Structural Entropy Based Visualization of Social Networks

Mingliang Xue[1], Lu Chen[2], Chunyu Wei[3], Shuowei Hou[3], Lizhen Cui[1], Oliver Deussen[4] and Yunhai Wang[3]

[1]School of Software & Joint SDU-NTU Center for Artificial Intelligence Research (C-FAIR), Shandong University, China
[2]State Key Lab of CAD&CG, Zhejiang University, China [3]School of Information, Renmin University of China, China
[4] Department of Computer and Information Science, University of Konstanz, Germany

Abstract

Social networks exhibit the small-world phenomenon, characterized by highly interconnected nodes (clusters) with short average path distances. While force-directed layouts are widely employed to visualize such networks, they often result in visual clutter, obscuring community structures due to high node connectivity. In this paper, we present a novel approach that leverages structural entropy and coding trees to enhance community visualization in social networks. Our method computes the structural entropy of graph partitions to construct coding trees that guide hierarchical partitioning with O(E) time complexity. These partitions are then used to assign edge weights that influence attractive forces in the layout, promoting clearer community separation while preserving local cohesion. We evaluate our approach through both quantitative and qualitative comparisons with state-of-the-art community-aware layout algorithms and present two case studies that highlight its practical utility in the analysis of real-world social networks. The results demonstrate that our method enhances community visibility without compromising layout performance. Code and demonstrations are available at `https://github.com/IDEAS-Laboratory/SEL`.

CCS Concepts

• ***Visualization*** → *Information Visualization; Network Visualization; Graph Layout;*

1. Introduction

Visualizing social networks is essential for understanding complex social dynamics and interactions. Among various analytical tasks, community discovery plays a central role in revealing the latent organization within social structures [MBK96, WF94]. To support this, many visualization methods [Fre00] have been developed to help sociologists explore and interpret social network data.

However, social networks often exhibit the *small-world phenomenon*, a property characterized by highly interconnected nodes and, at the same time, short average pairwise distances [WS98]. While this feature can offer insights into real-world social systems [HFB*04, New03], it often leads to dense and entangled visual representations that obscure meaningful structures.

Traditional force-directed layout methods, despite their frequent use in the visualization community, often struggle to produce interpretable visualizations of social networks. These methods typically result in cluttered "hairballs" due to their inability to separate complex connected regions. While recent works [Noa03, STSW24, KRM*17] attempt to incorporate community awareness by adjusting force calculations, they often suffer from limited scalability and poor performance on complex, real-world networks.

To mitigate these issues, alternative strategies have been proposed, including graph simplification [BSST13, ZMT10, RJH11, NLCB13] and clustering-based methods [ZSM09, SHW*19]. However, graph simplification can lead to the loss of critical structures and often suffers from high computational costs. Clustering-based techniques typically rely on shortest path distances, which are less effective in small-world networks where communities are not well-separated by path length. Moreover, these methods tend to be sensitive to outliers, further reducing their reliability in real-world applications.

To address the challenge of visualizing community structures in small-world social networks, we propose *Structural Entropy-based Layout (SEL)*, a novel layout method based on structural entropy [LP16]. Unlike traditional layout algorithms that rely on local heuristics or purely topological cues, SEL incorporates a *global, information-theoretic perspective* to reveal the network's inherent modular organization. Structural entropy measures the information required to describe the connectivity of a network given a specific hierarchical partition. A lower entropy implies a more regularized network organization, corresponding to clearer community structures. SEL builds a coding tree [WCXL22] that captures this hierarchy by recursively merging clusters to minimize entropy using a greedy strategy, revealing the network's multiscale community structure.

However, directly selecting a fixed-level partition from the coding tree often results in overly coarse or overly fragmented community assignments. To overcome this, SEL introduces an adaptive partition selection strategy based on a minimum entropy priority queue. This approach incrementally selects subgraphs for further partitioning by prioritizing the lowest entropy increase, enabling

DOI: 10.2312/pg.20251302

SEL to tailor community resolution locally, *i.e.*, capturing large, dense modules while avoiding over-splitting sparse regions. The resulting partition, composed of nodes at varying depths in the tree, is then integrated into the force-directed layout framework by modulating attractive and repulsive forces based on the structural entropy-derived weights: nodes within the same subtree are pulled closer together, while forces between nodes in distant subtrees are dampened. This entropy-informed force modulation promotes intra-community cohesion and inter-community separation, resulting in clearer and more informative visualizations of community structures in densely connected small-world networks.

We evaluate the layout quality of our method on several benchmark social networks, the largest of which can go up to more than 7,700 nodes and 360,000 edges. Both quantitative and qualitative results demonstrate SEL's ability to generate more meaningful and effective community partitions compared to widely used and latest layout algorithms. Furthermore, we present two case studies on real-world social networks to illustrate the practical utility of SEL in exploratory analysis and visual interpretation.

The main contributions of this work are summarized as follows:

- We propose SEL, a community-aware network layout method based on structural entropy. It employs a priority queue–based partitioning strategy that incrementally selects communities by minimizing entropy increase, enabling adaptive and hierarchical visualization of small-world social networks.
- SEL introduces an entropy-modulated force-directed layout that leverages the structure of a coding tree to adjust force weights, enhancing community separation and improving the interpretability of densely connected networks.
- We demonstrate the effectiveness of SEL through comprehensive comparisons with state-of-the-art layout methods and two case studies on real-world social networks.

2. Related Work

2.1. Force-Directed Graph Layout

Force-directed graph layout methods [Tut63, Kob13] are pivotal for network visualization. The spring-electric model [BK98] simulates nodes as charged particles with spring-connected edges; subsequent refinements include Fruchterman-Reingold's edge length optimization [FR91], SFDP's grid-based repulsion [Hu05], Noack's cluster-separating LinLog model [Noa07], ForceAtlas2's local structure balance [JVHB14], and tFDP's t-distribution acceleration [ZXZ*24]. In contrast, stress models [KK*89] prioritize global structure via energy minimization but sacrifice local details and efficiency—addressed partially by Maxent's distance-limited repulsion [GHN12]. Xue et al. [XWZ*23] proposed the Taurus framework, combining the insights of both force-directed methods and stress models to visualize graphs from multiple perspectives. However, these methods often fail to distinguish communities in real-world networks because they do not highlight the structural identity of communities specifically. We thus propose a community-aware, force-directed technique tailored for small-world networks.

2.2. Community-Aware Network Visualization

Network visualization aims to represent complex structures by spatial layout. These methods can be categorized as follows:

Graph simplification methods reduce the complexity of large networks by retaining only the most important relationships without changing the overall structure. Representative works include Nick et al. [NLCB13], who amplified hidden homophily to reveal community structures; the distance-preserving method [RJH11], which prioritized maintaining node-to-node distances during reduction; and spectral sparsification [BSST13], which leveraged spectral techniques to simplify graphs. In the context of visualizing small-world networks, centrality-based backbone visualization [VHW08] focused on central nodes that play significant roles in the network. Zhou et al. [ZMT10] reduced the size of a graph while keeping its connectivity intact, making the visualization process more manageable without losing essential information. However, these algorithms are inefficient for large networks. Moreover, simplification may remove key relationships, leading to misinterpretations and unclear community structures. Our approach avoids this problem by adjusting node positions instead of dropping edges.

