

A Visual Analytics Framework for Contrastive Network Analysis

Takanori Fujiwara*
University of California, Davis

Jian Zhao†
University of Waterloo

Francine Chen‡
FX Palo Alto Laboratory

Kwan-Liu Ma§
University of California, Davis

ABSTRACT

A common network analysis task is comparison of two networks to identify unique characteristics in one network with respect to the other. For example, when comparing protein interaction networks derived from normal and cancer tissues, one essential task is to discover protein-protein interactions unique to cancer tissues. However, this task is challenging when the networks contain complex structural (and semantic) relations. To address this problem, we design ContraNA, a visual analytics framework leveraging both the power of machine learning for uncovering unique characteristics in networks and also the effectiveness of visualization for understanding such uniqueness. The basis of ContraNA is cNRL, which integrates two machine learning schemes, network representation learning (NRL) and contrastive learning (CL), to generate a low-dimensional embedding that reveals the uniqueness of one network when compared to another. ContraNA provides an interactive visualization interface to help analyze the uniqueness by relating embedding results and network structures as well as explaining the learned features by cNRL. We demonstrate the usefulness of ContraNA with two case studies using real-world datasets. We also evaluate ContraNA through a controlled user study with 12 participants on network comparison tasks. The results show that participants were able to both effectively identify unique characteristics from complex networks and interpret the results obtained from cNRL.

Keywords: Contrastive learning, network representation learning, interpretability, network comparison, visual analytics.

1 INTRODUCTION

A network is a common form for modeling various types of relationships in real-world applications, such as social connections [14, 22], biological interactions [18, 39], and supercomputer communications [15]. In practice, comparative analysis of two networks is vital [28, 85], especially for the identification of the uniqueness of one network compared to another. We call this task *contrastive network analysis*. For example, when studying the effect of Alzheimer’s disease on a human brain [37], neuroscientists want to find unique functional connections in the brain network of a patient with Alzheimer’s disease by comparing to that of a healthy subject. Also, for researcher collaborations in different disciplines [62], analysts in a funding agency may want to reveal unique ways of collaboration in the disciplines for decision making.

Despite the demands for network comparison, there is little adequate visual analytics support. Most of the existing methods (e.g., [4, 55, 81]) presuppose the existence of node-correspondence (i.e., pairwise correspondence between nodes in two different networks) [85]. This is a critical limitation since we usually do not know such information in advance when the networks are collected from different resources. One potential solution is identifying the node-correspondence by using network alignment (or graph matching) [28, 85]. However, these algorithms notoriously have high computational costs [28, 85], and thus are only suitable for treating small networks (e.g., 100 nodes). Also, there may not exist a clear correspondence between nodes.

Another approach for visual comparison of networks is based on statistical measures (e.g., network density) [31], centralities (e.g., degree centrality) [92], graphlets [61], or a combination of these [89]. For example, with graphlets [74] (small, connected, and non-isomorphic subgraph patterns in a network), the similarities of two networks can be measured by comparing the frequency of appearance of each graphlet in each network [61]. While these approaches can provide a (dis)similarity between different networks, they compare networks only based on simple measures, which are often insufficient. Also, they only provide network-level similarities, and thus cannot compare networks at more detailed levels (e.g., a node-level). Without a detailed-level comparison, it is difficult to find which part of a network relates to its uniqueness.

To address the above problems, we introduce a novel visual analytics framework, *ContraNA*, for comparative network analysis, which integrates *contrastive network representation learning (cNRL)* [36] into interactive visualization. Empowered by cNRL, our framework allows for discovering unique characteristics of one network by contrasting with another in a comprehensive (i.e., using multiple advanced measures) and detailed (i.e., analyzing a node or subnetwork level) manner without node-correspondence information. Specifically, we employ an interpretable version of cNRL (i-cNRL) [36] to provide human-understandable explanations of discovered characteristics that are further revealed by novel visual representations. We enhance i-cNRL by designing an interactive visual interface that allows analysts to integrate their domain knowledge into the automated analysis. Particularly, we introduce a method to visually identify the uniqueness in one network based on the i-cNRL result, a visual summary to intuitively inform network features that highly contribute to the result, and interactive linkings with the existing network visualizations to explain and refine the result.

In summary, our main contributions include:

- A cNRL-based visual analytics framework, ContraNA, which aims to support a new network analysis approach, named contrastive network analysis, to effectively reveal unique characteristics in one network relative to another.
- Enhancements of i-cNRL with a visual interface that provides four major abilities—**DIIF**: (1) **D**iscovery of uniqueness in networks, (2) **I**nterpretability of features generated by i-cNRL, (3) **I**ntuitive analysis with common visualizations, and (4) **F**lexibility of adjusting i-cNRL based on analysts’ interests.
- Two case studies and a controlled user study with multiple real-world datasets, which assess the effectiveness and usefulness of ContraNA for contrastive network analysis.

2 BACKGROUND AND RELATED WORK

In this section, we first describe network representation learning and contrastive learning—two machine learning schemes used in ContraNA. Then, we review the relevant works for visual network comparison.

2.1 Network Representation and Contrastive Learning

Network representation learning (NRL) [17, 95], also known as graph embedding, aims to learn low-dimensional latent vectors that represent a network while maximally preserving certain network information, such as the structural and semantic characteristics. Once a low-dimensional representation obtained, we can easily and efficiently conduct network analysis tasks (e.g., node classification

*e-mail: tfujiwara@ucdavis.edu

†e-mail: jianzhao@uwaterloo.ca

‡e-mail: francine@acm.org

§e-mail: klma@ucdavis.edu

and link prediction). Typical NRL methods include node2vec [44], DeepGL [77], and some other deep neural networks [96]. More comprehensive descriptions of NRL methods have been included in several recent surveys [17, 95].

Contrastive learning (CL) [99] focuses on finding patterns that are more salient in one dataset relative to another [2]. This is unlike discriminant analysis (e.g., linear discriminant analysis [52]), which aims to discriminate data points based on their classes. Several CL methods have been developed in the machine learning community, such as contrastive versions of latent Dirichlet allocation [99], hidden Markov models [99], and regressions [38]. CL methods for representation learning have been also introduced [2, 3, 27, 38, 80], such as contrastive PCA (cPCA) [2, 38] and contrastive variational autoencoder (cVAE) [3, 80].

Recently, contrastive network representation learning (cNRL) [36] integrates the above two machine learning schemes to achieve comparative network analysis. It uses NRL to generate two sets of latent vectors for two networks and then employs CL to perform comparative analysis based on the vectors. This approach embeds network nodes into a low-dimensional space that reveals the uniqueness of one network compared to another. To offer interpretability, ContraNA uses a specific cNRL method (see Sect. 3) and further provides novel visual analysis capabilities to enable effective network comparison.

2.2 Visualization for Network Comparison

There exist three general approaches in visual comparison: juxtaposition, superposition, and explicit encoding [41]. Through a comprehensive survey, Gleicher [40] provided a framework of considerations for visual comparison, such as tasks, challenges, strategies, and designs. Here we review the relevant works in visual network comparison.

2.2.1 Static Network Comparison

Comparing multiple static networks has been a classic problem in visualization research. Alper et al. [4] presented several superposition designs for node-link and adjacency matrix visualizations to support weighted network comparison. TileMatrix [68] uses juxtaposition to place the triangular adjacency matrices of two networks onto upper and lower areas of a square matrix. On the other hand, John et al. [53] juxtaposed each pair of weighted links in a matrix cell. MatrixWave further extended this approach to support the comparison of multi-layer networks [98].

Researchers have focused on developing techniques for comparing brain networks due to its special characteristics (e.g., very dense) and importance. Shi et al. [81] opted to visualize links that are significantly different between two brain networks. Yang et al. [91] used a clustering algorithm with NodeTrix [50], a hybrid of node-link and adjacency matrix representations. Fujiwara et al. [32] enabled the comparison of a larger number of brain networks by providing an overview with dimensionality reduction. Some other domains have been addressed as well, such as genome interaction [55] and egocentric networks [65].

All the above methods require the information of exact node-correspondence [28], unlike ContraNA. While a few works [6, 59, 67] applied network alignment [28] to find node-correspondence before visualization, they do not scale well due to the computation cost.

2.2.2 Dynamic Network Comparison

Dynamic networks contain nodes and/or links changing over time. A comprehensive survey is provided by Beck et al. [11]. Here, we focus on the comparison of networks at different timestamps.

One approach is based on the juxtaposition of networks at different timestamps. Federico et al. [29] applied a 2D network layout that produces stable node positions across time and then juxtaposed networks at multiple time points in a 2.5D view. On the other hand,

TimeArcs [25] lays out a network at each time point in 1D, and uses an arc diagram to display links. A wall-size display was used to juxtapose an array of networks [63]. Moreover, animated transitions have been employed, which can be viewed as juxtaposition in the temporal domain, e.g., GraphDiaries [8] and DiffAni [78].

Moreover, several works summarize a dynamic network based on the similarity of the network at each timestamp. For example, Small MultiPiles [7] groups similar weighted adjacency matrices across consecutive time points and then shows a representative matrix for each group. EgoLines [97] effectively visualizes a k -hop dynamic egocentric network with a “subway map” metaphor. van den Elzen et al. [88] utilized dimensionality reduction to overview the similarities of networks across time. A similar approach was used to visualize dynamic brain networks [9] and compare dominance variation in animal groups [19].

Lately, researchers have started to utilize time-series or topological analysis to summarize or identify important trends in a dynamic network. Examples include using graph wavelet transform to classify nodes [24] and persistent homology to capture topological changes [45]. Fujiwara et al. [35] applied change point detection [5] to segment a dynamic network and generate summaries. Several works extended this approach in other cases, such as visualizing streaming networks [56, 72].

Again, the above methods still require the information of node-correspondence. To overcome this limitation, we utilize NRL to capture the network’s topological and semantic features.

2.2.3 Comparison without Node-Correspondence

Several systems were developed to support network comparison without the limitation of knowing node-correspondence. ManyNets [31] uses a tabular interface to list several basic network statistics (e.g., degree centrality) for each network. von Landesberger et al. [89] used graphlet frequencies and other network-statistics measures and to generate a self-organization map for arranging networks on a 2D grid. A similar approach was used by Harrigan et al. [47] to visualize egocentric networks, and by Kwon et al. [61] to show similar networks given an input network. In addition to node-level features, Gove [43] suggested network-level features (e.g., density) that are easier to interpret and faster to compute. Along this line, Graph Thumbnails [92] uses the k -core number in a nested circle packing representation of networks.

