the link attribute value, the accuracy rate of stickiness was found to be higher than that of thickness. Stickiness had an average accuracy rate of 91.5% (SD=0.03), while for thickness it showed an average accuracy rate of 84.0% (SD=0.06). These differences were found to be significant, F(1,30)=35.9, $p < 0.001$, η^2=.54. In task 2 of identifying connectivity, stickiness also outperformed thickness, with a slightly higher accuracy rate (avg=92.5%, SD=0.08), while the average accuracy rate of thickness was 90.9% (SD=0.02). While this difference is small, it was also found to be significant, F(1,30)=8.45, $p = 0.007$, η^2=.22. Figure 12 gives the accuracy rates of stickiness and thickness over the six conditions. As shown here, graph drawings with stickiness generally support more accurate performance in the two tasks than those using thickness.

Time For task 1, in the task of comparing link attributes, the stickiness version took less time than thickness in all of the conditions. On average, stickiness took 4.73s (SD=5.63) to finish a test, while thickness took 6.48s (SD=7.01) per test. This difference was significant, F(1,30)=11.3, $p = 0.002$, η^2=.32. For task 2, On average, stickiness took a bit more time to complete than thickness, i.e, stickiness had an average of 5.00s (SD=5.99) while thickness the average was 3.91 (SD=4.05) per test, F(1,30)=72.72, $p < 0.001$, η^2=.71.

The distribution of time cost per graph scale and density level is shown in Figure 13. Points with extremely long time costs are withdrawn in the plot. Heuristically we set the threshold at 15s in both task 1 and task 2. Therefore task 1 ends up with 186 points withdrawn, and 95% data preserved. In task 2, 116 points are withdrawn and 96.9% data points are preserved. We can see in Figure 13 that for task 1, the time cost increases as either the graph scale or average degree increases. But for task 2, time cost shows less relevance to the graph scale or average degree, except when the graph becomes dense enough (120 nodes with an average degree of 3.0). While edge-tracking accuracy is supposedly better in low-density graphs, edge-tracking speed is lower. This is a common speed-accuracy tradeoff. Also, this can be explained by that the identification of connectivity is only limited to the observation of the highlighted points, instead of the global context.

Preference At the end of the user study, participants were asked to answer a questionnaire comparing sticky links with the thickness-based node-link diagram. Five questions were asked, including four specific rating questions (quantitative encoding, connectivity encoding, visual aesthetics, and visual clarity) and one overall rating. Questions were asked on a 5-point Likert scale. Figure 14 shows the rating result. Overall, sticky links were preferred over thickness-based links (average score 3.63). 21 out of the 31 participants strongly preferred or preferred sticky links. Several participants praise sticky links for their capability in drawing dense graphs. Six participants commented that they preferred the thickness-based node-link diagram. Four participants stayed neutral. Sticky links received high scores (average score 4.37) in visual clarity. 63.3% of participants strongly agreed and 20% thought that sticky link diagrams have better visual clarity and cause less visual clutter than traditional lines. Over 70.0% of the participants thought that the sticky link diagram is more visually aesthetic than the thickness node-link diagram. One participant specifically commented ‘the non-uniform stroke shape of spikes brings more visual contrast. A graph with lines looks flat and plain’. 67.7% of participants strongly agreed or agreed that quantitative information is better encoded by sticky links. One participant said ‘as the graph becomes dense, it is very difficult to read the link weights in graphs with lines, but it is easy with sticky links’. Another participant commented that nodes and their links stand out better in the sticky link version. For connectivity perception, participants preferred conventional lines more, with an average score 2.40. The reasons given for this was familiarity with the classic node-link diagram and the fact that participants thought it is more straightforward. Nine participants thought it is easier to identify connectivity with sticky links. One participant mentioned ‘...[conventional] lines are hard to see when it is too dense’. Another participant appreciated spikes’ advantage in addressing the ambiguity in connectivity perception when multiple points are aligned.

Summary In this study, we evaluated the performance of sticky links on two basic graph tasks, i.e., comparing attributes and identifying connectivity. The result of the study suggests that sticky links are more accurate than conventional lines in encoding quantitative information as well as depicting connectivity, especially when the graph is large and dense. For task of comparing the attribute values of links, sticky links took significantly less time than thickness. While participants were slightly slower on the connectivity task when using sticky links, more accurate outcomes were achieved, which could be explained by the capability of sticky links in reducing visual clutter. Overall, 70% participants preferred sticky links over a thickness-based node-link diagram, and thought sticky links are more visually appealing.

