DOI: 10.1111/cgf.14856

EUROVIS 2023
S. Bruckner, R. Raidou, and C. Turkay
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 3
STAR – State of The Art Report

The State of the Art in Visualizing Dynamic Multivariate Networks

Bharat Kale[1,3], Maoyuan Sun[1], and Michael E. Papka[2,3]

[1]Department of Computer Science, Northern Illinois University, USA
[2]Department of Computer Science, University of Illinois Chicago, USA
[3]Argonne National Laboratory, USA

Abstract

Most real-world networks are both dynamic and multivariate in nature, meaning that the network is associated with various attributes and both the network structure and attributes evolve over time. Visualizing dynamic multivariate networks is of great significance to the visualization community because of their wide applications across multiple domains. However, it remains challenging because the techniques should focus on representing the network structure, attributes and their evolution concurrently. Many real-world network analysis tasks require the concurrent usage of the three aspects of the dynamic multivariate networks. In this paper, we analyze current techniques and present a taxonomy to classify the existing visualization techniques based on three aspects: temporal encoding, topology encoding, and attribute encoding. Finally, we survey application areas and evaluation methods; and discuss challenges for future research.

CCS Concepts

• ***Human-centered computing*** → ***Graph drawings;***

1. Introduction

Networks are important data structures for showing relationships in complex datasets. Networks are commonly used for modeling and solving many real-world problems across various domains (e.g., biology, business, transportation, communication, and security). A network, also known as a graph, consists of nodes (vertices) and edges (links). Nodes can represent real-world entities (e.g., people, organizations, and locations), and edges represent the relationships among these entities. Oftentimes, the nodes and edges of a network are also associated with various types of attributes. The primary challenge of visualizing such a network, called a multivariate network, is to consider the structure of the network while encoding the associated attributes. However, most real-world networks are both dynamic and multivariate in nature, which means their nodes, edges, and associated attributes are evolving with time. We refer to these networks as dynamic multivariate networks (DMVNs). Techniques for visualizing DMVNs must handle the grand challenge of visually showing the network structure and associated attributes along with the changes in the structure and/or attributes over time.

Firstly, visualizing a multivariate network is a challenging task because of the visual clutter that arises when representing the network structure and associated attributes together. The dynamic nature of a DMVN introduces yet another set of challenges for researchers due to its evolving topology and attributes. Available techniques for visualizing DMVNs use one of two approaches: encoding all the dynamics within the same view or using multiple coordinated views. Either way, the challenge is in choosing an efficient encoding scheme such that the techniques used for visualizing the network structure, attributes, and their temporal changes do not interfere with one another. For instance, when using color on nodes and edges to emphasize the addition or removal of nodes and edges with time, a designer should compromise on using color to encode categorical features on nodes and edges. Conversely, using colors to encode attributes on nodes restricts the designer in using color for emphasizing the structural changes with time.

DMNVs require understanding of network structure, node and edge attributes, and how these will change with time. Each of these topics have been studied extensively by visualization researchers. For example, graph drawing is an area of computer science and mathematics that deals with producing optimal graph layouts using different node placement strategies. Researchers in this field study the network structure and design novel and effective network layout algorithms. Multivariate networks discusses techniques for visualizing both network structure and associated attributes. Dynamic networks deals with studying changes in graph layouts with time. However, addressing the dynamic nature of both network structure and attributes at the same time is not well studied and remains an open research problem.

In this paper, we conduct a comprehensive survey and intend to report the state of the art in visualizing DMVNs. Our key contributions include a new taxonomy of DMVN visualization techniques and a comprehensive discussion of recommendations, challenges, and future directions.