While the above methods can be used for comparing networks without node-correspondence, they lack the ability to compare networks from multiple levels. ContraNA addresses this by employing the state-of-the-art NRL method, allowing for comparison at both node and subgraph levels. Further, by leveraging CL, we focus on revealing the uniqueness in one network relative to another, which is different from the purpose of the above works (i.e., identifying similarities of networks). To the best of our knowledge, the only work using CL for visual analytics [34] focuses on high-dimensional data. However, ContraNA focuses on comparative analysis of networks.

In sum, the existing methods have limited flexibility in use due to the requirement of node-correspondence or to insufficient analysis ability due to the absence of multiple level comparison. ContraNA addresses these issues by utilizing cNRL, which we describe in Sect. 3. Then, with interactive visualizations, ContraNA further supplements cNRL’s limitations that are identified in Sect. 4.

3 CONTRASTIVE NETWORK REPRESENTATION LEARNING

Here, we provide a brief introduction to the core analysis method used in ContraNA: contrastive network representation learning (cNRL) [36].

3.1 cNRL Architecture

Fig. 1 shows an overview of the cNRL architecture. Given two networks, a target network G_T and a background network G_B , the

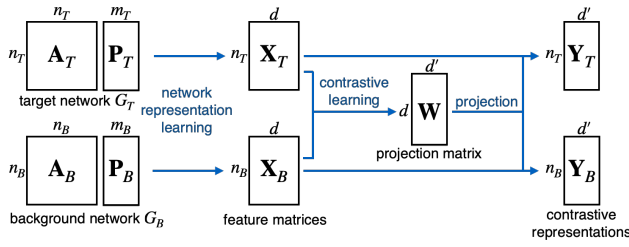


Figure 1: General architecture for cNRL.

objective of cNRL is learning a contrastive representation \mathbf{Y}_T that reveals unique characteristics in G_T relative to G_B . To achieve this, cNRL employs a two-step embedding process: (1) NRL, which obtains network features of G_T and G_B , and (2) CL, which generates a contrastive representation from the network features.

The input networks G_T and G_B can be any combination of undirected or directed, unweighted or weighted, and non-attributed or attributed networks. In their adjacency matrices \mathbf{A}_T and \mathbf{A}_B , the numbers of G_T and G_B nodes, n_T and n_B , do not have to be the same. Similarly, when G_T and G_B are attributed, the numbers of attributes m_T and m_B in matrices of node attributes \mathbf{P}_T and \mathbf{P}_B may be different. The first embedding with NRL produces target and background networks' feature matrices \mathbf{X}_T and \mathbf{X}_B , where both \mathbf{X}_T and \mathbf{X}_B need to have the same d features. Using NRL, cNRL preserves the target and background network information in \mathbf{X}_T and \mathbf{X}_B with explicit and comprehensive network features. Based on \mathbf{X}_T and \mathbf{X}_B , the second embedding using CL generates a projection matrix \mathbf{W} of d rows and d' columns ($d' \leq d$) and then G_T and G_B 's contrastive representations \mathbf{Y}_T and \mathbf{Y}_B can be produced by multiplying \mathbf{X}_T and \mathbf{X}_B with \mathbf{W} , respectively. Through CL, the contrastive representation \mathbf{Y}_T captures relationships (e.g., the network structural differences among network nodes) that appear in G_T but do not appear in G_B .

The cNRL architecture provides flexibility in selection of methods for both NRL and CL. For NRL, we can choose any algorithm that can produce the same features across networks, such as *inductive* NRL methods (e.g., GraphSAGE [46] and DeepGL [77]), which learn transferable knowledge from a training network to other networks. For CL, we can choose any method designed for representation learning. See the work about cNRL [36] for further details about available algorithm options for NRL and CL.

3.2 Interpretable cNRL (i-cNRL)

In ContraNA, to interactively examine the identified unique characteristics with human-understandable explanations, we specifically employ an interpretable version of cNRL, called *i-cNRL*, where DeepGL [77] and cPCA [2] are used as the NRL and CL methods, respectively. Providing interpretability is the core feature of ContraNA as it helps the analysts understand the meaning of unique characteristics found in one network and the reason why such uniqueness can be seen in only that network.

NRL with DeepGL. Using DeepGL [77] as NRL, *i-cNRL* generates feature matrices \mathbf{X}_T and \mathbf{X}_B with interpretable network features. The features consist of the *base feature* \mathbf{x} and *relational function* f .

A base feature \mathbf{x} is a measure we can obtain for each node, such as in-, out-degree, degeneracy (k -core numbers) [73], PageRank [73], or an attribute (e.g., gender of a node in a social network).

A relational function f is a combination of *relational feature operators* (RFOs), each of which summarizes base feature values of one-hop neighbors of a node. For example, the operator can be a computation of the *mean*, *sum*, *maximum* base feature values of one-hop neighbors of a node. Also, the neighbors can be either *in-*, *out-*, *total-neighbors*. Together with the summary measure S , the operators are denoted Φ_S^- , Φ_S^+ , and Φ_S , respectively. For example, $\Phi_{\text{mean}}^-(\mathbf{x})$ computes the mean \mathbf{x} of the in-neighbors of a node. Moreover, the RFO can be applied repeatedly. For ex-

ample, $f = (\Phi_{\text{mean}}^+ \circ \Phi_{\text{max}}^-)(\mathbf{x})$ first computes the maximum \mathbf{x} of in-neighbors for each out-neighbor of a node and then produces the mean of these maximum values.

During the learning process, from the user-input base features, RFOs, and the maximum number of hops to be considered, DeepGL evaluates combinations of these inputs and parameters and automatically selects important network features for preserving the topological (and semantic) information. For example, when using in-degree and out-degree as base features, Φ_{mean} and Φ_{sum} as RFOs, and 2 as the maximum number of hops, DeepGL may generate the network features {in-degree, out-degree, Φ_{mean} (in-degree), Φ_{sum} (in-degree), $\Phi_{\text{mean}} \circ \Phi_{\text{mean}}$ (in-degree)}.

CL with cPCA. From the target and background feature matrices \mathbf{X}_T and \mathbf{X}_B , cPCA [2] produces *contrastive principal components* (cPCs), which are analogous to principal components (PCs) in ordinary PCA [54]. cPCs are low-dimensional representative directions where \mathbf{X}_T has high variance but \mathbf{X}_B has low variance. That is, \mathbf{Y}_T , an embedding of \mathbf{X}_T with cPCs, depicts unique characteristics (with the consideration of variance) of a target network G_T relative to a background network G_B .

cPCA requires one hyperparameter α ($0 \leq \alpha \leq \infty$), called a *contrast parameter*. The contrast parameter α controls the trade-off between having high target variance and low background variance in cPCs. When $\alpha = 0$, cPCs only maximize the variance of \mathbf{X}_T , the same as those in classical PCA. As α increases, cPCs place greater emphasis on directions that reduce the variance of \mathbf{X}_B . Because α has a strong impact on the result, researchers have developed semi-automatic [2] and automatic [36] selection of α .

Similar to PCs in PCA, cPCs are represented as linear transforms of d features of \mathbf{X}_T and \mathbf{X}_B . Analogous to PC loadings, cPCA provides *cPC loadings* [34], which indicate how strongly each of the d input features contributes to the corresponding cPC. By examining a list of d learned features via NRL and cPC loadings, we can understand the relationships between the d features and cPCs; and we can also interpret the contrastive representation \mathbf{Y}_T .

4 DESIGN CONSIDERATIONS

The aforementioned i-cNRL can generate a contrastive representation which highlights the uniqueness of a target network. However, to thoroughly understand the uniqueness, we opt to empower the automated analysis with interactive visualization, which can tightly integrate the knowledge and adaptability of human experts with the statistical learning of machines [79, 84]. We comprehensively identify a set of limitations to i-cNRL for contrastive network analysis in depth, which leads to the following design considerations for our visual analytics framework, ContraNA. In general, we aim to amplify the **Discovery**, **Interpretability**, **Intuitiveness**, and **Flexibility** (**DIIF**) in visual contrastive network analysis.

DC1: Support the discovery of whether a target network is unique compared to a background network, and which part of the network relates to the uniqueness. The uniqueness of a target dataset relative to the base is embedded in the contrastive representation \mathbf{Y}_T generated by CL-based representation learning methods, including cNRL. Many previous works attempted to display this data to reveal the uniqueness [2, 3, 27]. However, because \mathbf{Y}_T only contains the information of the target network G_T , reviewing only \mathbf{Y}_T is not sufficient to understand how well the CL method finds uniqueness. Also, it is difficult to identify which data points (i.e., network nodes in our case) highly relate to the found uniqueness. The visual analytics framework should support discovering the uniqueness and the associated nodes by presenting the information in both the target and background networks.

DC2: Enhance the interpretability of the features learned by NRL and the cPCs generated by CL. Investigating the relationships among the network features, cPCs, and the representation \mathbf{Y}_T is important to interpret the uniqueness of G_T . While i-cNRL is designed