Clustering-based methods are widely used to visualize community structures within networks. Auber et al. [ACJM03] used multi-scale clustering to detect communities at different levels of granularity. Zaidi et al. [ZSM09] uncovered communities in large-scale networks for web page analysis. Persistent homology-guided force-directed method [SHW*19] leveraged topological features to guide graph layout, which improved community detection by focusing on persistent features across scales. Rieck et al. [RFLL17] introduced a topological analysis approach to identify clique-based communities. Huang et al. [HWZ*21] explicitly leveraged community detection to identify highly interconnected subgroups within large networks to enhance visual analytics. Shen et al. [STSW24] proposed GEGraph, an embedding-driven graph layout method that is both aesthetic-aware and community-aware. In addition, many constraint-based force-directed methods [HWW22, RMFS*11] are also suitable for visually representing communities in networks. However, identifying communities is not an easy task, especially in large social networks. To tackle this problem, we adopt structural entropy, a measurement widely used in graph machine learning, to obtain more accurate community partitioning.

2.3. Structural Entropy in Network Analysis

Structural entropy [LP16] is a powerful metric in network science. It quantifies the structural complexity of a graph based on a coding tree, which captures the graph's hierarchical organization. Initially developed for structure analysis in bioinformatics [LYP16, LYX*18], structural entropy has been utilized in various domains including network security [LZP17], community detection [LLZ*19, PZF21], and graph mining [WWW*24, CPYY24]. Integration of structural entropy into neural networks has shown significant success. SEP [WCXL22] introduced the graph pooling operation based on optimal encoding trees to mitigate local structure distortion during coarsening. MinGE [LLS*21] and MEDE [YZW*23] adopt structural entropy to estimate node embedding dimensions in graph neural networks, and SR-MARL [ZPL23] applied structural information principles to hierarchical role discovery in multi-agent reinforcement learning. Despite these advances, structural entropy's use in generating network layouts remains largely unexplored. While the potential in visualizing multi-scale structures has been shown by hierarchical decomposition [LP16], its capacity to enhance community-awareness in social network visualizations remains untapped. To our knowledge, our work is the first to utilize structural entropy's hierarchical encoding capability to address the "hairballs" issue in network visualization, enabling better discovery and exploration of community organization.

3. Structural Entropy-Based Layout

Our approach consists of three key steps to reveal the hierarchical community structure of a network through visualization. First, we construct a coding tree that represents hierarchical graph partitions with minimum structural entropy, providing a multilevel interpretation of the graph's organization for subsequent steps. Second, we introduce a priority queue–based algorithm that progressively determines which non-leaf nodes in the coding tree to divide by prioritizing those with minimal entropy increase when partitioned, allowing cross-hierarchy graph partitions to be selected adaptively. Finally, we design a structure entropy-modulated force-directed layout method that incorporates the selected graph partitions into the layout by adjusting attractive forces based on structural entropy–derived edge weights, effectively separating communities in the visualization while preserving local structures.

3.1. Hierarchical Graph Partitioning via Structural Entropy

To provide a principled multiscale partitioning of the network, we leverage structural entropy [LP16], which quantifies the amount of information required to describe the network's connectivity under a given hierarchical partition. Formally, the structural entropy of a graph $G(V,E)$ with respect to a coding tree T is defined as:

$$H_T(G) = -\sum_{v_t \in T} \frac{g_{v_t}}{\mathrm{vol}(V)} \log \frac{\mathrm{vol}(v_t)}{\mathrm{vol}(v_t^+)}, \tag{1}$$

where v_t denotes a non-root node in T, which corresponds to a subset of graph nodes V as defined by its leaf descendants; v_t^+ is the parent of v_t; and g_{v_t} refers to the number of edges with an endpoint in the leaf node partition of v_t; $\mathrm{vol}(V)$ and $\mathrm{vol}(v_t^+)$ are the sums of degrees of leaf nodes in V and v_t, respectively. A lower value of $H_T(G)$ reflects a more regular and compressible graph structure under the given partition.

The optimal structural entropy of a graph is defined as the minimum value of $H_T(G)$ across all possible coding trees. In practice, we are often interested in partitions at a specific resolution. The *k-dimensional structural entropy* constrains the coding tree to have height k, resulting in the most informative hierarchy with k levels:

$$H^{(k)}(G) = \min_{\forall T:\mathrm{Height}(T)=k} H_T(G). \tag{2}$$

This measure provides a principled way to identify multilevel partitions that balance structure and detail, while suppressing noise and stochastic irregularities.

Figure 1 illustrates how a coding tree partitions a graph recursively. In subfigure (a), a three-level coding tree is depicted, where each non-leaf node corresponds to a subgraph consisting of all its leaf nodes, and its corresponding subgraph can continue to be partitioned into smaller subgraphs by expanding the non-leaf node into the next level of non-leaf nodes. The graph partitions corresponding to each level are visualized in Figure 1(b)-(d), where each node in the graph is drawn in the same color as its corresponding partition (non-leaf node) in the coding tree.

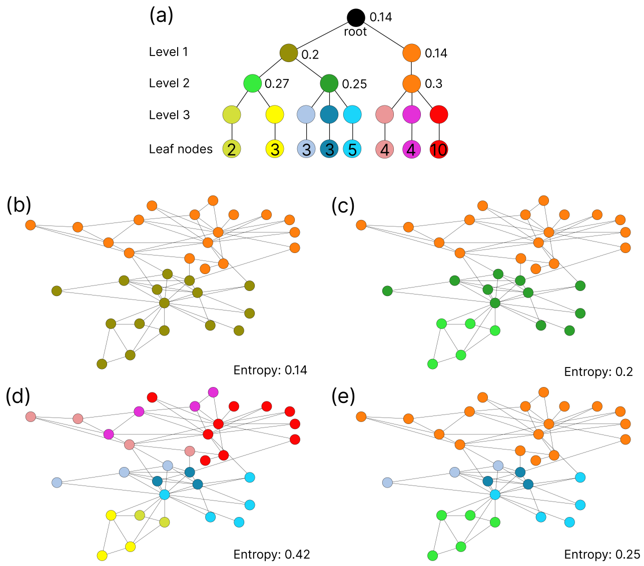

Figure 1: *An example of coding tree on Zachary's karate club dataset [Zac77]: (a) The coding tree with height $k = 3$. Numbers next to non-leaf nodes represent the structural entropy of partitioning their corresponding subgraphs, while labels on leaf nodes indicate the size of the subgraph they represent. Node colors represent community ID. (b-d) Graph partitions corresponding to levels 1 to 3 of the coding tree. (e) Graph partitioning produced by our minimum entropy priority queue method. Compared to (c), it reveals finer-grained clusters, and unlike (d), it avoids over-fragmented clusters while achieving a lower total structural entropy.*

To compute the optimal coding tree with a desired hierarchical depth k, we employ a three-stage algorithm [WCXL22]: (1) Bottom-up construction: Build a full-height binary coding tree by iteratively merging two child nodes to form a new division, maximizing structural entropy reduction at each step. (2) Tree compression: Reduce the tree to height k by selectively removing intermediate nodes in a way that maintains minimal structural entropy. (3) Cross-layer link resolution: To ensure structural consistency, we add intermediate nodes where necessary to preserve valid hierarchical relationships between layers without altering the computed entropy.

The resulting coding tree T represents a hierarchy of communities at multiple scales (k levels). This algorithm operates with complexity $O(2|V| + h_{max}(|E|\log|V| + |V|))$, where h_{max} is the height of the initially constructed full tree. Since the tree tends to be balanced through structural entropy minimization, h_{max} is typically around $\log|V|$, making the algorithm scale almost linearly with the number of edges in practice.