We start with a brief introduction to the related surveys (Section

2) and provide definitions that are central to the understanding of the DMVNs, their characteristics (Section 3), followed by various tasks that users try to accomplish using DMVNs (Section 4). Later we discuss our methodology for conducting this survey (Section 5). Based on our survey of the collected literature, we introduce a new taxonomy for DMVN visualization techniques (Section 6). A supplementary website provides the overview of our taxonomy along with an interactive tool to filter publications at intersections of categories in the taxonomy. We also discuss common application areas (Section 7) and strategies for evaluating DMVN techniques (Section 8). Finally, we discuss the challenges, identified through this survey, for future research (Section 10).

2. Background

Existing literature on network visualization techniques is rich and constantly growing. There are several surveys and taxonomies on different variants of networks and their visualization techniques (see Table 1), such as the design space of temporal graph visualization [KKC14] and State-of-the-Art Reports (STARs) on dynamic graphs [BBDW17] and multivariate graphs [NMSL19]. While the prior works are exhaustive, they focus on studying either the dynamic or multivariate nature of the networks rather than analyzing the challenges posed when both characteristics are present. Most relevant to DMVNs is the survey on temporal multivariate networks by Archambault et al. [AAK*14]. The work focused on visualizing DMVNs with node-link diagrams, and surveyed their applications in software engineering, but did not cover matrix- and list-based visualization techniques, or emphasize techniques for visualizing the evolution of attributes. With this paper, we aim to fill this gap by describing the design space of visualization techniques that simultaneously encode the dynamic and multivariate characteristics of a network, and systematically categorize them.

Besides the surveys and taxonomies on network visualizations, approaches to visual comparisons by Gleicher et al. [GAW*11] and the design space of composite visualizations by Javed and Elmqvist [JE12] are of particular interest to our work. We draw on their generic categorization of visual approaches for organizing multiple coordinated views to categorize techniques in our corpus.

3. Dynamic and Multivariate Nature of Networks

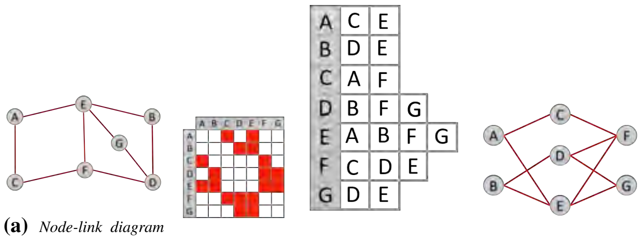
Before going into the survey, we introduce the definition of a network and its variants that are related to the topic of this research.

Definition 3.1 A **network** (or **graph**), represented as $G := (N, E)$, is composed of a set of objects called nodes (or vertices) N, and relationships between the nodes called *edges* E: $E \subseteq N \times N$. This type of network is also called a **static network**, because the nodes and edges of the network do not change.

A network is a mathematical structure, meaning that there is no inherent spatial position associated with nodes to arrange them in an n-dimensional space. However, in order to visualize networks, we need a strategy to place nodes in a given space and represent relationships. Node-link diagrams are one of the heavily studied network representations. Layout algorithms are well studied in the literature to generate readable network visualizations. Other common approaches include matrix visualizations and list-based representations. Figure 1 shows an example of different network representations. Von Landesberger et al. [VLKS*11] conducted an extensive survey on visual analysis of large graphs to discuss visualization techniques for various types of networks.

Table 1: *Related State-of-the-Art Reports.*

Reference	Description	Key difference
[HS12]	Reviewed the applications of dynamic networks across different fields	Focused only on the dynamic nature of network structure
[KKC14]	Consolidated the design space of dynamic network visualization techniques.	
[KKC15]	Derived a task taxonomy to explain various analysis tasks with dynamic networks.	
[BBDW17]	Surveyed visualization techniques for dynamic networks and derived a taxonomy.	
[WEF*14]	Discusses interaction techniques for multivariate graph visualizations and provides guidelines for novel designs.	Focused only on the encoding of attributes and structure of static networks simultaneously
[NMSL19]	Surveyed visualization techniques for multivariate networks and proposed typologies for tasks and visualization techniques.	
[AAK*14]	Studied node-link diagrams for dynamic multivariate networks and surveyed their applications in software engineering.	Focused on both dynamic & multivariate nature but only covered node-link visualization techniques
[HSS15]	Created a generic multi-facet graph visualization framework by unifying taxonomies that focus on single facet at a time but did not cover matrix-based and hybrid visualizations.	