8 Discussion

In this section, we discuss some limitations of sticky links and introduce possible future works.

Encoding Connecting Direction *Sticky links* uses symmetric shapes to encode quantitative information, therefore the ability to use tapering to indicate direction is impacted as shown in previous related work [29]. Other visual designs related to shapes used on edge drawings such as curvature, glyphs, and arrowheads, have now also become

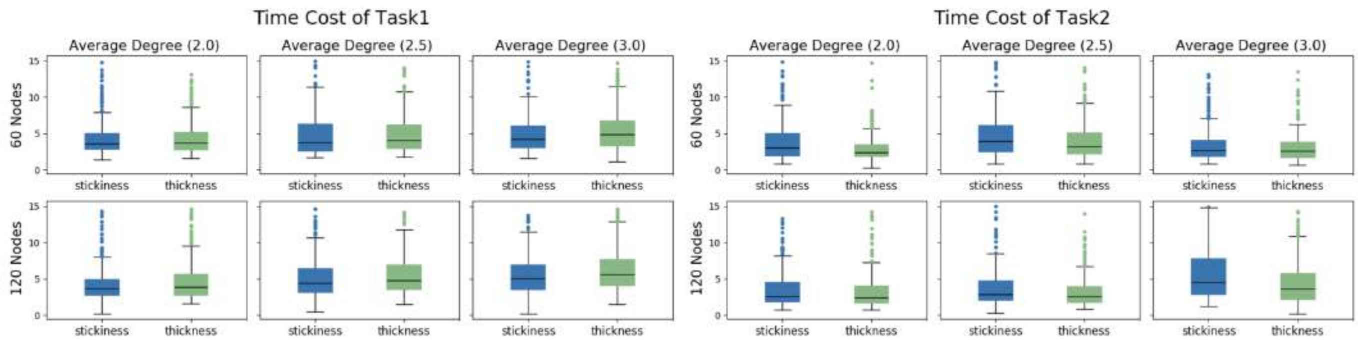

Fig. 13. Time cost for the two tasks: time costs for the two edge drawing types over 2 graph scales and 3 levels of average degrees.

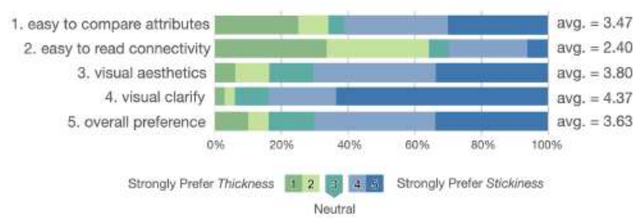

Fig. 14. Preference rating between sticky links and conventional links on a 5-point likert scale.

impossible. How to encode spatial mapping on sticky links to indicate direction could be an interesting research problem in the following step.

Connectivity Perception Effort Edges with small stickiness break into spikes effectively reduce edge crossings and address visual clutters. However, for some connectivity-oriented graph analytic tasks, thickness-based edge drawing may be more effective. In the user study, we found when a pair of nodes are at a long distance, it takes more time in connection perception. Sticky links leverage Gestalt continuity, which may be able to cover cases where two nodes are relatively spatially close and the graph is sparse. It is unclear if this also applies when there are a lot of longer links between locally connected components. Whether users accurately track long links using partial edge drawing would be an interesting problem to study in the future. Also, in this work we focused on the performance study of sticky links on implying connectivity of a pair of nodes. Other connectivity-related graph tasks, such as identifying shortest paths, should be further examined in the future.

Possible Design Variations In Figure 7, we show a possible design variation based on sticky links, where dashed lines are drawn to maintain the connectivity. In this work, we focus on advocating the basic idea of sticky links and the study of their performance. We believe sticky links open up a set of edge design possibilities for future research. For example in this work, we have only changed the visual encoding of the links, without modifying the graph structure or layout. An interesting research direction is to consider a hybrid method, where the complexity or the layout of the graph itself is considered in conjunction with the use of stickiness. This may allow perceiving even more information with less ink.

Scalability in Large Graphs As explained in Section 4.2, the time complexity of computing sticky links is $O(n)$ since it encompasses solely linear interpolations of control points. This scalability makes it suitable for larger graphs. In the case of the wiki dataset (Section 5.1), sticky links prove effective in mitigating visual clutter. However, their visual impact on large-scale graphs is related to the quality of the underlying graph layout. In large graphs where nodes are too densely clustered, the use of sticky links may offer limited improvements to readability.