3.2. Adaptive Partition with Minimum Entropy Priority Queue

Although the coding tree provides a full hierarchical partitioning of the graph, directly selecting a fixed-level partition (e.g., all non-leaf tree nodes at level k) often leads to over- or under-partitioning of the graph. Specifically, deeper levels may split tightly connected

Algorithm 1 Pseudo-code for Minimum Entropy Priority Queue

Input: graph $G=(V,E)$, partition tree T.
Initialization: priority queue $Q \leftarrow [T.root]$, $cur\_T \leftarrow \{\}$, $last\_node \leftarrow T.root$, edge weights $W \leftarrow \{\}$, entropy $E_l = 0$.
repeat
    $cur\_node \leftarrow Q.pop()$
    $cur\_T.add(cur\_node)$
    $E_c \leftarrow totalStructuralEntropy(cur\_T)$
    if cur_node has child tree nodes: **then**
        for all $cur\_node.child$: **do**
            $Q.push(child)$
        end for
    end if
    for all edges $(i,j))$ in G: **do**
        if i,j are from distinct non-leaf nodes in Q **then**
            $W[(i,j)] \leftarrow totalStructuralEntropy(cur\_T)$
        end if
    end for
    $last\_node \leftarrow cur\_node$
    $E_l \leftarrow E_c$
until Q is empty
return W

communities into overly fine groups or even isolated nodes, while shallower levels may merge distinct substructures into overly coarse clusters. To address this, we propose an adaptive partitioning strategy that selects communities across different levels of the hierarchy based on structural entropy, ensuring that these partitions better reflect the actual organization of the graph.

Our key idea is to incrementally select which non-leaf nodes to partition by prioritizing those that contribute the least to the overall structural entropy when split. This is achieved by introducing a *minimum entropy priority queue*. Specifically, each non-leaf node in the coding tree is assigned a priority value that is equal to the entropy cost of partitioning that node. The queue is initialized with the root of the coding tree. At each iteration, the node that leads to the lowest entropy increase when partitioned is removed from the queue, and the tree is partitioned accordingly. Then, the child nodes (if any) of the partitioned node are added to the queue with their corresponding entropy scores. This greedy strategy avoids partitioning entire levels at once; instead, it decomposes the tree adaptively, allowing some communities to be represented by higher-level clusters and others by more fine-grained divisions. For each partitioning, we assign weights to inter-community edges, which reflect the structural entropy at the moment the edge is cut. These edge weights are later used to guide layout optimization, allowing both strongly and weakly connected regions to be visualized faithfully.

The pseudo-code for our adaptive selection of partitions is shown in Algorithm 1, where a priority queue is initialized from the root node of the coding tree. We then progressively remove tree nodes from the queue and add them to the set of current tree nodes (cur_T) to calculate their entropy values. If the current tree node has children, the child nodes are added back to the queue to continue the traversal. For each edge (i,j) in the graph, if node i and j are from subgraphs represented by different non-leaf tree nodes in the queue, the weight of that edge is set to the current structural entropy. After traversing all non-leaf nodes in the coding tree, the algorithm finally returns a set of all the edge weights W to be used for layout.

For visual illustration, Figure 1 employs Zachary's karate club network [Zac77] to explain how our minimum entropy priority queue algorithm selects the optimal partition of a graph. The algorithm starts by partitioning the root node of the coding tree in (a), yielding subfigure (b). Following the minimum entropy principle, it then partitions the orange tree node. However, as the orange node has only one child, no change occurs in the resulting partition. Subsequent partitioning of the brown and dark green tree nodes leads to (c) and (e), respectively. The final layout (e) reveals five communities that better describe the organization of the network than naively adopting the three-level partition in (d), which shows eight overly fragmented small clusters. This adaptive strategy leads to more interpretable and visually coherent layouts by preserving balanced community sizes and reducing outliers.

3.3. Structural Entropy-Modulated Force-Directed Layout

After determining the community partition based on the structural entropy and the coding tree, we can design the layout based on the partition results to produce community-aware visualizations. Our goal is to minimize the distance between nodes in the same community, while maximizing the distance between nodes in different communities. To achieve this, we refer to the idea of maximum entropy layout [GHN12] and maximize the layout entropy (H) while constraining attractive forces between nodes in the same community. The optimization model is formulated as follows:

$$max_x \quad H(\mathbf{X}) = \sum_{(i,j)\in E} \ln \|\mathbf{x}_i - \mathbf{x}_j\| \tag{3}$$

$$\text{s.t.} \quad \min \sum_{c\in C} \sum_{(i,j)\in c} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \tag{4}$$

Where X is the layout position of nodes, C is the set of communities, and c represents each community in C.

Based on this formulation, we further design a force-directed algorithm to generate the corresponding layouts. Our strategy is to adjust the weight of the attractive force according to the weight of the edge based on the structural entropy, and achieve the effect of partitioning communities by reducing the attractive force corresponding to the edges between different communities. This strategy is adaptable across all force-directed models. For example, taking the attractive and repulsive forces in FR [FR91] as a blueprint, our force model unfolds as follows:

$$F^a_{i,j} = w_{i,j}||\mathbf{x}_i - \mathbf{x}_j||^2, \tag{5}$$

$$F^r_{i,j} = 1/||\mathbf{x}_i - \mathbf{x}_j||, \tag{6}$$

where $w_{i,j}$ is the coefficient related to edge weights, we determine the partitioning of the community in the layout by adjusting this coefficient. Since Algorithm 1 has calculated all hierarchical partition results and mapped them to edge weights, we can adjust the mapping function and its associated parameters to achieve different partitioning effects.

$$w_{i,j} = \begin{cases} f_{\text{map}}\left(1 - \frac{\text{SE}_{\text{target}}}{W[i,j]}\right) & \text{if } \frac{\text{SE}_{\text{target}}}{W[i,j]} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