(a) *Node-link diagram (force directed)* **(b)** *Matrix* **(c)** *Adjacency list* **(d)** *List view*

Figure 1: *Visual representations of a network topology.*

Definition 3.2 A **dynamic network** is defined as a network whose nodes and edges can change over time.
A dynamic network is denoted as $T := (G_1, G_2, G_3, \ldots, G_n)$, where $G_i \; \forall \; i \in \{1, 2, \ldots, n\}$ is a static graph and i indicates a time step.

Different from static networks, structure evolves over time in dynamic networks. Capturing these structural changes is the key aspect of dynamic network visualization techniques. Multiple comprehensive surveys have been conducted to study dynamic networks, regarding task taxonomy [KKC15] and visualization technique taxonomy [BBDW17]. Figure 2 gives an example of visualizing dynamic nature: all three time steps are juxtaposed as node-link diagrams. We can see the differences in the network structure as we move from one time step to the next.

Definition 3.3 A **multivariate network** is a network whose nodes or edges, or both, have attributes associated with them.
A multivariate network is denoted as $\overline{G} := (\overline{N}, \overline{E}, \rho)$, where

$\overline{N}$ represents *nodes*,

Table 2: *DMVN tasks at various levels of granularity categorized (rows) into three categories (columns). Domain-specific examples from the literature are listed for each combination of level of granularity and task category.*

Granularity	***Change*-oriented Tasks**	***Search*-oriented Tasks**	***Comparison*-oriented Tasks**
Network	Describe the evolution of whole network with time - *How do inter-academic collaborations evolve from early to later stages of the timeline? [PTLZ18]* - *Which software revision introduced the most substantial changes? [VBW15]*	Find nodes associated with a given attribute pattern for at least n time steps - *Identify which author used the word "visualization" in the longest span of consecutive years [GGK*11]* - *In 1996, who co-authored with the largest number of people? [GGK*11]*	Compare the temporal trends in node/edge attributes of network - *What are the attribute similarities over time for the top 7 spatially-close authors in the network? [LZH*17]* - *How are merchants with similar attributes connected in time? [BBS*20]*
Subnetwork	Describe the evolution of a selected cluster with time - *What is the trend of the red group? Does it grow, shrink, remain stable, or become unstable? [BPF13]* - *In what year is the red group the largest? [BPF13]*	Find the groups over which a specified pattern occurs for at least n time steps - *Identify which venue keeps the strongest relationship with IEEE TVCG over time [LZH*17]*	Compare the evolutionary patterns of selected clusters in terms of attributes and relationships - *How different is the venue group "journal of visualization" compared to "IEEE TVCG" in terms of number of authors and stable relationships? [LZH*17]*
Individual	Describe the evolution of ego network and attribute trend of a selected node - *Did the overall network size of an ego increase or decrease in year1–year2? [ZGC*16]* - *How is the discussion on bird flu in 2013 on Sina Weibo influenced by celebrities? [LHS*15]*	Find the alters with a specified pattern for at least n time steps - *How many people had relationships with the ego for n+ years? [ZGC*16]* - *How many alters of Kwan-Liu Ma collaborated in more than 1 paper consecutively? [HZL*16]*	Compare the evolutionary patterns of ego networks and domain attributes of selected egos - *Compare ego networks of ENRON CEOs Jeffrey Skilling and Kenneth Lay [FMW*21]* - *Ego network of Daniel A. Keim vs. Kwan-Liu Ma [PTLZ18]*