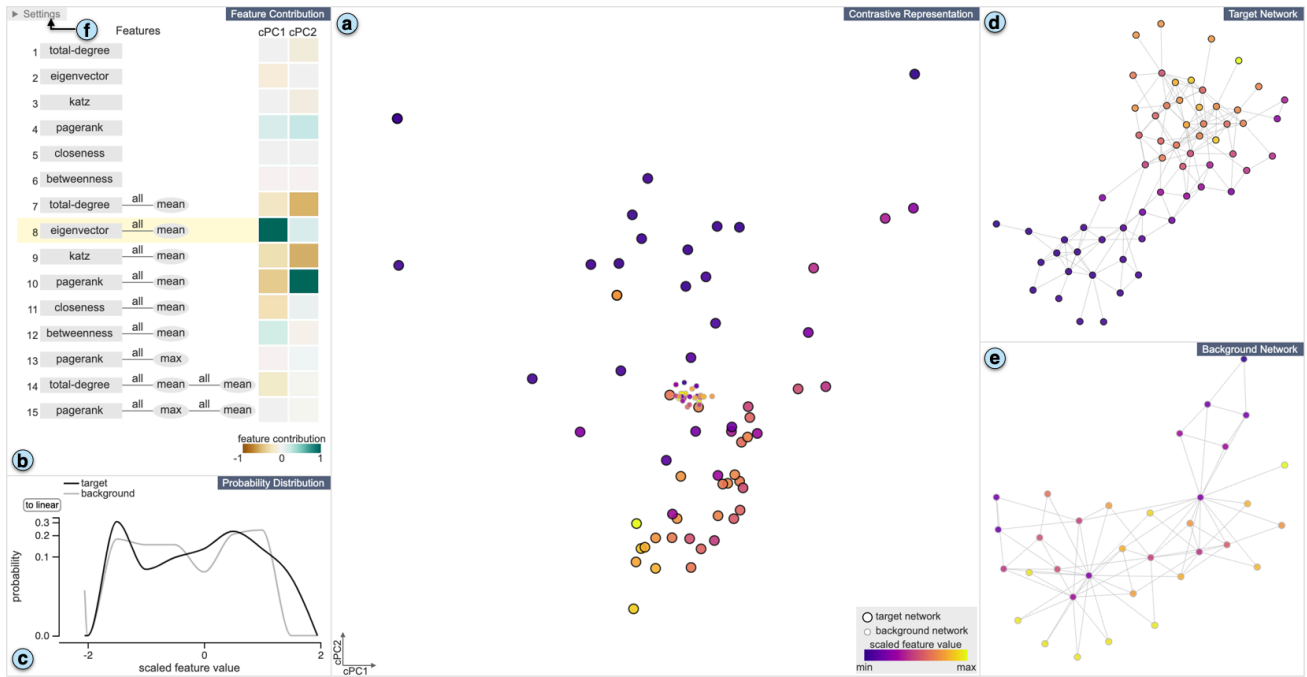


Figure 2: The analyst is using ContraNA to conduct a contrastive analysis of the Dolphin social network [69] (the target network) and the Zachary’s karate club network [94] (the background network). (a) A contrastive representation view shows contrastive representations of target and background networks. (b) A feature contribution view visualizes network features generated by DeepGL and their contributions to each cPC (i.e., scaled cPC loadings). (c) A probability distribution view depicts target and background networks’ probability distributions of the selected network feature in (b). (d)(e) A network layout view draws laid-out target and background networks, respectively. (f) The analyst can change several settings of the algorithm and visualizations from the drop-down menu.

to provide interpretable network features and cPCs, understanding them from i-cNRL’s direct outputs is not straightforward. For example, DeepGL could generate a sophisticated relational function such as $(\Phi_{\text{sum}}^+ \circ \Phi_{\text{max}} \circ \Phi_{\text{mean}}^-)(\mathbf{x})$. Moreover, examining cPC loadings for each feature would be time-consuming when DeepGL produces many network features. The framework should provide visualizations to facilitate easy understanding of the above information.

DC3: Offer intuitiveness in understanding a target network’s uniqueness by relating it to common network visualizations. The contrastive representation \mathbf{Y}_T generated could contain complicated patterns that are difficult to understand. Thus, it is not intuitive enough to just view these patterns directly based on the i-cNRL results in the embedding space. To help analyze such patterns, the framework should provide links between the results of i-cNRL and commonly used visualizations for network analysis, such as laid-out networks and probability distributions of network centralities.

DC4: Provide the flexibility to interactively adjust the i-cNRL parameters to generate results based on the analysts’ interest. The results of i-cNRL heavily depend on the parameters used for each embedding step. For example, changing a value of the contrast parameter α might reveal different unique characteristics in G_T . For analysts with advanced knowledge on NRL and CL, the framework should provide abilities for interactively tuning the i-cNRL results based on their needs.

5 FRAMEWORK OVERVIEW

Grounded by the **DIIF** design considerations, we develop ContraNA which augments the back-end i-cNRL algorithm with interactive visualization (Fig. 2), supporting visual contrastive network analysis.

Fig. 3 shows a workflow of conducting contrastive network analysis with ContraNA. The workflow starts from (A) generation of the i-cNRL results that includes NRL with DeepGL and CL with cPCA (Fig. 1). Afterward, the analyst can first (B) identify whether or not there are any unique characteristics only found in a target network

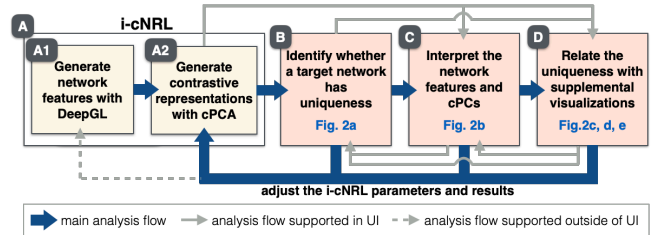


Figure 3: Contrastive network analysis workflow with ContraNA.

from the contrastive representations visualized by ContraNA (Fig. 2-a). If such characteristics exist, to understand the uniqueness, the analyst can (C) interpret the network features and cPCs generated by i-cNRL with visualizations in Fig. 2-b. They can also (D) analyze the contrastive representations, network features, and cPCs by relating them with probability distributions (Fig. 2-c) and laid-out networks (Fig. 2-d, e). Based on findings during the exploration, the analyst might want to adjust the parameters of i-cNRL.

The above procedure is our expected main analysis workflow as indicated by the thick blue arrows in Fig. 3. However, the ContraNA UI provides the flexibility in the analysis activities, shown by the solid gray arrows in Fig. 3. For example, the analyst might want to start to (D) see laid-out networks in order to grasp the topological differences between target and background networks at a glance, and then (B) examine the differences with the contrastive representations. Also, such an interactive analysis often requires to go back and forth between different views to validate findings obtained in one view.

Due to the high computational cost of NRL with DeepGL (e.g., 20 seconds for a network of 6K nodes and 20K links), we decided to support the interactive parameter adjustment only for cPCA. After the analyst updates DeepGL’s parameters and generates the network features, they can analyze the results with the ContraNA UI.

We have developed ContraNA as a web application. For the

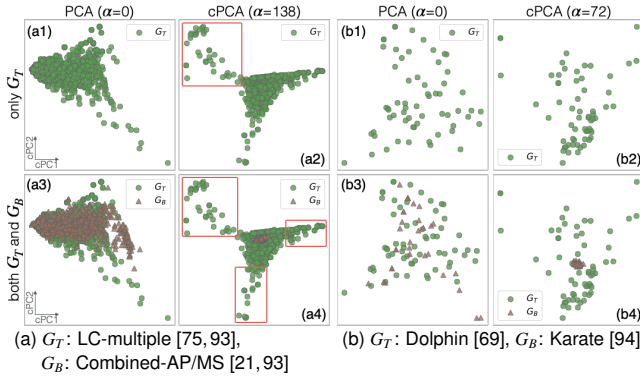


Figure 4: Two-dimensional projections based on (contrastive) representations obtained with PCA (when $\alpha = 0$) and cPCA.

back-end algorithms, we use Python to integrate the existing i-cNRL implementation [36]. The front-end UI is implemented with a combination of HTML5, JavaScript, D3 [16], and WebGL. D3 is used for the feature contribution and probability distribution views (Fig. 2-b, c). For the other views (Fig. 2-a, d, e), we utilize WebGL to support efficient rendering and interaction as networks often consist of many nodes and links (e.g., several thousand nodes). We use WebSocket to communicate between the front- and back-end modules.

6 CONTRANA VISUAL INTERFACE

As shown in Fig. 2, the ContraNA UI consists of four interactively coordinated views, including a contrastive representation view, a feature contribution view, a probability distribution view, and a network layout view, designed with the considerations in Sect. 4. Here, we describe the views provided by the UI through contrastive network analysis of two social networks, the Dolphin social network [69] as G_T and Zachary’s karate club network [94] as G_B . A demonstration video of the interface is available at our online site [1].

6.1 Visualization of Contrastive Representations

With the results generated by i-cNRL, the first step of our analysis workflow (Fig. 3-A), ContraNA’s contrastive representation view (Fig. 2-a) visualizes the results to reveal whether or not there is uniqueness in the target network compared to the background network, serving as the following step (Fig. 3-B, **DC1-Discovery**).

Visual Identification of Target Network’s Uniqueness. Similar to existing works [2, 3, 27, 80], a potential solution is comparing the results of ordinary PCA and cPCA. For example, given the two protein interaction networks, LC-multiple [75, 93] and Combined-AP/MS [21, 93], Fig. 4-a1, a2 show contrastive representations \mathbf{Y}_T generated with i-cNRL using the contrastive parameter $\alpha = 0$ (PCA) and $\alpha = 138$ (cPCA), respectively. In Fig. 4-a2, comparing with Fig. 4-a1, we can see the emergence of a new cluster, as annotated with the red rectangle. It indicates that cPCA successfully finds directions (i.e., cPCs) where G_T has a higher variance than G_B (i.e., the uniqueness). However, in many cases, it is difficult to see clear pattern differences between the results of PCA and cPCA, as shown in Fig. 4-b1, b2 with the networks of dolphins [69] as G_T and Karate club members [94] as G_B .

The problem is mainly because we do not know how nodes in a background dataset distribute in the embedding space generated by CL. Thus, we introduce a method that plots the contrastive representations of target and background datasets, \mathbf{Y}_T and \mathbf{Y}_B , together. As shown in Fig. 4-a3, a4, b3, b4, \mathbf{Y}_T and \mathbf{Y}_B are visualized as green circles and brown triangles, respectively.

When a network has high variance in the embedded space, its nodes are widely distributed along cPCs. Thus, the uniqueness of a target network G_T can be identified by comparing the scatteredness of nodes in \mathbf{Y}_T and \mathbf{Y}_B . As shown in Fig. 4-a4, b4, cPCA reveals

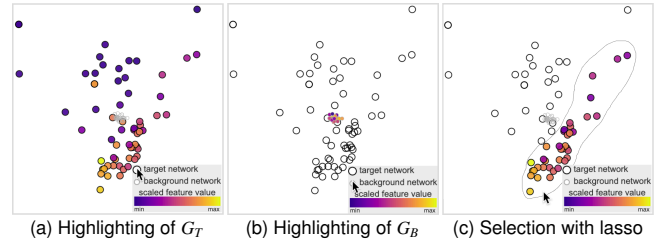


Figure 5: Node highlighting and selection supported in the contrastive representation view.