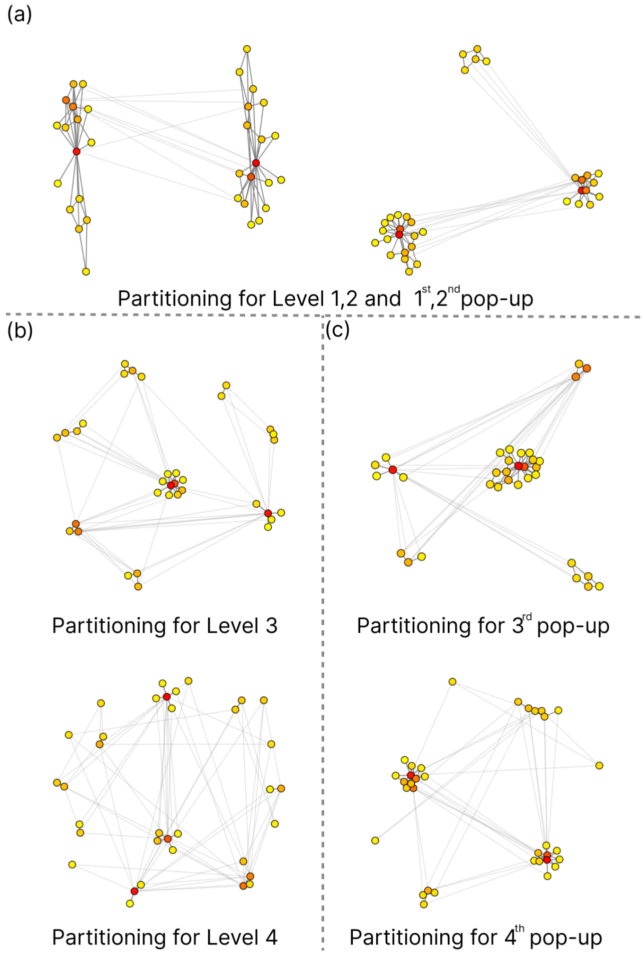

Figure 2: *Comparison between minimum entropy priority queue-based partition selection and fixed-level partitioning from the coding trees on Zachary's Karate Club dataset [Zac77] in Figure 1: (a) Layout after the first and second partitions using either fixed-level or minimum entropy priority queue strategy; (b) Layout after two additional levels of partitioning using the fixed-level approach; (c) Layout after further partitioning the third and fourth non-leaf nodes using the priority queue method. In all subfigures, nodes with higher degrees are rendered with higher red hues to indicate centrality.*

where SE_{target} [†] represents the value of structural entropy corresponding to the partition used for the final layout. The mapping function f_{map} can be *linear*, *Sigmoid*, or *tanh*. Based on our tests, the effects of these three functions are similar. Influenced by such weights, the attractive force between neighboring nodes assigned to different communities will be zero, while those within the same community will persist. Moreover, nodes from different communities will be pushed away by repulsive forces, resulting in clear and separated groups in the layout.

Figure 2 illustrates the intermediate force-directed layouts of our adaptive graph partitioning based on Algorithm 1 compared to naive fixed-level partition selection, using Zachary's Karate Club dataset [Zac77] as an example network. Subfigure (a) presents the results of our SEL that partitions the first two non-leaf nodes in Figure 1, which are equivalent to directly selecting the first and second levels of the coding tree as partitions. The difference emerges when computing finer communities. Subfigure (b) shows the layouts obtained by directly applying the third and fourth levels of the coding tree for partitioning, while (c) displays the layout generated by SEL using the minimum entropy priority queue. We observe that fixed-level partitioning in (b) causes early fragmentation of communities, leading to overly scattered layouts, particularly at level 4. In contrast, (c) demonstrates that our entropy-prioritized queue, combined with the entropy-modulated force-directed layout, better preserves community structure. Furthermore, by tuning the SE_{target} value, SEL allows flexible control over the granularity of community partitioning in the visualization.

4. Evaluation

We compared our method and five other layout methods that consider community representation from three perspectives. Firstly, we use a set of quantitative metrics to measure the community awareness and readability of the layout generated by each method with multiple real social networks. Secondly, we show the layout quality qualitatively for all the layout methods. Lastly, we present two case studies to demonstrate the application potential of our method.

4.1. Experiment Setting

Datasets. Table 1 describes the information of the graph datasets used for the evaluation, including the number of nodes and edges. These networks are derived from the real social network data in the Facebook100 dataset [RA15].

Table 1: *The statistics of tested graphs.*

Name	#Nodes	#Edges	Name	#Nodes	#Edges
Reed98	962	18812	Trinity100	2613	111996
Haverford76	1446	59589	USFCA72	2682	65252
Simmons81	1518	32988	American75	6386	217662
Swarthmore42	1659	61050	WashU32	7755	367541

Baselines. We compared SEL with five representative layout methods: FR [FR91], Linlog [Noa03], tFDP [ZXZ*24], GEGraph [STSW24], and PH [SHW*19]. Among them, FR is the most classical force-directed layout algorithm. Linlog and tFDP are layout algorithms that reflect community awareness by adjusting their force calculation. Linlog emphasizes cluster separation through a strong short-range attractive force, while tFDP prioritizes clustering through its parameter and interaction scheme. GEGraph is an embedding-based layout algorithm that integrates graph topology and node attributes into a latent space to produce layouts with community preservation. PH is a clustering-based method that allows users to manually identify clusters on graphs through interaction. We exclude graph simplification approaches from our comparison, as these methods fundamentally alter the network structure by removing edges. In contrast, SEL preserves the complete topology and

[†] In our experiments, we use a simple heuristic for determining the threshold parameter SE_{target}. First, we use half of the maximum structural entropy as an upper bound to prevent excessive partitioning. Then, we apply the elbow method to the entropy decay curve to identify the threshold at the point where a steep drop transitions into minor subsequent changes.

Proceedings published by Eurographics - The European Association for Computer Graphics.

reveals community structure purely through layout. This fundamental difference makes direct comparison with simplification-based techniques methodologically inappropriate.

Parameters. We used the default parameters in the official code of each baseline. An exception was made for tFDP, where we opted for clustering-oriented parameters delineated in their demo ($\alpha = 0.1, \beta = 8, \gamma = 5$). We use the sigmoid function in the weight calculation of the attractive forces for SEL. In the PH method, to be consistent with SEL, we select the longest k bars in the interactive frame of the barcode to determine the layout, where k is the number of non-leaf tree nodes that were partitioned in the SEL.

4.2. Quantitative Comparison

Evaluation Metrics. For our assessment of community discovery and readability of SEL and baseline methods, we employ six key metrics. As our test graphs primarily consist of social networks lacking predefined cluster labels, we initially utilize the DBSCAN algorithm [EKSX96] to generate cluster labels within the layout results. The following are the definitions of these six metrics:

- *Cluster Separation* (*CS*) quantifies the differentiation between different clusters within the layout [VHR08]. We have simplified this metric by evaluating the ratio of the nearest cluster distance to the maximum intra-cluster dispersal, indicating cluster distinctiveness: $\mathrm{CS} = \frac{\bar{d}_{\text{intra}}}{\bar{d}_{\text{intra}} + \bar{r}_{\text{inter}}}$, where $\bar{d}_{\text{intra}}$ denotes the average radius within a cluster and $\bar{r}_{\text{inter}}$ denotes the average distance between clusters. A higher value indicates better cluster separation.
- *Silhouette Coefficient* [Řez18] (*SC*) reflects the compactness and separation of clustering structures: $SC = \frac{1}{|V|}\sum_{|V|}^{i} \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$, where $a(i)$ is the average distance from node i to other nodes in the same cluster, and $b(i)$ is the average distance from node i to all node in the nearest other clusters. Higher values signify clearer cluster boundaries.
- *Noise Ratio* (*NR*) indicates the percentage of outlier nodes not reliably assigned to any cluster: $NR = \frac{n}{|V|}$, where n is the number of nodes labeled as noise. Lower values suggest better community partitioning capabilities.
- *Edge Crossings* [Pur02] (*EC*) evaluate the frequency of edge intersections: $EC = 1 - e^{c/c_{max}}$, where c is the number of edge crossings, and c_{max} is the maximal number in each graph. A lower value indicates fewer edge crossings.
- *Node Occlusions* [HWP*19] (*NO*) evaluates the extent of visual information loss in the layout due to spatial overlaps in the layout: $NO = \frac{OP}{\frac{1}{2}|V|^2}$, where OP represents the number of the occluded node pairs, for which the distance between the two nodes is less than a threshold in the layout.
- *Ideal Edge Length* (*IL*) [Pur02] assesses the consistency between geometric distances in the layout and topological distances in the graph: $IL = \frac{1}{|E|}\sum_{(i,j)\in E}\frac{(||\mathbf{x}_i - \mathbf{x}_j|| - \ell)^2}{\ell^2}$, where ℓ is 1 by default.