9 Conclusion and Future Work

We presented *Sticky Links*, a new graph edge drawing technique to encode the quantitative data of links by designing a novel visual variable *stickiness*. Sticky links draw the connection as a pair of symmetrical spikes that are attached to each of the linked nodes. The key contribution of our method is the spatially-varying control of shape and length, which brings more visual expressiveness in the representation of quantitative information, compared to the conventional lines with width-even thickness. As we demonstrated, sticky links significantly improve the performance in attribute-related tasks, being more expressive than thickness. Meanwhile, sticky links declutter the graph by reducing ink in the areas where links are overly drawn. Using the Gestalt Continuity principle, connections can be well perceived even when the spikes are not fully connected.

Currently, stickiness is a simple linear function of the quantitative data of graph connections. However, this function can be distilled, by a more elaborate function that may consider the human perception and learn a more elaborated dependency that may lead to a better trade-off between the connectivity perception and graph visual clutterness. For example, the stickiness can be adapted to the zoom level in graph navigation, such that when zooming into local areas, edge drawing turns into more connected links to enhance the connectivity perception as nodes become apart and less visually cluttered.

An important aspect of sticky links is their aesthetic appeal. Previous work mainly considered the placement of nodes and edges in graphs, e.g., maximizing symmetry, minimizing edge bends, etc. Here, the aesthetics of sticky links stem from their visual design itself. However, in our work, we did not emphasize the general eye-pleasing aspect of the graphs, and its potential impact on the effectiveness of node-link graphs, as this requires further validation and research, which are beyond the scope of this work.

Acknowledgments

This work is supported in parts by NSFC (62161146005, U2268205, 42171449, 6230072051), National Key Research and Development Program of China (2021YFB3301500), Shenzhen Science and Technology Program (20231122121504001), Israel Science Foundation (3441/21), China Railway Group Co., Ltd. Science and Technology Research and Development Program (N2022J014), and Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ).

References

[1] J. Abello and F. Van Ham. Matrix zoom: A visual interface to semi-external graphs. In *IEEE symposium on information visualization*, pp. 183–190. IEEE, 2004.

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

[3] M. Baur and U. Brandes. Crossing reduction in circular layouts. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pp. 332–343. Springer, 2004.

[4] R. A. Becker, S. G. Eick, and A. R. Wilks. Visualizing network data. *IEEE Transactions on visualization and computer graphics*, 1(1):16–28, 1995.

[5] C. Bennett, J. Ryall, L. Spalteholz, and A. Gooch. The aesthetics of graph visualization. In *CAe*, pp. 57–64, 2007.

[6] C. Binucci, G. Liotta, F. Montecchiani, and A. Tappini. Partial edge drawing: Homogeneity is more important than crossings and ink. In *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pp. 1–6. IEEE, 2016.

[7] M.-J. Bludau, M. Dörk, and C. Tominski. Unfolding Edges for Exploring Multivariate Edge Attributes in Graphs. In J. Byška, S. Jänicke, and J. Schmidt, eds., *EuroVis 2021 - Posters*. The Eurographics Association, 2021. doi: 10.2312/evp.20211070

[8] T. Bruckdorfer, S. Cornelsen, C. Gutwenger, M. Kaufmann, F. Montecchiani, M. Nöllenburg, and A. Wolff. Progress on partial edge drawings. In *International Symposium on Graph Drawing*, pp. 67–78. Springer, 2012.

[9] T. Bruckdorfer and M. Kaufmann. Mad at edge crossings? break the edges! In *International Conference on Fun with Algorithms*, pp. 40–50. Springer, 2012.

[10] M. Burch. A user study on judging the target node in partial link drawings. In *2017 21st International Conference Information Visualisation (IV)*, pp. 199–204. IEEE, 2017.

[11] M. Burch, H. Schmauder, A. Panagiotidis, and D. Weiskopf. Partial link drawings for nodes, links, and regions of interest. In *2014 18th International Conference on Information Visualisation*, pp. 53–58. IEEE, 2014.

[12] M. Burch, C. Vehlow, N. Konevtsova, and D. Weiskopf. Evaluating partially drawn links for directed graph edges. In *International Symposium on Graph Drawing*, pp. 226–237. Springer, 2011.

[13] W. Büschel, S. Vogt, and R. Dachselt. Investigating link attributes of graph visualizations in mobile augmented reality. In *Proceedings of the CHI 2018 workshop on data visualization on mobile devices. MobileVis*, vol. 18, 2018.