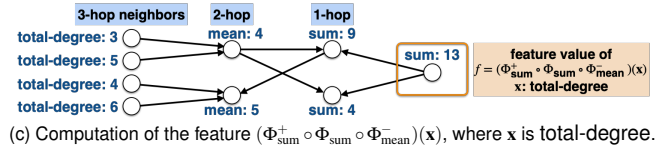
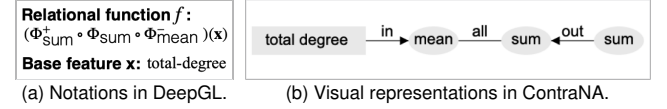


Figure 6: Representations of network features in DeepGL and ContraNA. Here, as an example, we use a complex feature (consisting of three relational feature operators) that DeepGL may produce. (a) and (b) represent the same feature: *the sum of out-neighbors of the sum of all-neighbors of the mean of in-neighbors of total-degrees*. (c) shows an example of the computational flow of this feature. In (c), the circles and arrows represent nodes and directed links of a network.

that \mathbf{Y}_T has much higher scatteredness than \mathbf{Y}_B . Moreover, we can easily grasp which parts of a target network have strong uniqueness. Similar to other representation methods (e.g., PCA and MDS [86]), a distance in the embedding space of cPCA represents a dissimilarity between nodes. Thus, when the target network nodes are highly unique, they are placed far away from the nodes in the background network (e.g., the nodes in the red boxes of Fig. 4-a4).

Integration into ContraNA. We employ the above visualization as the contrastive representation view of ContraNA (Fig. 2-a), where the values of a network feature selected in the feature contribution view (see Fig. 2-b and Sect. 6.2) are color-coded with a purple-yellow scheme [82]. To encode nodes in target and background networks, we first explored different shapes, including circle, triangle, and squares; however, circles and squares are hard to distinguish and triangles require much higher rendering cost with WebGL than circles and squares. We then used circles with different sizes and borders, with larger and black-border circles for the target network and smaller and gray-border circles for the background network. Moreover, the analyst can highlight the target or the background network by hovering over the corresponding legend as shown in Fig. 5-a, b. The contrastive representation view also provides fundamental interactions, such as zooming, panning, and lasso selection (Fig. 5-c). From the different scatteredness of G_T and G_B nodes in Fig. 2-a, we can decide that there exists uniqueness in the Dolphin network.

6.2 Interpretation of Network Features and cPCs

With the above observation from the contrastive representation view, we move on to interpret the network features and cPCs (Fig. 3-C, **DC2-Interpretability**) with the feature contribution view (Fig. 2-b).

Visual Representation of Network Features. The left part of the feature contribution view lists all the network features generated by DeepGL. They usually consist of a few relational feature operators (RFOs), which are represented with mathematical notations (Fig. 6-a). However, it is difficult for analysts to interpret features with such notations. We thus design an intuitive visual representation

of the features (Fig. 6-b).

A network feature learned by DeepGL consists of the base feature (e.g., total-degree), summary measures (e.g., mean), and neighbor types (e.g., in-neighbors). We use a gray rectangle and an ellipse with text labels to denote a base feature and a summary measure, respectively. Then, we connect them with a line and, to indicate the neighbor type, annotate with a text label (*in*, *out*, or *all*). Also, for in- and out-neighbors, we use an arrowhead to indicate the direction. Lastly, we order them from left to right based on the computational flow to obtain the feature value. The resultant representation in Fig. 6-b visually summarizes the neighborhood relationships and the computational flow, which is further explained in Fig. 6-c.

Visualization of cPC Loadings. The right part of the feature contribution view visualizes cPC loadings [34] described in Sect. 3.2 as a heatmap. Each row and column correspond to a network feature and cPC, respectively. Similar to Fujiwara et al.’s work [34], we generate scaled cPC loadings (or *feature contributions*) between $[-1, 1]$ by dividing each cPC’s loadings by their maximum absolute value. Then, we encode the scaled cPC loadings with a brown-to-blue-green diverging colormap [26, 48]. The magnitude of the loading represents how strongly a feature contributes to the corresponding cPC. For example, the feature at the eighth row in Fig. 2-b (F8: the mean of all-neighbors’ eigenvector centralities [73]), has the most influence on cPC1. Also, the sign of the loading indicates the contributed direction along the cPC (+: positive; -: negative). For example, in Fig. 2-a where each node is colored by F8, we can see that the feature values of G_T generally vary from low to high along the positive x -direction.

By default, ContraNA automatically selects the feature that most strongly contributes to cPC1 (e.g., F8 in Fig. 2-b) and highlights the corresponding row in yellow. The analyst can select a different feature, and all other views are updated based on the selected feature (e.g., node colors in the contrastive representation view).

By using the contrastive representation and feature contribution views together, we discover that the uniqueness of the Dolphin network G_T highly relates to F8. From the nodes colored by the feature values (Fig. 2-a), we can see that the nodes around the top-left have low values while the nodes around the bottom-right tend to have higher values.

6.3 Relating to Common Network Visualizations

With above results, we further analyze the uniqueness by relating F8 to common network visualizations (Fig. 3-D). ContraNA provides two perspectives for network analysis (**DC3-Intuitiveness**): probability distributions and laid-out networks [10]. Probability distributions are often used to compare the distributions of target and background networks’ centralities (e.g., whether the degree distribution follows the power law [10]), and laid-out networks are helpful for viewing the topological differences (e.g., whether multiple communities exist).

Linking with Probability Distributions. The probability distribution view (Fig. 2-c) shows the distributions of the selected feature values in the feature contribution view (i.e., F8 in Fig. 2-b), for target and background networks. Its x - and y -coordinates represent a (scaled) feature value and its probability (or relative frequency), respectively. Both logarithmic and linear scales for the y -coordinate are supported. We colorcode the probability distribution lines with the same colors used for the node borders in the contrastive representation view (i.e., black: target network, gray: background network).

Linking with Network Layouts. The network layout view in Fig. 2-d, e visualizes laid-out target and background networks, with the scalable force-directed placement [51]. Same as the contrastive representation view, each node is colored based on the selected feature in the feature contribution view (e.g., F8 in Fig. 2-b) and outlined in black (target network) or gray (background network). The network layout view also supports several basic interactions

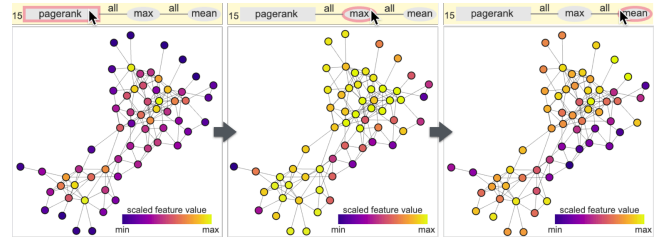


Figure 7: Visualization of intermediate computational results of feature F15.

such as zooming, panning, and lasso selection, and is fully linked with other views. For example, by reviewing Fig. 2-a, b, d together, we notice that the two node groups found previously (i.e., nodes with small and high F8 values, placed around the top-left and bottom-right in Fig. 2-a) seem to correspond to distinct communities at the bottom-left and top-right in Fig. 2-d. This can be confirmed by performing a lasso selection on the nodes in Fig. 2-a, as demonstrated in Fig. 5-c.

Understanding Complicated Network Features. The linkings above can be utilized to further help understand the network feature that consists of multiple RFOs. As shown in Fig. 7, by hovering over either the base feature or summary measure in the feature contribution view, the network layout view and the contrastive representation view show the intermediate computational results of the feature values. For instance, Fig. 7 (from left to right) visualizes PageRank values of G_T ’s nodes, the maximum of all-neighbors of PageRank values, and the mean of all-neighbors of them. Thus, the analyst can visually understand how the base feature values spread across the neighbors and how the final network feature values are derived.

Through the analysis from Fig. 3-A to D, we can conclude that the Dolphin network G_T has unique characteristics relative to the Karate network G_B . The uniqueness highly relates to F8: eigenvector centralities of each node’s neighbors, and it clearly reveals the separation of the two communities in G_T , which cannot be seen in G_B .

6.4 Refinement of Contrastive Representations

The cPCA used in i-cNRL automatically selects the contrastive parameter α and computes cPCs to generate the optimized contrastive representations, i.e., maximizing the variation in \mathbf{X}_T while simultaneously minimizing the variation in \mathbf{X}_B [36] (Fig. 1). However, the analyst may want to loosen or strengthen the reduction of the variation of \mathbf{X}_B in order to elucidate the found patterns or discover different patterns. For example, around the top-left in Fig. 2-a, an orange node, with a high value of F8, is mixed up with the nodes with lower values (as annotated in the green box in Fig. 8-b). Also, the resultant cPCs might not apt to interpret visually found patterns. For example, in Fig. 2-a, the value of F8 tends to increase along the diagonal line, but not along cPC1 (the x -axis). To handle such cases, ContraNA supports interactive adjustments of α and cPCs (**DC4-Flexibility**).

Adjustment of Contrastive Parameter. ContraNA allows the analyst to interactively change the contrastive parameter α with a range slider (Fig. 2-f), based on the efficiency of cPCA (e.g., the completion time is less than 3ms for 10,000 nodes with 10 features [34]). However, the update of α in cPCA causes an arbitrary sign flipping for each cPC, similar to PCA [33, 87]. Fig. 8-a shows an example of the flipping along both horizontal and vertical directions when α is changed, making it difficult to follow.

To address this, we employ a similar solution used for PCA [87]. For each of cPC1 and cPC2, we compute the cosine similarity between the coordinates of all nodes before and after the update; then if the similarity is negative, we flip the sign generated by cPCA. Fig. 8-b shows the result with the sign adjustment. As α decreases to 38, the orange node annotated with the green rectangle moves to

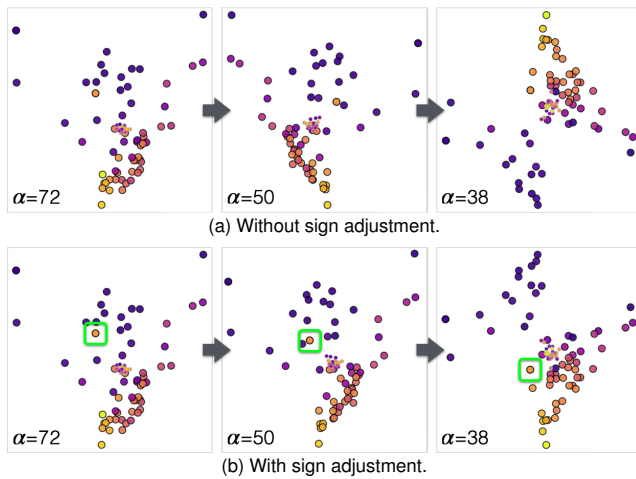


Figure 8: Changes of node projection by updating contrastive parameter α (a) without and (b) with sign adjustment. Corresponding animations are available online [1].