To maintain consistency where smaller values are preferable, we reflect the final values v of *CS* and *SC* as $1 - v$.

Result Analysis. The heatmap shown in Figure 3 illustrates the *CS*, *SC*, *NR*, *EC*, *NO*, and *IL* of the layouts produced by six baseline methods across eight test graphs. Since the primary goal of community-aware network visualization is to reveal the community

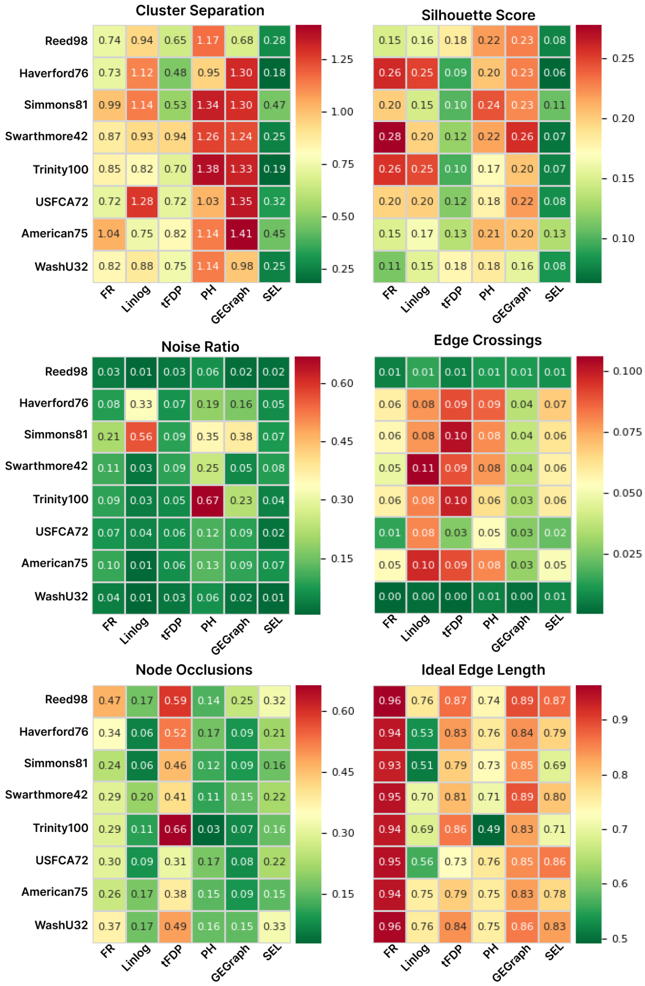

Figure 3: *Heatmaps to present the values of six metrics for visualizations generated by six layout methods on eight tested networks.*

structures within the layout, a successful visualization is expected to excel in *CS*, *SC*, and *NR*, while not too bad in *EC*, *NO*, and *IL* in case that it hampers users from understanding the visualization.

For *CS*, *SC*, and *NR*, SEL performs well on almost all datasets, indicating that our method is effective in generating community-aware layouts. The average values of SEL proportionate to FR, Linlog, tFDP, PH, and GEGraph on *CS* are merely 35%,30%, 42%, 25%, and 25%, respectively. Similarly, for *SC* and *NR*, our SEL showcases proportionate average values of 42%, 44%, 67%, 42%, 39% and 50%, 35%, 75%, 20%, 34% compared to the aforementioned baseline methods, respectively. This highlights the advantage of our structural entropy-based segmentation method. We attribute the poor *CS* and *SC* performance of FR, Linlog, and tFDP to their monolithic representation of the whole network without specialized modeling of the communities. As for PH and GEGraph, their reliance on user-specified clusters and node labels makes them unable to autonomously craft suitable partitions, and thus less effective in visualizing unlabeled social networks.

In terms of *EC*, SEL closely trails GEGraph and FR across most datasets. The average *EC* values of SEL proportionate to FR, Linlog,

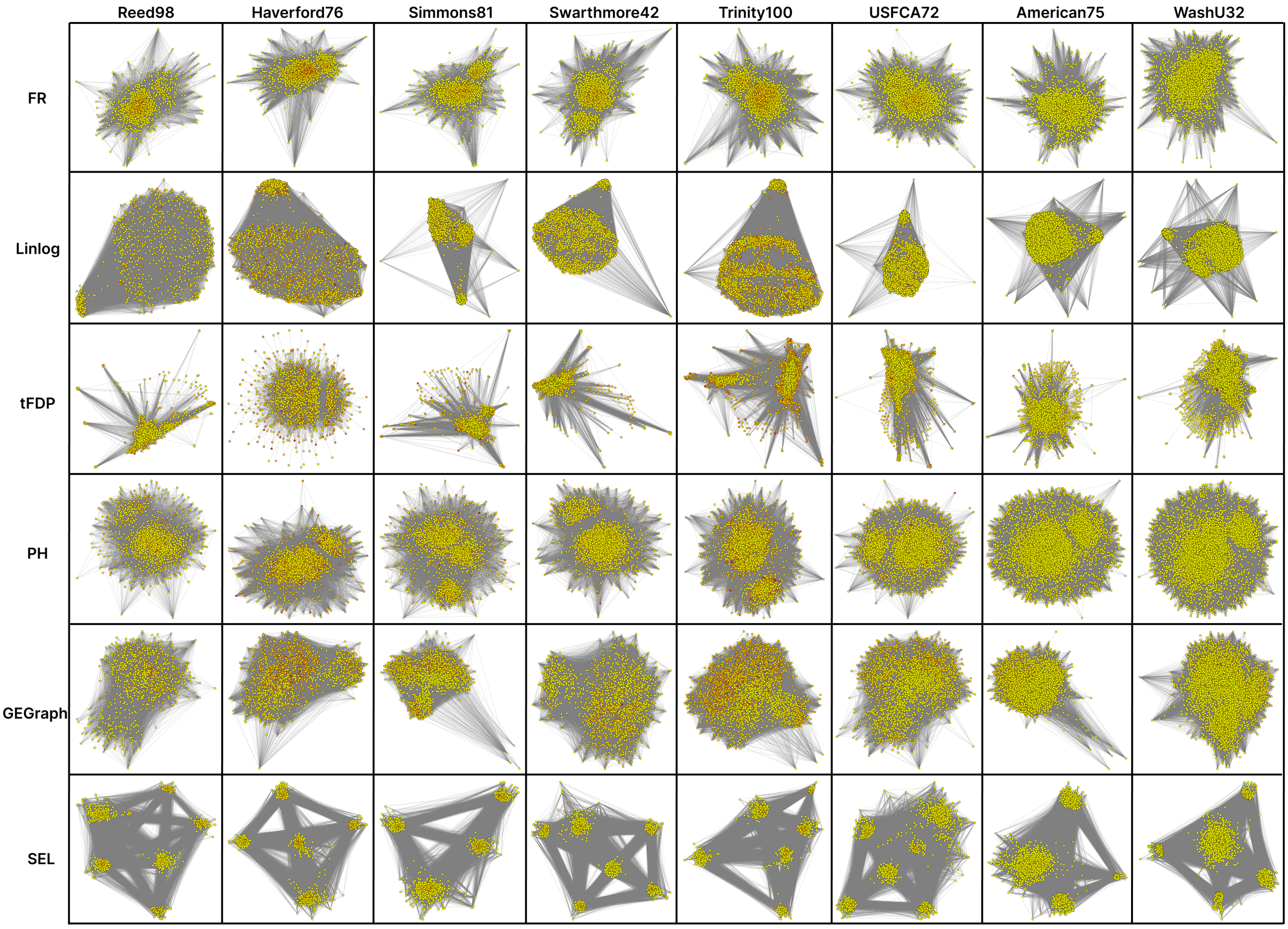

Figure 4: *Layout results generated by six methods for the eight social networks in Facebook100 dataset [RA15]. Nodes with higher degrees tend to be more inclined towards the color red.*