[14] S. Buschmann, M. Trapp, and J. Döllner. Animated visualization of spatial–temporal trajectory data for air-traffic analysis. *The Visual Computer*, 32:371–381, 2016.

[15] S.-H. Cheong and Y.-W. Si. Force-directed algorithms for schematic drawings and placement: A survey. *Information Visualization*, 19(1):65–91, 2020.

[16] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):1–31, 2009.

[17] C. Collins, F. B. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pp. 91–98. IEEE, 2009.

[18] W. Didimo, L. Grilli, G. Liotta, F. Montecchiani, and D. Pagliuca. Visual querying and analysis of temporal fiscal networks. *Information Sciences*, 505:406–421, 2019.

[19] W. Dong, S. Wang, Y. Chen, and L. Meng. Using eye tracking to evaluate the usability of flow maps. *ISPRS International Journal of Geo-Information*, 7(7), 2018.

[20] 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*, pp. 3247–3256, 2013.

[21] P. Eades. A heuristic for graph drawing. *Congressus numerantium*, 42:149–160, 1984.

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

[23] P. Goyal and E. Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[24] H. Guo, J. Huang, and D. H. Laidlaw. Representing uncertainty in graph edges: An evaluation of paired visual variables. *IEEE transactions on visualization and computer graphics*, 21(10):1173–1186, 2015.

[25] N. Henry and J.-D. Fekete. Matrixexplorer: a dual-representation system to explore social networks. *IEEE transactions on visualization and computer graphics*, 12(5):677–684, 2006.

[26] N. Henry, J.-D. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics*, 13(6):1302–1309, 2007.

[27] I. Herman, M. S. Marshall, G. Melançon, D. J. Duke, M. Delest, and J.-P. Domenger. Skeletal images as visual cues in graph visualization. In *Data Visualization'99*, pp. 13–22. Springer, 1999.

[28] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on visualization and computer graphics*, 6(1):24–43, 2000.

[29] 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 *2011 IEEE Pacific Visualization Symposium*, pp. 195–202. IEEE, 2011.

[30] D. Holten and J. J. Van Wijk. Force-directed edge bundling for graph visualization. In *Computer graphics forum*, vol. 28, pp. 983–990. Wiley Online Library, 2009.

[31] D. Holten and J. J. Van Wijk. A user study on visualizing directed edges in graphs. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 2299–2308, 2009.

[32] M. Hummel, F. Klute, S. Nickel, and M. Nöllenburg. Maximizing ink in partial edge drawings of k-plane graphs. In *International Symposium on Graph Drawing and Network Visualization*, pp. 323–336. Springer, 2019.

[33] C. Hurter, O. Ersoy, and A. Telea. Graph bundling by kernel density estimation. In *Computer graphics forum*, vol. 31, pp. 865–874. Wiley Online Library, 2012.

[34] 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, 2014.

[35] T. Jankun-Kelly and K.-L. Ma. Moiregraphs: Radial focus+ context visualization and interaction for graphs with visual nodes. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714)*, pp. 59–66. IEEE, 2003.

[36] B. Kale, M. Sun, and M. E. Papka. The state of the art in visualizing dynamic multivariate networks. In *Computer Graphics Forum*, vol. 42, pp. 471–490. Wiley Online Library, 2023.

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

[38] J. F. Kruiger, P. E. Rauber, R. M. Martins, A. Kerren, S. Kobourov, and A. C. Telea. Graph layouts by t-sne. In *Computer Graphics Forum*, vol. 36, pp. 283–294. Wiley Online Library, 2017.

[39] T. K. Landauer. Behavioral research methods in human-computer interaction. In *Handbook of human-computer interaction*, pp. 203–227. Elsevier, 1997.

[40] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pp. 1–5, 2006.

[41] Y. Liu, T. Safavi, A. Dighe, and D. Koutra. Graph summarization methods and applications: A survey. *ACM computing surveys (CSUR)*, 51(3):1–34, 2018.

[42] Y. Liu, J. Xia, S. Zhou, S. Wang, X. Guo, X. Yang, K. Liang, W. Tu, Z. S. Li, and X. Liu. A survey of deep graph clustering: Taxonomy, challenge, and application. *arXiv preprint arXiv:2211.12875*, 2022.

[43] C. Matuszewski, R. Schönfeld, and P. Molitor. Using sifting for k-layer straightline crossing minimization. In *International Symposium on Graph Drawing*, pp. 217–224. Springer, 1999.