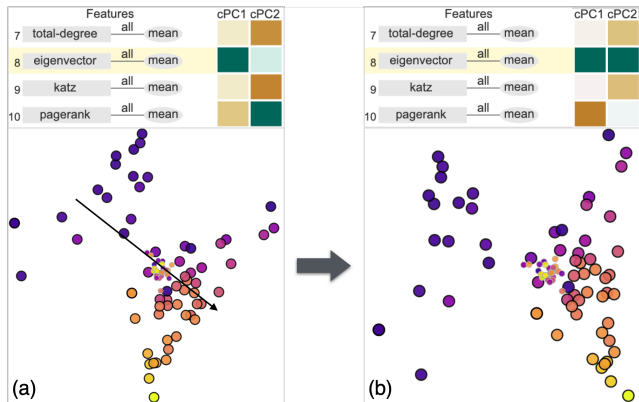


Figure 9: Adjustment of the cPCs. (a) and (b) show feature contributions of Features 7–10 and the contrastive representation view before and after the adjustment, respectively.

ward the right-bottom and the separation of nodes with low (purple) and higher values (pink, orange, and yellow) becomes more salient.

Adjustment of Contrastive Principal Components. We introduce an interactive method for customizing cPCs, which can be used for both PCA and cPCA. First, in the contrastive representation view, a preferable axis direction for cPC1 can be drawn as a straight line (Fig. 9-a). We then rotate the projection based on the angle between the drawn line and cPC1 (Fig. 9-b). As a result, we also need to update the cPC loadings shown in Fig. 2-b. Similar to the rotation in ordinary PCA [76], the cPC loadings can be obtained by simply multiplying a rotation matrix with the above user-defined angle. For example, Fig. 9-a, b show a subset of the cPC loadings corresponding to F7-10 before and after the rotation. We can see that F8 has a strong contribution to both cPC1 and cPC2 and F10 has a stronger contribution to cPC1 than before.

Note that we can also use a method developed by Kwon et al. [60] for general scatterplots, including cPCA projection results. It generates new axes based on the user-drawn freeform line over the plot and nonlinear transformation. However, we use the above linear transformation, so that we can update cPC loadings, which are important to interpret the result of cPCA.

7 CASE STUDIES

In Sect. 6, we have shown the effectiveness of ContraNA through an example of comparing two social networks. Here, we demonstrate

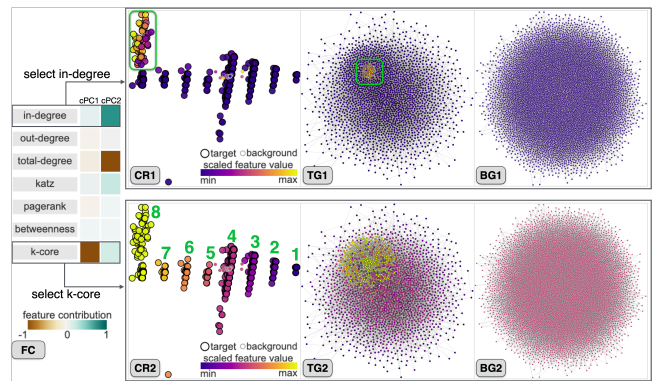


Figure 10: Case study 1. (FC) shows a subset of network features visualized in the feature contribution view (note: a full set of them includes a feature consisting of multiple RFOs). (CR1, TG1, BG1) show the contrastive representation, target network layout, background network layout views after selecting in-degree, respectively. Similarly, (CR2, TG2, BG2) are the results after selecting k -core. As a default, ContraNA selects k -core as the most contributed feature to cPC1.

two additional case studies, including an evaluation of a network model and a comparison of protein interaction networks.

7.1 Study 1: Evaluation of a Network Model

Network modeling is essential to simulate and understand real-world networks (e.g., how they grow and shrink). It also can be used to perform what-if analysis (e.g., what elimination of a hub node will cause), as well as to generate synthetic datasets [42]. ContraNA can help evaluate network models by comparing them with real-world networks. In this case study, we focus on peer-to-peer (P2P) networks, where a precise network model is essential for analyzing the robustness of a P2P network [66]. P2P networks are often scale-free [66]; thus, we use the Price’s model [73] as an evaluation target. As listed in Table 1 (see Sect. 8), to identify which characteristics the Price’s model does not simulate well, we set a real-world P2P network (p2p-Gnutella08) as G_T , and the network generated with the Price’s model (Price 2) as G_B .

As shown in Fig. 10-CR2, the contrastive representation view indicates the differences in the node distributions of G_T and G_B . The P2P network (G_T) has clear groups of nodes, unlike Price 2 (G_B). From Fig. 10-FC, in-degree, total-degree, and k -core are identified as main contribution features. After selecting in-degree in Fig. 10-FC to review the related information with the other views (Fig. 10-CR1, TG1, BG1), we notice that G_T has a region where nodes have a much higher in-degree than the other nodes, as annotated with the green boxes in Fig. 10-CR1, TG1. Similar findings appear when the total-degree is selected in Fig. 10-FC. As for the k -core number (Fig. 10-CR2, TG2, BG2), there is no obvious difference of this feature in G_B , but a clear distinction with 8 groups of nodes in G_T , as annotated in Fig. 10-CR2. Moreover, from Fig. 10-FC, we can see that cPC1 is more related to the k -core values and cPC2 is more related to the degree of nodes. Therefore, unique characteristics in the P2P network have been identified.

Also, the result above reveals more variations of the k -core number in the P2P network G_T . The k -core number informs that a node at least connects to other k nodes [73], indicating that the Price’s model presents a significant difference in the network robustness from the P2P network. This issue arises because the Price’s model forms a network by always adding a new node with a fixed number of links; as a result, the generated nodes have a constant k -core number. Therefore, to better simulate the P2P network, we should develop or use a model that can generate multiple k -core numbers, such as the dual-Barabási-Albert model [71] or its extension [36].

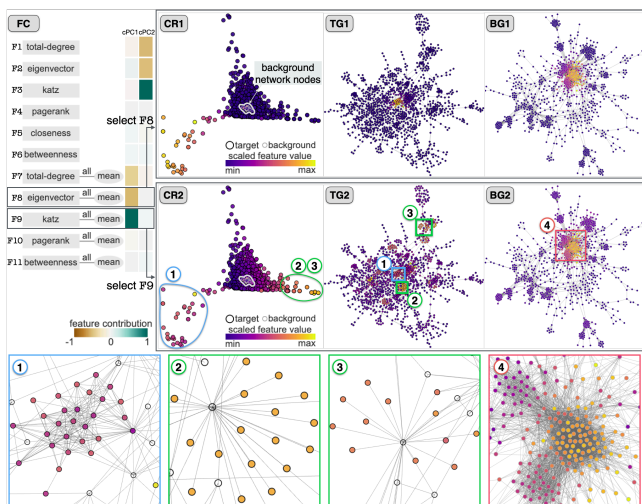


Figure 11: Case study 2. (FC) shows a subset of features in the feature contribution view. (CR1, TG1, BG1) show the contrastive representation, target network layout, background network layout views after selecting F8. Similarly, (CR2, TG2, BG2) are the results after selecting F9. ①–④ at the bottom show the zoomed regions from TG2 and BG2.

7.2 Study 2: Comparison of Interactome Networks

In this case study, we compare two *interactome* networks, LC-multiple and Combined-AP/MS [93], which represent protein-protein interactions of the yeast *S. cerevisiae*. While LC-multiple is the literature-curated (LC) interactome from several *low-throughput* experiments [75, 93], Combined-AP/MS is generated through a *high-throughput* approach, specifically, affinity purification/mass spectrometry (AP/MS) [21, 93]. Low and high-throughput approaches have different pros and cons in capturing the protein interactions [90, 93], and thus they produce different interactomes. A comparison of these interactomes is fundamental to assess the quality and characteristics of each approach [93]. As listed in Table 1, we set LC-multiple and Combined-AP/MS as G_T and G_B , respectively.

As shown in Fig. 11-CR2, TG2, BG2, ContraNA automatically selects F9 (because of its high contribution) and generates related visualizations. A difference is revealed in the scatteredness of the target and background networks in Fig. 11-CR2. We also notice that LC-multiple G_T has relatively high feature values towards both left and right directions of cPC1, as annotated with the light blue and green. This indicates that cPC1 is not dominantly decided by F9. Therefore, we select the secondary contributed feature F8, and the results are shown in Fig. 11-CR1, TG1, BG1. From Fig. 11-CR1, we can see that only the area annotated light blue in Fig. 11-CR2 has high values of F8. F8 and F9 are related to the eigenvector and Katz centralities. Both centralities measure how strongly a node influences other nodes; however, the eigenvector centrality tends to be high only when a node is in a strongly connected region while the Katz centrality can be high even when a node is in a weakly connected region [73]. Thus, we can expect that the nodes annotated with the light blue and green in Fig. 11-CR2 are in strongly and weakly connected regions, respectively.

To visually confirm the above patterns, we select the annotated nodes from Fig. 11-CR2 and zoom into the related regions in Fig. 11-TG2, as detailed by Fig. 11-①, ②, ③. Here we show only two from all the regions related to the nodes annotated with the green color in Fig. 11-CR2, TG2. Similarly, in Fig. 11-④, we show the region where the nodes have high F9 values in Fig. 11-BG2. We can see that the nodes in Fig. 11-①, ④ are strongly connected, but not in Fig. 11-②, ③. From these observations, we can confirm that the uniqueness is derived from the fact that G_T has two different types

of nodes linked to high Katz centrality node(s) in either strongly or weakly connected region, which cannot be seen in G_B . This finding indicates that using LC-multiple or Combined-AP/MS to identify important proteins for *S. cerevisiae* could reach different conclusions. Therefore, additional validation would be needed before deciding their importance based on only one dataset.

8 CONTROLLED USER STUDY

In addition to the case studies, we conducted a controlled user study to assess the usefulness of ContraNA for contrastive network analysis. We aimed to answer these research questions: **(Q1)** Can analysts effectively identify unique characteristics in a target network (compared to a background network), and **(Q2)** Can analysts properly interpret and explain the found uniqueness? We expected that **Q1** would be primarily addressed by the contrastive representation view, and that all the other coordinated views would help answer **Q2**. We provide the materials used for the study online [1], including the datasets listed in Table 1, their visualized results with ContraNA, and questionnaires.

8.1 Study Design

As far as we know, ContraNA is the first framework designed for contrastive network analysis, and thus we were not able to find a baseline system to compare against. Therefore, we design the following study to evaluate the usability of ContraNA in terms of discovering a target network’s uniqueness and interpreting it.