tFDP, PH, and GEGraph are 113%, 63%, 66%, 73%, and 154%, respectively. For *NO*, SEL underperforms Linlog, PH, and GEGraph by relative 42%, 40%, and 45%, but surpasses FR and tFDP by more than 2 times. Regarding *IL*, SEL slightly lags behind Linlog and PH by 17% and 10% relatively, while outperforming GEGraph, FR, and tFDP by large percentages. Considering all three aesthetic metrics, SEL is at an intermediate level that does NOT affect the readability of the layout significantly.

In summary, SEL excels at generating community-aware layouts and ensures a balanced level of readability, fulfilling our objectives.

4.3. Qualitative Comparison

The visualizations generated by SEL and baseline methods are illustrated in Figure 4. FR fails to capture community information within social networks, exhibiting limited capacity to represent complex social networks. Linlog excels in revealing cluster structures, but for complex social networks, it struggles to divide large clusters properly, as shown by its regular simplification of the network into two major clusters and several outliers. tFDP can aggregate closely related nodes when utilizing special clustering-oriented parameters, yet it fails to distinguish different intricate communities in small-world networks, and produces even more outliers than Linlog. On the other hand, PH heavily relies on user-specified clusters, requiring manual effort to identify and separate communities. For our tested unlabeled social networks, both PH and GEGraph generate overly coarse partitions and fail to uncover meaningful organization.

In contrast, SEL operates without user-defined partitions, node attributes, or labels. It automatically detects communities based on structural entropy and clearly expresses them in the layout. SEL effectively separates communities, maintains a balanced partitioning, and reduces the number of outliers. Overall, SEL outperforms the baselines by clearly revealing community structures and is especially well-suited for analyzing unlabeled social networks.

4.4. Case Studies

We present two case studies on real-world networks to illustrate how SEL generates community-aware layouts and supports multi-resolution exploration.

Les Misérables [Knu93]: This is a character relationship network from the famous novel Les Misérables. Nodes represent characters,

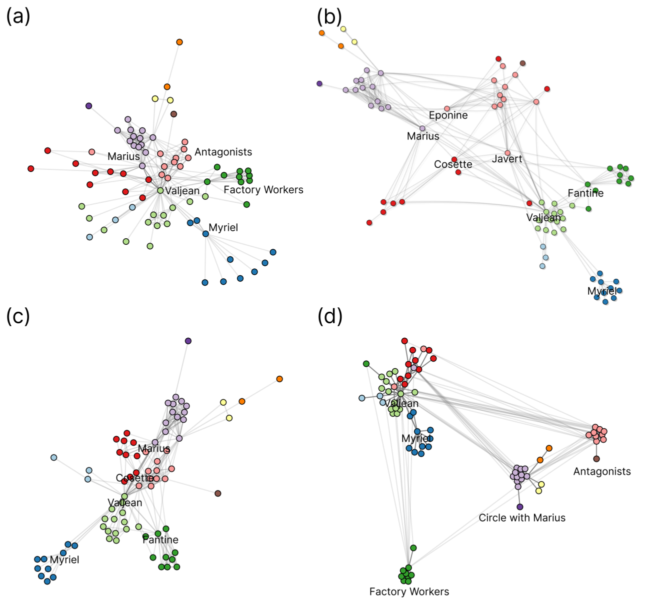

Figure 5: *Layout results from different methods of the Les Miserables dataset [Knu93]: (a) FR [FR91], (b) PH [SHW*19], (c) GEGraph [STSW24], and (d) SEL (ours).*

and edges indicate that the two characters appeared in the same chapter. Figure 5(a) is the classic FR layout, where inter-node distances are relatively uniform. Nodes belonging to the same community are not placed closely enough to distinguish different communities. As shown in Figure 5(b), PH places nodes of the same community closer together than FR. However, this result emphasizes individual character relationships, such as those involving Javert, Cosette, Marius, and Eponine, rather than highlighting the broader community structure of the network. Figure 5(c) is generated by GEGraph, which achieves tighter clustering within communities but does not effectively capture the distinctions between different communities. In contrast, in our layout (Figure 5(d)), the central character Valjean is positioned within the largest community, while three related communities, the Factory Workers, the circle around Marius, and the Antagonists, are distinctly separated from Valjean's main network. Although SEL's layout is denser and slightly less readable than PH's, it more clearly reflects the community structure inherent in the characters' relationships.

Facebook4039 [LM12]: This dataset is a subset of the Facebook social network, comprising 4,039 nodes and 88,234 edges. Figure 6 demonstrates the hierarchical layouts generated by SEL, where node colors indicate community assignments based on coding tree partitions. Subfigures (a)–(d) correspond to different $\mathrm{SE}_{\mathrm{target}}$ values: 0.001, 0.025, 0.065, and 0.1, respectively. In (a) and (b), the small target structural entropy values lead to coarse partitions with fewer communities. Compared to (a), (b) distinguishes more communities and captures finer-grained local structures while maintaining an overall coherent layout. As the target structural entropy increases in (c) and (d), the layout becomes more fragmented, with many peripheral nodes identified as outliers. Despite this, the backbone structure of the network and the fine community divisions are still clearly visible. This highlights SEL's strength in supporting multi-resolution community analysis: it enables users to interactively explore the graph structure at varying levels of granularity while maintaining visually meaningful and interpretable layouts.

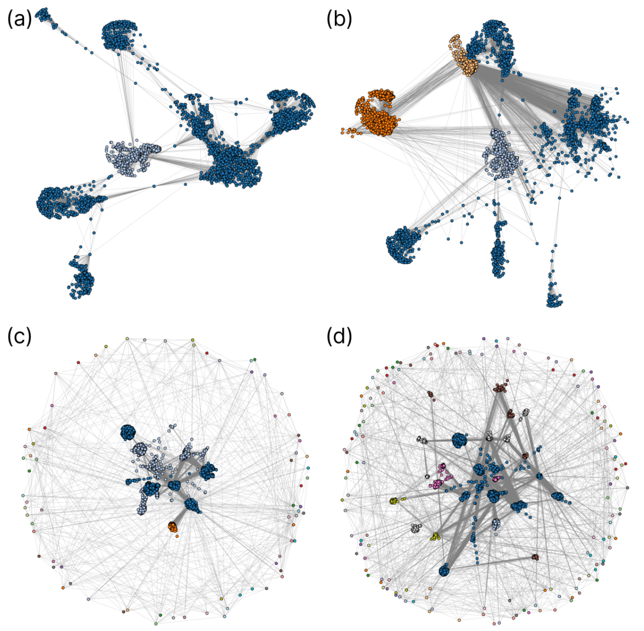

Figure 6: *Layouts of the Facebook4039 dataset [LM12] generated by SEL with target structural entropy values of (a)* 0.001*, (b)* 0.025*, (c)* 0.065*, and (d)* 0.1*, respectively.*

5. Conclusion

In this paper, we introduced SEL, a social network visualization method based on structural entropy and coding trees. SEL addresses the key challenge of visualizing small-world networks by leveraging a minimum entropy priority queue–based partitioning strategy and entropy-modulated force-directed layout. This enables SEL to create community-aware visualizations that preserve intra-cluster cohesion, outperforming traditional methods as shown in our quantitative and qualitative evaluations. SEL supports many social network analysis tasks, such as community detection, clique targeting, and subgroup exploration. One limitation of SEL is that the dense node arrangement may hinder the observation of intra-community structures, which can be alleviated via local refinement in existing interactive systems. In future work, we plan to explore broader applications of structural entropy in visualization.