[44] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[45] K. Misue and K. Akasaka. Graph drawing with morphing partial edges. In *International Symposium on Graph Drawing and Network Visualization*, pp. 337–349. Springer, 2019.

[46] S. Navlakha, R. Rastogi, and N. Shrivastava. Graph summarization with bounded error. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 419–432, 2008.

[47] J. Pan, W. Chen, X. Zhao, S. Zhou, W. Zeng, M. Zhu, J. Chen, S. Fu, and Y. Wu. Exemplar-based layout fine-tuning for node-link diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1655–1665, 2020.

[48] H. Purchase. Which aesthetic has the greatest effect on human understanding? In *International Symposium on Graph Drawing*, pp. 248–261. Springer, 1997.

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

[50] E. M. Reingold and J. S. Tilford. Tidier drawings of trees. *IEEE Transactions on software Engineering*, (2):223–228, 1981.

[51] N. H. Riche, T. Dwyer, B. Lee, and S. Carpendale. Exploring the design space of interactive link curvature in network diagrams. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 506–513, 2012.

[52] H. Romat, C. Appert, B. Bach, N. Henry-Riche, and E. Pietriga. Animated edge textures in node-link diagrams: A design space and initial evalua-

tion. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2018.

[53] A. Rusu, A. J. Fabian, R. Jianu, and A. Rusu. Using the gestalt principle of closure to alleviate the edge crossing problem in graph drawings. In *2011 15th International Conference on Information Visualisation*, pp. 488–493. IEEE, 2011.

[54] M. Sathiyanarayanan and D. Pirozzi. Social network visualization: Does partial edges affect user comprehension? In *2017 9th international conference on communication systems and networks (COMSNETS)*, pp. 570–575. IEEE, 2017.

[55] H. Schmauder, M. Burch, and D. Weiskopf. Visualizing dynamic weighted digraphs with partial links. *IVAPP*, 2015:123–130, 2015.

[56] S. Schöffel, J. Schwank, J. Stärz, and A. Ebert. Multivariate networks: A novel edge visualization approach for graph-based visual analysis tasks. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 2292–2298, 2016.

[57] J. Schwank, S. Schöffel, J. Stärz, and A. Ebert. Visualizing uncertainty of edge attributes in node-link diagrams. In *2016 20th International Conference Information Visualisation (IV)*, pp. 45–50, 2016. doi: 10.1109/IV.2016.19

[58] M. A. Smith, B. Shneiderman, N. Milic-Frayling, E. Mendes Rodrigues, V. Barash, C. Dunne, T. Capone, A. Perer, and E. Gleave. Analyzing (social media) networks with nodexl. In *Proceedings of the fourth international conference on Communities and technologies*, pp. 255–264, 2009.

[59] M. Streit, S. Gratzl, M. Gillhofer, A. Mayr, A. Mitterecker, and S. Hochreiter. Furby: fuzzy force-directed bicluster visualization. *BMC bioinformatics*, 15:1–13, 2014.

[60] E. R. Tufte. The visual display of quantitative information. *The Journal for Healthcare Quality (JHQ)*, 7(3):15, 1985.

[61] S. Van den Elzen and J. J. Van Wijk. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2310–2319, 2014.

[62] F. Van Ham. Using multilevel call matrices in large software projects. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714)*, pp. 227–232. IEEE, 2003.

[63] F. Van Ham and J. J. Van Wijk. Interactive visualization of small world graphs. In *IEEE Symposium on Information Visualization*, pp. 199–206. IEEE, 2004.

[64] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner. Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer graphics forum*, vol. 30, pp. 1719–1749. Wiley Online Library, 2011.

[65] M. Wattenberg. Visual exploration of multivariate graphs. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 811–819, 2006.

[66] M. Wertheimer. Untersuchungen zur lehre von der gestalt. ii. *Psychological Research*, 4(1):301–350, 1923.

[67] L. Wilkinson. *The grammar of graphics*. Springer, 1999.

[68] K. Wong and D. Sun. On evaluating the layout of uml diagrams for program comprehension. *Software Quality Journal*, 14(3):233–259, 2006.

[69] W. Yang, M. Liu, Z. Wang, and S. Liu. Foundation models meet visualizations: Challenges and opportunities. *arXiv preprint arXiv:2310.05771*, 2023.

[70] X. Yuan, L. Che, Y. Hu, and X. Zhang. Intelligent graph layout using many users' input. *IEEE transactions on visualization and computer graphics*, 18(12):2699–2708, 2012.