Datasets. As shown in Table 1, we generated random networks (Random 1, 2) with Gilbert’s random graph [10] and scale-free networks (Price 1, 2) with the Price’s preferential attachment models [73], as well as used several public datasets. We categorized the analysis tasks into three by carefully selecting target and background networks: (a) no uniqueness is in G_T (# of RFOs is N/A), (b) the uniqueness in G_T can be identified and interpreted with a network feature containing only the base feature (# of RFOs = 0), and (c) containing RFOs (# of RFOs ≥ 1). As the number of RFOs increases, a feature becomes more complicated and the task becomes harder.

Participants. We recruited 12 participants (4 females and 8 males; aged 18–44) at a local university, with 10 from computer science and 2 from political science. There were 1 postdoc-fellow, 10 PhDs, and 1 Master’s. We pre-screened participants to ensure that they have fundamental knowledge of network science. Their self-reported familiarity with network analysis had the median of 5 (■■■■■), on a scale of 1 (not familiar) to 7 (use regularly). Out of 7 network centralities/measures used in the study (i.e., degree, closeness, betweenness, eigenvector, Katz centralities, PageRank, and k -core number [73]), participants’ knowledge of these had the median of 3 (■■■■■).

Apparatus. The study was conducted on an iMac (4 GHz Intel Core i7, 16GB 1,600 MHz DDR3) with a 27-inch display (5,120 \times 2,880 pixels), connected with an Apple Magic Mouse 2. The UI was presented with Google Chrome in full-screen mode. Because the refinement of contrastive representations (Sect. 6.4) was not relevant to our study tasks, we disabled the related functionalities.

Tasks and Design. Based on **Q1** and **Q2**, given target and background networks, participants were asked to perform comparative analysis using ContraNA and complete two subtasks, **(ST1)** and **(ST2)**: **(ST1)** identifies whether or not the target network has any uniqueness compared to the background network, and **(ST2)** explains the found uniqueness (if any) or the reason of concluding there is no uniqueness. ST1 required a selection from options of *Yes*, *No*, and *I’m not sure*; for ST2, participants were asked to write down their explanation. We employed a within-subjects design for our study. Each participant completed three comparative network analysis tasks in our main study, using three different pairs of networks (Tasks A, B, and C in Table 1). The order of tasks was counterbalanced across participants.

Table 1: Networks used for the controlled user study and case studies, where n and l represent the numbers of nodes and links, respectively.

	Target Network G_T	Background Network G_B	# of RFOs*
Tutorial 1	Price1 ($n = 100, l = 294$)	Random1 ($n = 100, l = 471$)	0
Tutorial 2	Random1 ($n = 100, l = 471$)	Random2 ($n = 100, l = 525$)	N/A
Tutorial 3,	LC-multiple [93]	Combined-AP/MS [93]	1
Case Study 2	($n = 1,536, l = 2,925$)	($n = 1,622, l = 9,070$)	
Training	School-Day2 [83] ($n = 238, l = 5,539$)	School-Day1 [83] ($n = 236, l = 5,899$)	2
Task A	Brain-Low [57] ($n = 233, l = 2,627$)	Brain-High [57] ($n = 246, l = 3,355$)	N/A
Task B,	p2p-Gnutella08 [64]	Price2	0
Case Study 1	($n = 6,301, l = 20,777$)	($n = 6,301, l = 18,897$)	
Task C	Dolphin [69] ($n = 62, l = 159$)	Karate [94] ($n = 34, l = 78$)	1

* # of RFOs in a network feature that highly contributes to the uniqueness in G_T .

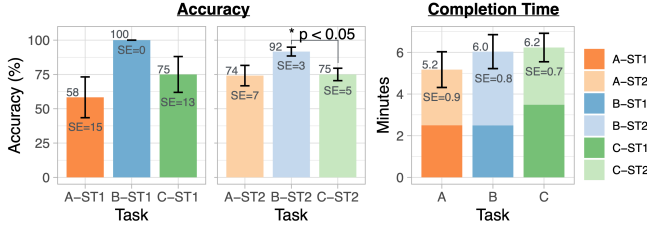


Figure 12: Accuracy (left) and completion time (right) for each subtask.

Procedure. At the beginning, participants provided their demographics and backgrounds on a survey. A brief tutorial was then presented including explanations of the definition of the uniqueness, the above 7 network centralities/measures, the usage of ContraNA, and 3 concrete analysis examples. Afterward, participants completed a training session, allowing them to get familiar with ContraNA and the task, followed by the real study consisting of three tasks. The datasets used in the tutorials, training, and study tasks are shown in Table 1. Think aloud protocol was used during the training and task sessions. They were allowed to ask questions about the ContraNA UI and the network centralities and measures. No time limit was set for the tasks. Lastly, participants provided their feedback with the NASA TLX [49], a questionnaire about ContraNA’s visual interface, and a semi-structured interview. The whole study lasted around 1 hour per participant.

8.2 Results

This section reports our controlled study results including task accuracy, completion time, and participants’ subjective feedback.

Accuracy. The accuracy for each subtask is shown in Fig. 12-left. Two network science experts independently rated participants’ explanations in ST2 with a scale of 1 (the worst) to 5 (the best) based on correctness and comprehensiveness. Weighted Cohen’s kappa coefficient indicates high reliability of the ratings ($\kappa = 0.83$, in the range of 0.81–1.00: *almost perfect agreement*) [20].

In general, Task B has the highest mean accuracy for both ST1 (100%) and ST2 (92%), which might be because the uniqueness of the target network can be understood easily with the base feature. However, for ST1, a Cochran’s Q test [70] does not show any significant differences across tasks. For ST2, a Friedman test [70] reveals significant differences ($\chi^2 = 7.55, p < 0.05$). A post-hoc analysis using Wilcoxon signed-rank exact test with Bonferroni correction [13] indicates that Task B has significantly higher accuracy than Task C ($p < 0.05$) that has the most difficulty. Additionally, participants’ scores of ST2 show a weak positive correlation (Pearson’s correlation coefficient $\rho = 0.31$) with the numbers of network centralities/measures they knew, which generally represent their level of expertise in network science. Thus, higher expertise seems to help provide better explanations.

Completion Time. Fig. 12-right shows the completion time for each task. However, a Friedman test does not show any significant

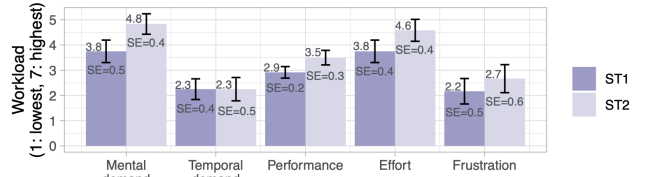


Figure 13: NASA TLX results (the lower the better).

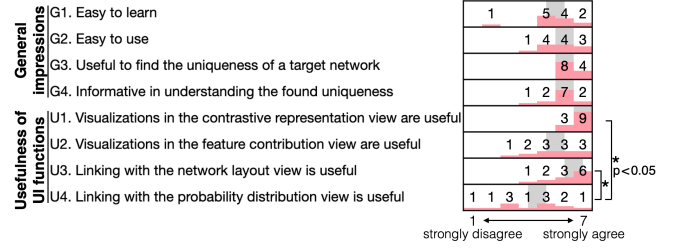


Figure 14: Histograms of participants’ ratings on the overall impression and usefulness of each UI function (the higher the better). Numbers over the bins represent the frequency. Median ratings are indicated in gray.

difference across tasks. There is a weak negative correlation ($\rho = -0.33$) between the completion times and the numbers of known network centralities/measures (i.e., the expertise helped finish tasks faster). For Tasks A and B, ST2 (2.7 minutes and 3.5 minutes, respectively) took longer than ST1 (both 2.5 minutes). But for Task C, it is the opposite (ST1: 3.5 minutes, ST2: 2.7 minutes). From our observation, the reason might be that participants tried to find the explanation (ST2) before deciding their answer to ST1.

Subjective Feedback. Fig. 13 lists participants’ ratings with the NASA TLX. Generally, ST2 has higher mean values than ST1 in each task; however, a Wilcoxon signed-rank exact test does not show any significant difference in each pair of subtasks. Participants expressed relatively high mental demand and effort for performing the tasks, which is plausible because the network analysis needs high concentration. Fig. 14 shows the questionnaire results on the impression of ContraNA. Overall, participants felt that ContraNA is easy to learn, easy to use, and useful to perform ST1 and ST2. For the usefulness of each UI function, the contrastive representation, feature contribution, and network layout views receive high ratings, especially the contrastive representation view, whereas the probability distribution view has relatively low scores. Also, a Friedman test ($\chi^2 = 18.0, p < 0.001$) and a post-hoc analysis using the Wilcoxon signed-rank exact test with Bonferroni correction [13] show significant differences of the probability distribution view from the contrastive representation ($p < 0.05$) and network layout ($p < 0.05$) views on participants’ ratings. One reason we obtained from the interviews is that the uniqueness can be identified and explained with other views, while the probability distribution view is not necessary, although it is helpful to confirm uniqueness.

During the interview, we collected the participants’ preference for the feature representations in DeepGL (Fig. 6-a) and ContraNA (Fig. 6-b). Ten out of 12 participants preferred ContraNA’s representation because it is “visually more clear” (p4, 6) and “more intuitive to understand” (other 8 participants). The rest of the participants preferred DeepGL’s notation because using mathematical symbols has less ambiguity. Eleven participants applauded the usefulness of ContraNA’s visualization of intermediate computational results (Fig. 7), which was used to understand complicated network features: “Those are particularly useful because you can see the levels of how these [i.e., features] are getting computed like that” (p3). Two participants with expert knowledge mentioned that they wanted to use the opposite order from the current representation (i.e., from left to right, placing RFOs first and then a base feature) because they

mentally converted each network feature in this order. However, others stated that they were used to understand each feature from a base feature.

9 DISCUSSION

We have presented ContraNA and validated it with case studies and a controlled user study. Here, we provide a thematic discussion on additional aspects of ContraNA as well as the studies.

Limitations in Visual Scalability. While the studies indicate the usefulness and effectiveness of ContraNA, it is not without limitations. ContraNA employs scatterplots, node-link diagrams, and heatmaps in its interface, but these techniques suffer from scalability issues. We can enhance these techniques with filtering, aggregation, and focus+context methods to mitigate the issues [23]. A specific scalability issue in ContraNA is the visual representation of network features, where we use rectangles in the feature contribution view, ellipses, lines with text labels. This may limit the number of RFOs to display in a network feature. However, this is not a major issue, because NRL methods (including DeepGL) that generate features based on relationships of node neighbors are generally utilized with only a few hops (typically 2 or 3) of neighbors [58, 77]. Another issue could be caused when NRL produces a large number of features (e.g., 100 features). This issue can be addressed by only displaying features that highly contribute to cPCs as such features are most important to interpret the cNRL results.