Acknowledgments

This work is supported by grants from the NSFC (No.62402284, No.92367202, No.62132017, and No.U2436209), NSF of Shandong province (No.ZR2024QF212 and No.ZQ2022JQ32), National Key R&D Program of China (No.2022ZD0160805), the Beijing Natural Science Foundation (No.L247027), the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China.

References

[ACJM03] Auber D., Chiricota Y., Jourdan F., Melançon G.: Multiscale visualization of small world networks. In *IEEE symposium on information visualization 2003 (IEEE Cat. No. 03TH8714)* (2003), IEEE, pp. 75–81. doi:10.1109/INFVIS.2003.1249011. 2

[BK98] Biedl T., Kant G.: A better heuristic for orthogonal graph drawings. *Computational Geometry 9*, 3 (1998), 159–180. URL: https://www.sciencedirect.com/science/article/pii/S0925772197000266, doi:10.1016/S0925-7721(97)00026-6. 2

[BSST13] Batson J., Spielman D. A., Srivastava N., Teng S.-H.: Spectral sparsification of graphs: theory and algorithms. *Communications of the ACM 56*, 8 (2013), 87–94. doi:10.1145/2492007.2492029. 1, 2

[CPYY24] Cao Y., Peng H., Yu Z., Yu P. S.: Hierarchical and incremental structural entropy minimization for unsupervised social event detection. In *Proceedings of the AAAI conference on artificial intelligence* (2024), vol. 38, pp. 8255–8264. 2

[EKSX96] Ester M., Kriegel H.-P., Sander J., Xu X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (1996), KDD'96, AAAI Press, p. 226–231. URL: http://www.aaai.org/Library/KDD/1996/kdd96-037.php. 6

[FR91] Fruchterman T. M., Reingold E. M.: Graph drawing by force-directed placement. *Software: Practice and Experience 21*, 11 (1991), 1129–1164. doi:10.1007/978-3-658-21742-6_49. 2, 4, 5, 8

[Fre00] Freeman L. C.: Visualizing social networks. *Journal of social structure 1*, 1 (2000), 4. URL: https://www.cmu.edu/joss/content/articles/volume1/Freeman.html. 1

[GHN12] Gansner E. R., Hu Y., North S.: A maxent-stress model for graph layout. *IEEE Transactions on Visualization and Computer Graphics 19*, 6 (2012), 927–940. doi:10.1109/TVCG.2012.299. 2, 4

[HFB*04] Horn D. B., Finholt T. A., Birnholtz J. P., Motwani D., Jayaraman S.: Six degrees of jonathan grudin: a social network analysis of the evolution and impact of cscw research. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (2004), pp. 582–591. doi:10.1145/1031607.1031707. 1

[Hu05] Hu Y.: Efficient high-quality force-directed graph drawing. *Mathematica Journal 10*, 1 (2005), 37–71. 2

[HWP*19] Haleem H., Wang Y., Puri A., Wadhwa S., Qu H.: Evaluating the readability of force directed graph layouts: A deep learning approach. *IEEE computer graphics and applications 39*, 4 (2019), 40–53. doi:10.1109/MCG.2018.2881501. 6

[HWW22] Han L., Wang B., Wang S.: Force-directed graph layout based on community discovery and clustering optimization. In *International Conference on Medical Imaging and Computer-Aided Diagnosis* (2022), Springer, pp. 561–571. doi:https://doi.org/10.1007/978-981-16-6775-6_46. 2

[HWZ*21] Huang Z., Wu J., Zhu W., Wang Z., Mehrotra S., Zhao Y.: Visualizing complex networks by leveraging community structures. *Physica A: Statistical Mechanics and Its Applications 565* (2021), 125506. doi:10.1016/j.physa.2020.125506. 2

[JVHB14] Jacomy M., Venturini T., Heymann S., Bastian M.: Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLOS ONE 9*, 6 (2014), e98679. doi:10.1371/journal.pone.0098679. 2

[KK*89] Kamada T., Kawai S., et al.: An algorithm for drawing general undirected graphs. *Information Processing Letters 31*, 1 (1989), 7–15. doi:10.1142/9789814434478_0005. 2

[Knu93] Knuth D. E.: The stanford graphbase: a platform for combinatorial algorithms. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms* (USA, 1993), SODA '93, Society for Industrial and Applied Mathematics, p. 41–43. URL: https://dl.acm.org/doi/10.5555/313559.313609. 7, 8

[Kob13] Kobourov S. G.: *Force-Directed Drawing Algorithms*. Chapman and Hall/CRC, Oxford, 2013. doi:10.1201/b15385-15. 2

[KRM*17] Kruiger J. F., Rauber P. E., Martins R. M., Kerren A., Kobourov S., Telea A. C.: Graph layouts by t-SNE. *Computer Graphics Forum 36*, 3 (2017), 283–294. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13187, doi:10.1111/cgf.13187. 1

[LLS*21] Luo G., Li J., Su J., Peng H., Yang C., Sun L., Yu P. S., He L.: Graph entropy guided node embedding dimension selection for graph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence* (2021), pp. 2767–2774. doi:10.24963/ijcai.2021/381. 2

[LLZ*19] Liu Y., Liu J., Zhang Z., Zhu L., Li A.: Rem: From structural entropy to community structure deception. *Advances in Neural Information Processing Systems 32* (2019). URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/328347805873e9a9c700591812fb0ec2-Paper.pdf. 2

[LM12] Leskovec J., Mcauley J.: Learning to discover social circles in ego networks. *Advances in neural information processing systems 25* (2012). URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/7a614fd06c325499f1680b9896beedeb-Paper.pdf. 8

[LP16] Li A., Pan Y.: Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory 62*, 6 (2016), 3290–3339. doi:10.1109/TIT.2016.2555904. 1, 2, 3

[LYP16] Li A., Yin X., Pan Y.: Three-dimensional gene map of cancer cell types: Structural entropy minimisation principle for defining tumour subtypes. *Scientific reports 6*, 1 (2016), 1–26. doi:10.1038/srep20412. 2

[LYX*18] Li A., Yin X., Xu B., Wang D., Han J., Wei Y., Deng Y., Xiong Y., Zhang Z.: Decoding topologically associating domains with ultra-low resolution hi-c data by graph structural entropy. *Nature communications 9*, 1 (2018), 1–12. doi:10.1038/s41467-018-05691-7. 2

[LZP17] Li A., Zhang X., Pan Y.: Resistance maximization principle for defending networks against virus attack. *Physica A: Statistical Mechanics and its Applications 466* (2017), 211–223. doi:10.1016/j.physa.2016.09.009. 2

[MBK96] McGrath C., Blythe J., Krackhardt D.: Seeing groups in graph layouts. *Connections 19*, 2 (1996), 22–29. URL: https://www.heinz.cmu.edu/faculty-research/profiles/krackhardt-davidm/_files/1996-seeing-groups-in-graph-layout.pdf. 1

[New03] Newman M. E.: The structure and function of complex networks. *SIAM review 45*, 2 (2003), 167–256. doi:10.1137/S003614450342480. 1

[NLCB13] Nick B., Lee C., Cunningham P., Brandes U.: Simmelian backbones: Amplifying hidden homophily in facebook networks. In *Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining* (2013), pp. 525–532. doi:10.1145/2492517.2492569. 1, 2

[Noa03] Noack A.: An energy model for visual graph clustering. In *International Symposium on Graph Drawing* (2003), Springer, pp. 425–436. doi:10.1007/978-3-540-24595-7_40. 1, 5

[Noa07] Noack A.: Energy models for graph clustering. *Journal of Graph Algorithms and Applications 11*, 2 (2007), 453–480. doi:10.7155/jgaa.00154. 2

[Pur02] Purchase H. C.: Metrics for graph drawing aesthetics. *Journal of Visual Languages and Computing 13*, 5 (2002), 501–516. doi:10.1006/jvlc.2002.0232. 6

[PZF21] Pan Y., Zheng F., Fan B.: An information-theoretic perspective of hierarchical clustering. *arXiv preprint arXiv:2108.06036* (2021). doi:10.48550/arXiv.2108.06036. 2

[RA15] ROSSI R. A., AHMED N. K.: The network data repository with interactive graph analytics and visualization. In *AAAI* (2015). URL: https://networkrepository.com. 5, 7

[Řez18] ŘEZANKOVÁ H.: Different approaches to the silhouette coefficient calculation in cluster evaluation. In *21st international scientific conference AMSE applications of mathematics and statistics in economics* (2018), pp. 1–10. doi:https://www.amse-conference.eu/wp-content/uploads/2018/09/AMSE_2018_Rezankova.pdf. 6

[RFLL17] RIECK B., FUGACCI U., LUKASCZYK J., LEITTE H.: Clique community persistence: A topological visual analysis approach for complex networks. *IEEE transactions on visualization and computer graphics 24*, 1 (2017), 822–831. doi:10.1109/TVCG.2017.2744321. 2

[RJH11] RUAN N., JIN R., HUANG Y.: Distance preserving graph simplification. In *2011 IEEE 11th International Conference on Data Mining* (2011), IEEE, pp. 1200–1205. doi:10.1109/ICDM.2011.57. 1, 2

[RMFS*11] RODRIGUES E. M., MILIC-FRAYLING N., SMITH M., SHNEIDERMAN B., HANSEN D.: Group-in-a-box layout for multi-faceted analysis of communities. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing* (2011), IEEE, pp. 354–361. doi:10.1109/PASSAT/SocialCom.2011.139. 2

[SHW*19] SUH A., HAJIJ M., WANG B., SCHEIDEGGER C., ROSEN P.: Persistent homology guided force-directed graph layouts. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 697–707. doi:10.1109/TVCG.2019.2934802. 1, 2, 5, 8

[STSW24] SHEN L., TAI Z., SHEN E., WANG J.: Graph exploration with embedding-guided layouts. *IEEE Transactions on Visualization and Computer Graphics 30*, 7 (2024), 3693–3708. doi:10.1109/TVCG.2023.3238909. 1, 2, 5, 8

[Tut63] TUTTE W. T.: How to draw a graph. *Proceedings of the London Mathematical Society s3-13*, 1 (1963), 743–767. URL: https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s3-13.1.743, arXiv:https://londmathsoc.onlinelibrary.wiley.com/doi/pdf/10.1112/plms/s3-13.1.743, doi:10.1112/plms/s3-13.1.743. 2

[VHR08] VAN HAM F., ROGOWITZ B.: Perceptual organization in user-generated graph layouts. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1333–1339. doi:10.1109/TVCG.2008.155. 6

[VHW08] VAN HAM F., WATTENBERG M.: Centrality based visualization of small world graphs. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 975–982. doi:10.1111/j.1467-8659.2008.01232.x. 2

[WCXL22] WU J., CHEN X., XU K., LI S.: Structural entropy guided graph hierarchical pooling. In *Proceedings of the 39th International Conference on Machine Learning* (17–23 Jul 2022), Chaudhuri K., Jegelka S., Song L., Szepesvari C., Niu G., Sabato S., (Eds.), vol. 162 of *Proceedings of Machine Learning Research*, PMLR, pp. 24017–24030. URL: https://proceedings.mlr.press/v162/wu22b.html. 1, 2, 3

[WF94] WASSERMAN S., FAUST K.: *Social network analysis: Methods and applications*. Cambridge university press, 1994. 1

[WS98] WATTS D. J., STROGATZ S. H.: Collective dynamics of 'small-world' networks. *nature 393*, 6684 (1998), 440–442. doi:10.1038/30918. 1

[WWW*24] WANG S., WANG P., WU B., ZHU Y., LUO W., PAN Y.: Structural entropy minimization combining graph representation for money laundering identification. *International Journal of Machine Learning and Cybernetics 15*, 9 (2024), 3951–3968. 2

[XWZ*23] XUE M., WANG Z., ZHONG F., WANG Y., XU M., DEUSSEN O., WANG Y.: Taurus: Towards a unified force representation and universal solver for graph layout. *IEEE Transactions on Visualization and Computer Graphics 29*, 01 (2023), 886–895. doi:10.1109/TVCG.2022.3209371. 2

[YZW*23] YANG Z., ZHANG G., WU J., YANG J., SHENG Q. Z., PENG H., LI A., XUE S., SU J.: Minimum entropy principle guided graph neural networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2023), WSDM '23, Association for Computing Machinery, p. 114–122. doi:10.1145/3539597.3570467. 2

[Zac77] ZACHARY W. W.: An information flow model for conflict and fission in small groups. *Journal of anthropological research 33*, 4 (1977), 452–473. doi:10.1086/jar.33.4.3629752. 3, 4, 5

[ZMT10] ZHOU F., MALHER S., TOIVONEN H.: Network simplification with minimal loss of connectivity. In *2010 IEEE international conference on data mining* (2010), IEEE, pp. 659–668. doi:10.1109/ICDM.2010.133. 1, 2

[ZPL23] ZENG X., PENG H., LI A.: Effective and stable role-based multi-agent collaboration by structural information principles. In *Proceedings of 37th AAAI Conference on Artificial Intelligence* (2023). doi:10.1609/aaai.v37i10.26390. 2

[ZSM09] ZAIDI F., SALLABERRY A., MELANÇON G.: Revealing hidden community structures and identifying bridges in complex networks: An application to analyzing contents of web pages for browsing. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology* (2009), vol. 1, IEEE, pp. 198–205. doi:10.1109/WI-IAT.2009.36. 1, 2

[ZXZ*24] ZHONG F., XUE M., ZHANG J., ZHANG F., BAN R., DEUSSEN O., WANG Y.: Force-directed graph layouts revisited: a new force based on the t-distribution. *IEEE Transactions on Visualization and Computer Graphics 30*, 7 (2024), 3650–3663. doi:10.1109/TVCG.2023.3238821. 2, 5