Ambiguity of Uniqueness. In spite of high mental demand and effort, participants achieved high accuracy in identifying (Q1) and explaining (Q2) the uniqueness in a target network when it actually exists (Tasks B and C). However, when a target network did not have clear uniqueness (Task A), the accuracy for ST1 was relatively low, though there was no significant difference (Sect. 8.2). Potential reasons might be associated with the ambiguity of uniqueness and participants' expectation, as noted by p1: *"It wasn't too difficult [to learn and use the contrastive representation view] but I had a question of how much separation is enough to define uniqueness."* While the contrastive representation view in Task A showed the similar scatteredness between target and background networks, participants were able to find some small regions that seemed to relate to the uniqueness if they had an expectation for uniqueness. We found that all 5 participants who answered *Yes* for Task A-ST1 did not provide a convincing explanation, with mean accuracy for Task A-ST2 52% (Fig. 12). How to better define and inform a threshold of containing the uniqueness should be addressed in the further work.

Importance of Interpretability and cNRL. One notable result is that participants spent similar time in completing ST1 and ST2. This surprises us because we expected that ST1 would be finished much faster because they only needed to review the contrastive representation view and select an answer, while ST2 required the use of multiple views and writing an explanation. From our observation, we noticed that although they quickly recognized the uniqueness from the contrastive representation view, before selecting the answer, they tried to understand the reasons behind to convince themselves. This points out the importance of providing the interpretability in algorithms, including NRL and CL methods. This fact also influenced the mean accuracy of Task C-ST1. Three participants chose *I'm not sure* because they were not able to completely understand why the target network was unique while the potential uniqueness was found, which may be due to their lower expertise in network science.

All the views except for the probability distribution view seemed to be useful according to participants. From our interviews, several participants mentioned that for easier tasks (e.g., Task B) it was not necessary to use the probability distribution view; for more difficult tasks (e.g., Task C), the probability distribution view was not helpful to reveal the uniqueness. This indicates the limitation of network comparison based on probability distributions, which is a popular analysis approach, and the necessity of more advanced

embedding based approaches, such as cNRL. Further, when asked about how to perform similar tasks without ContraNA, participants provided approaches of either comparing probability distributions of basic network centralities or comparing laid-out networks. Also, they mentioned that they might be able to find the uniqueness with their stated approach but it would *"be awful"* (p9) and *"take longer"* (p1), and *"I might miss some uniqueness"* (p5). In contrast, using ContraNA is *"much easier because it supports a lot of stuff you need to deal with... Comparing the target and background in the contrastive representation view is really helpful. If you see spreading patterns [of a target], it might be unique."* (p11).

Usage with Other Algorithms. ContraNA employs i-cNRL because of its interpretability; however, most of ContraNA functionalities are generic enough to be well adapted with other NRL and CL methods in the architecture. For example, if the interpretability is not required, DeepGL can be replaced with GraphSAGE [46], where only network features are changed. Thus, ContraNA is still applicable by updating the visual representations of features in the feature contribution view. Similarly, we can switch cPCA with the other CL methods, such as cVAE [3, 80] which can find uniqueness in a target dataset even when its data points and latent features have non-linear relationships. As we cannot obtain the features' contributions, in this case, we can simply remove the heatmap from the feature contribution view. Also, once other interpretable CL methods are developed, we do not need any changes to integrate them into ContraNA. Another potential extension is cooperating with link feature learning, which is also supported by DeepGL. In this case, we just need to add visual encodings of links to the views of ContraNA.

Adaption for Application Domains. As presented, the linking with laid-out networks is important to intuitively understand uniqueness. Networks are often visualized in a specific manner according to the application domain. For example, when analyzing brain networks, neuroscientists often use adjacency-matrix based visualizations or 2D/3D node-link diagrams [30]. This is because the former is useful to find correlated brain regions with matrix-reordering algorithms [12] and the latter can help relate analysis results to the actual locations in a brain. By customizing the network layout views, ContraNA can support such analysis tasks in this specific domain. Also, as shown in Sect. 8.2, the way to understand network features generated by DeepGL is different by the analyst. Therefore, we should consider adding settings to customize the representation of network features based on the analyst's preference.

10 CONCLUSION AND FUTURE WORK

We have presented ContraNA, a visual analytics framework for network comparison, which utilizes two machine learning schemes—network representation learning and contrastive learning—together with an intuitive visual interface. ContraNA provides the capability for effectively identifying and understanding unique characteristics of one network relative to another, supporting four key capabilities as outlined by **DIIF**. As our case studies indicate, ContraNA promises to extract insights from networks found in various application domains. Our controlled user study also reflects the usefulness and effectiveness of ContraNA with carefully-designed analysis tasks. In the future, to provide more variations for contrastive network analysis, we plan to extend the framework for comparison of two groups of multiple networks, including dynamic network comparison. We also wish to adapt ContraNA to other cNRL algorithms (e.g., based on GraphSAGE [46], cVAE [80], etc.) and application domains (e.g., brain network analysis).

ACKNOWLEDGMENTS

This research is sponsored in part by the U.S. National Science Foundation through grant IIS-1741536, the Natural Sciences and Engineering Research Council of Canada through the Discovery Grant, and FXPAL through its internship program.

REFERENCES

- [1] The supplementary materials: Videos, datasets, and user study materials. <https://takanori-fujiwara.github.io/s/contrana/>.
- [2] A. Abid, M. J. Zhang, V. K. Bagaria, and J. Zou. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature Communications*, 9(1):2134, 2018.
- [3] A. Abid and J. Zou. Contrastive variational autoencoder enhances salient features. *arXiv:1902.04601*, 2019.
- [4] B. Alper, B. Bach, N. Henry Riche, T. Isenberg, and J.-D. Fekete. Weighted graph comparison techniques for brain connectivity analysis. In *Proc. CHI*, pp. 483–492, 2013.
- [5] S. Aminikhanghahi and D. J. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367, 2017.
- [6] K. Andrews, M. Wohlfahrt, and G. Wurzing. Visual graph comparison. In *Proc. IV*, pp. 62–67, 2009.
- [7] B. Bach, N. Henry-Riche, T. Dwyer, T. Madhyastha, J.-D. Fekete, and T. Grabowski. Small MultiPiles: Piling time to explore temporal patterns in dynamic networks. *Computer Graphics Forum*, 34(3):31–40, 2015.
- [8] B. Bach, E. Pietriga, and J.-D. Fekete. GraphDiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Trans. on Visualization and Computer Graphics*, 20(5):740–754, 2013.
- [9] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic. Time Curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):559–568, 2016.
- [10] A.-L. Barabási et al. *Network Science*. Cambridge university press, 2016.
- [11] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1):133–159, 2017.
- [12] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete. Matrix reordering methods for table and network visualization. *Computer Graphics Forum*, 35(3):693–716, 2016.
- [13] A. Benavoli, G. Corani, and F. Mangili. Should we really use post-hoc tests based on mean-ranks? *J. of Machine Learning Research*, 17(1):152–161, 2016.
- [14] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J.-D. Fekete. GraphDice: A system for exploring multivariate social networks. *Computer Graphics Forum*, 29(3):863–872, 2010.
- [15] G. Bhanot, A. Gara, P. Heidelberger, E. Lawless, et al. Optimizing task layout on the Blue Gene/L supercomputer. *IBM J. of Research and Development*, 49(2.3):489–500, 2005.
- [16] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.
- [17] H. Cai, V. W. Zheng, and K. C.-C. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans. on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.
- [18] H. Chen and B. M. Sharp. Content-rich biological network constructed by mining pubmed abstracts. *BMC Bioinformatics*, 5(1):147, 2004.
- [19] D. Coelho, I. Chase, and K. Mueller. PeckVis: A visual analytics tool to analyze dominance hierarchies in small groups. *IEEE Trans. on Visualization and Computer Graphics*, 26(4):1650–1660, 2020.
- [20] J. Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213, 1968.
- [21] S. R. Collins, P. Kemmeren, X.-C. Zhao, J. F. Greenblatt, et al. Toward a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*. *Molecular & Cellular Proteomics*, 6(3):439–450, 2007.
- [22] T. Crnovrsanin, C. Muelder, R. Faris, D. Felmlee, and K.-L. Ma. Visualization techniques for categorical analysis of social networks with multiple edge sets. *Social Networks*, 37:56–64, 2014.
- [23] W. Cui. Visual analytics: A comprehensive overview. *IEEE Access*, 7:81555–81573, 2019.
- [24] A. Dal Col, P. Valdivia, F. Petronetto, F. Dias, C. T. Silva, and L. G. Nonato. Wavelet-based visual analysis of dynamic networks. *IEEE Trans. on Visualization and Computer Graphics*, 24(8):2456–2469, 2017.
- [25] T. N. Dang, N. Pendar, and A. G. Forbes. TimeArcs: Visualizing fluctuations in dynamic networks. *Computer Graphics Forum*, 35(3):61–69, 2016.
- [26] M. Davis. Palettable. <https://jiffyclub.github.io/palettable/>. Accessed: 2020-3-31.
- [27] A.-H. Dirie, A. Abid, and J. Zou. Contrastive multivariate singular spectrum analysis. In *Proc. Allerton Conference*, pp. 1122–1127, 2019.
- [28] F. Emmert-Streib, M. Dehmer, and Y. Shi. Fifty years of graph matching, network alignment and network comparison. *Information Sciences*, 346:180–197, 2016.
- [29] P. Federico, W. Aigner, S. Miksch, F. Windhager, and L. Zenk. A visual analytics approach to dynamic social networks. In *Proc. of i-KNOW*, pp. 1–8, 2011.
- [30] A. Fornito, A. Zalesky, and E. Bullmore. *Fundamentals of Brain Network Analysis*. Academic Press, 2016.
- [31] M. Freire, C. Plaisant, B. Shneiderman, and J. Golbeck. ManyNets: an interface for multiple network analysis and visualization. In *Proc. CHI*, pp. 213–222, 2010.
- [32] T. Fujiwara, J.-K. Chou, A. M. McCullough, C. Ranganath, and K.-L. Ma. A visual analytics system for brain functional connectivity comparison across individuals, groups, and time points. In *Proc. PacificVis*, pp. 250–259, 2017.
- [33] T. Fujiwara, J.-K. Chou, S. Shilpika, P. Xu, L. Ren, and K.-L. Ma. An incremental dimensionality reduction method for visualizing streaming multidimensional data. *IEEE Trans. on Visualization and Computer Graphics*, 26(1):418–428, 2020.
- [34] T. Fujiwara, O.-H. Kwon, and K.-L. Ma. Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE Trans. on Visualization and Computer Graphics*, 26(1):45–55, 2020.
- [35] T. Fujiwara, J. K. Li, M. Mubarak, C. Ross, C. D. Carothers, R. B. Ross, and K.-L. Ma. A visual analytics system for optimizing the performance of large-scale networks in supercomputing systems. *Visual Informatics*, 2(1):98–110, 2018.
- [36] T. Fujiwara, J. Zhao, F. Chen, Y. Yu, and K.-L. Ma. Interpretable contrastive learning for networks. *arXiv:2005.12419*, 2020.
- [37] C. Gaiteri, S. Mostafavi, C. J. Honey, P. L. De Jager, and D. A. Bennett. Genetic variants in alzheimer disease-molecular and brain network approaches. *Nature Reviews Neurology*, 12(7):413, 2016.
- [38] R. Ge and J. Zou. Rich component analysis. In *Proc. ICML*, pp. 1502–1510, 2016.
- [39] N. Gehlenborg, S. I. O’Donoghue, N. S. Baliga, A. Goesmann, M. A. Hibbs, H. Kitano, et al. Visualization of omics data for systems biology. *Nature Methods*, 7:S56–S68, 2010.
- [40] M. Gleicher. Considerations for visualizing comparison. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):413–423, 2018.
- [41] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.
- [42] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2):129–233, 2010.
- [43] R. Govc. Gragnostics: Fast, interpretable features for comparing graphs. In *Proc. IV*, pp. 201–209, 2019.
- [44] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proc. KDD*, pp. 855–864, 2016.
- [45] M. Hajij, B. Wang, C. Scheidegger, and P. Rosen. Visual detection of structural changes in time-varying graphs using persistent homology. In *Proc. PacificVis*, pp. 125–134, 2018.
- [46] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Proc. NIPS*, pp. 1024–1034, 2017.
- [47] M. Harrigan, D. Archambault, P. Cunningham, and N. Hurley. EgoNav: Exploring networks through egocentric spatializations. In *Proc. AVI*, pp. 563–570, 2012.
- [48] M. Harrower and C. A. Brewer. Colorbrewer.org: An online tool for selecting colour schemes for maps. *Cartographic Journal*, 40(1):27–37, 2003.
- [49] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati, eds., *Human Mental Workload*, vol. 52 of *Advances*

- in *Psychology*, pp. 139–183. North-Holland, 1988.
- [50] N. Henry, J.-D. Fekete, and M. J. McGuffin. NodeTriX: a hybrid visualization of social networks. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.
- [51] Y. Hu. Efficient, high-quality force-directed graph drawing. *Mathematica Journal*, 10(1):37–71, 2005.
- [52] Y. Jia, F. Nie, and C. Zhang. Trace Ratio Problem Revisited. *IEEE Trans. on Neural Networks*, 20(4):729–735, 2009.
- [53] M. John and M. Baumann. A visual approach for the comparative analysis of character networks in narrative texts. In *Proc. PacificVis*, pp. 247–256, 2019.
- [54] I. T. Jolliffe. Principal component analysis and factor analysis. In *Principal Component Analysis*, pp. 115–128. Springer, 1986.
- [55] P. Kerpedjiev, N. Abdennur, F. Lekschas, C. McCallum, K. Dinkla, H. Strobel, J. M. Lubner, S. B. Ouellette, A. Azhir, N. Kumar, et al. HiGlass: web-based visual exploration and analysis of genome interaction maps. *Genome biology*, 19(1):1–12, 2018.
- [56] S. P. Kesavan, T. Fujiwara, J. K. Li, C. Ross, M. Mubarak, C. D. Carothers, R. B. Ross, and K.-L. Ma. A visual analytics framework for reviewing streaming performance data. In *Proc. PacificVis*, pp. 206–2015, 2020.
- [57] A. N. Khambhati, J. D. Medaglia, E. A. Karuza, S. L. Thompson-Schill, and D. S. Bassett. Subgraphs of functional brain networks identify dynamical constraints of cognitive control. *PLOS Computational Biology*, 14(7):e1006234, 2018.
- [58] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*, 2016.
- [59] D. Koop, J. Freire, and C. T. Silva. Visual summaries for graph collections. In *Proc. PacificVis*, pp. 57–64, 2013.
- [60] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. AxiSketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):221–230, 2017.
- [61] O.-H. Kwon, T. Crnovrsanin, and K.-L. Ma. What would a graph look like in this layout? a machine learning approach to large graph visualization. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):478–488, 2018.
- [62] V. Larivière, Y. Gingras, and É. Archambault. Canadian collaboration networks: A comparative analysis of the natural sciences, social sciences and the humanities. *Scientometrics*, 68(3):519–533, 2006.
- [63] A. Lee, D. Archambault, and M. Nacenta. Dynamic Network Plaid: A tool for the analysis of dynamic networks. In *Proc. of ACM CHI*, pp. 1–14, 2019.
- [64] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, 2014. Accessed: 2020-4-10.
- [65] D. Liu, F. Guo, B. Deng, H. Qu, and Y. Wu. egoComp: A node-link-based technique for visual comparison of ego-networks. *Information Visualization*, 16(3):179–189, 2017.
- [66] L. Liu, J. Xu, D. Russell, P. Townend, and D. Webster. Efficient and scalable search on scale-free p2p networks. *Peer-to-Peer Networking and Applications*, 2(2):98–108, 2009.
- [67] S. Liu, X. Wang, J. Chen, J. Zhu, and B. Guo. TopicPanorama: A full picture of relevant topics. In *Proc. VAST*, pp. 183–192, 2014.
- [68] X. Liu and H.-W. Shen. The effects of representation and juxtaposition on graphical perception of matrix visualization. In *Proc. CHI*, pp. 269–278, 2015.
- [69] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, et al. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.
- [70] E. McCrum-Gardner. Which is the correct statistical test to use? *British J. of Oral and Maxillofacial Surgery*, 46(1):38–41, 2008.
- [71] N. Moshiri. The dual-Barabási-Albert model. *arXiv:1810.10538*, 2018.
- [72] S. Murugesan, K. Bouchard, J. Brown, M. Kiran, D. Lurie, B. Hamann, and G. H. Weber. State-based network similarity visualization. *Information Visualization*, 19(2):96–113, 2020.
- [73] M. Newman. *Networks*. Oxford university press, 2018.
- [74] N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.
- [75] T. Reguly, A. Breikreutz, L. Boucher, B.-J. Breikreutz, et al. Comprehensive curation and analysis of global interaction networks in *saccharomyces cerevisiae*. *J. of Biology*, 5(4):11, 2006.
- [76] M. B. Richman. Rotation of principal components. *J. of Climatology*, 6(3):293–335, 1986.
- [77] R. A. Rossi, R. Zhou, and N. Ahmed. Deep inductive graph representation learning. *IEEE Trans. on Knowledge and Data Engineering*, 32(3):438–452, 2020.
- [78] S. Rufiange and M. J. McGuffin. DiffAni: Visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2556–2565, 2013.
- [79] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):241–250, 2017.
- [80] K. A. Severson, S. Ghosh, and K. Ng. Unsupervised learning with contrastive latent variable models. In *Proc. AAAI*, vol. 33, pp. 4862–4869, 2019.
- [81] L. Shi, H. Tong, and X. Mu. Brainquest: Perception-guided brain network comparison. In *Proc. ICDM*, pp. 379–388, 2015.
- [82] N. Smith and S. van der Walt. `mpl.colormaps`. <https://github.com/colormaps/>. Accessed: 2020-3-31.
- [83] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS ONE*, 6(8), 2011.
- [84] G. K. Tam, V. Kothari, and M. Chen. An analysis of machine-and human-analytics in classification. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):71–80, 2017.
- [85] M. Tantardini, F. Ieva, L. Tajoli, and C. Piccardi. Comparing methods for comparing networks. *Scientific Reports*, 9(1):17557, 2019.
- [86] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952.
- [87] C. Turkey, E. Kaya, S. Balcişoy, and H. Hauser. Designing progressive and interactive analytics processes for high-dimensional data analysis. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):131–140, 2017.
- [88] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):1–10, 2016.
- [89] T. von Landesberger, M. Gerner, and T. Schreck. Visual analysis of graphs with multiple connected components. In *Proc. VAST*, pp. 155–162, 2009.
- [90] M. R. Wilkins. Hares and tortoises: The high-versus low-throughput proteomic race. *Electrophoresis*, 30(S1):S150–S155, 2009.
- [91] X. Yang, L. Shi, M. Daianu, H. Tong, Q. Liu, and P. Thompson. Blockwise human brain network visual comparison using nodetrix representation. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):181–190, 2017.
- [92] V. Yoghoudjian, T. Dwyer, K. Klein, K. Marriott, and M. Wybrow. Graph Thumbnails: Identifying and comparing multiple graphs at a glance. *IEEE Trans. on Visualization and Computer Graphics*, 24(12):3081–3095, 2018.
- [93] H. Yu, P. Braun, M. A. Yildirim, I. Lemmens, et al. High-quality binary protein interaction map of the yeast interactome network. *Science*, 322(5898):104–110, 2008.
- [94] W. W. Zachary. An information flow model for conflict and fission in small groups. *J. of Anthropological Research*, 33(4):452–473, 1977.
- [95] D. Zhang, J. Yin, X. Zhu, and C. Zhang. Network representation learning: A survey. *IEEE Trans. on Big Data*, 2018.
- [96] Z. Zhang, P. Cui, and W. Zhu. Deep learning on graphs: A survey. *IEEE Trans. on Knowledge and Data Engineering*, 2020 (Early Access).
- [97] J. Zhao, M. Glueck, F. Chevalier, Y. Wu, and A. Khan. Egocentric analysis of dynamic networks with egolines. In *Proc. of ACM CHI*, pp. 5003–5014, 2016.
- [98] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. MatrixWave: Visual comparison of event sequence data. In *Proc. of ACM CHI*, p. 259–268, 2015.
- [99] J. Y. Zou, D. J. Hsu, D. C. Parkes, and R. P. Adams. Contrastive learning using spectral methods. In *Proc. NIPS*, pp. 2238–2246, 2013.