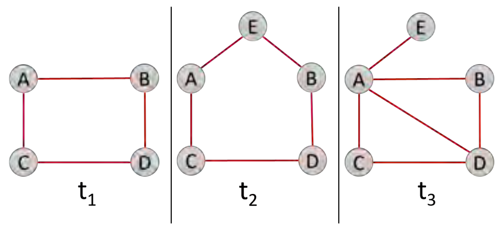


Figure 2: *Small multiples arrangement of node-link diagrams from consecutive time steps.*

$\overline{E} = (E_1', E_2', ..., E_k')$: E_j' represents a set of all edges with j^{th} attribute value $\forall\ j \in \{1,2,...,k\}$, and

ρ represents a function that returns an attribute vector for any node $n \in \overline{N}$

Multivariate visualization techniques are used to analyze network structure and node and edge attributes together. Nobre et al. [NMSL19] introduced new typologies to study tasks and techniques for visualizing multivariate networks.

Definition 3.4 A **dynamic multivariate network** is defined as a network whose nodes, edges, and attributes can change over time. It is denoted as $T := (\overline{G}_1, \overline{G}_2, \overline{G}_3, ..., \overline{G}_n)$, where $\overline{G}_i\ \forall\ i \in \{1,2,...,n\}$ is a multivariate graph and i indicates a time step.

Archambault et al. [AAK*14] studied the modeling of time and attributes in DMVNs using node-link diagrams in detail, but did not introduce a taxonomy to classify existing techniques.

4. Dynamic Multivariate Network Tasks

Understanding the tasks that users need to address is an important aspect of designing and evaluating useful visualization techniques. Here we review related task taxonomies and discuss the tasks, along with domain-specific examples for DMVNs at various levels of granularity. A majority of visualizations focus on static networks; hence, many task taxonomies were proposed to meet the growing demand for visualizing dynamic networks, yet only a few include the dynamic nature of both topology and attributes of networks.

Ahn et al.'s [APS13] task taxonomy focuses on tasks for studying network evolution. They arranged the tasks along three dimensions: *entities*, *properties*, and *temporal features*. Though the *properties* dimension considers both structural and domain properties, their taxonomy as a whole does not focus on tasks that rely on temporal changes in both topology and domain attributes. Bach et al. [BPF13] tried to simplify the taxonomy with an alternative proposal that categorizes tasks as *temporal*, *topological*, and *behavioral*. The *behavioral* tasks consider the dynamics in structural properties computed using network algorithms but not domain properties in the data. Later, Kerracher et al. [KKC15] proposed a more generic taxonomy to improve the task coverage by considering the full picture. They included tasks that depend on temporal dynamics in both the network topology and attributes.

In the context of networks, the primary targets of analysis tasks are topological structures ranging from a whole network to a single element. We distinguish between three levels of granularity, similar to Ahn et al. [APS13]: *network*, *subnetwork*, and *individual* levels. *Network*-level tasks focus on temporal changes in topology and attributes of the entire network. *Subnetwork*-level tasks focus on the evolution of groups, such as clusters or paths, and the temporal changes in their attributes. *Individual*-level tasks focus on a single node or edge in the network. Table 2 gives examples of DMVN analysis tasks across the different levels of network granularity along with domain-specific examples from the literature. At each level of granularity, we categorized tasks into three categories: *change*-oriented tasks, *search*-oriented tasks, and *comparison*-oriented tasks. Change-oriented tasks focus on tasks studying evolution of the topological structures. Search-oriented tasks focus on finding topological structures based on certain attribute and temporal patterns. Comparison-oriented tasks focus on tasks comparing topological structures based on attributes or comparing attribute trend of a topological structure across time steps.

5. Methodology

Our focus is on techniques for visualizing DMVNs. We followed a systematic approach and retrieved the techniques from an identified list of references. We started by defining the scope of our study and then compiled a corpus by manually searching the relevant journals and conference proceedings as well as references cited in the collected publications. Next, we analyzed the collected references by tagging each reference according to the publication type, application area, evaluation method, and techniques used for encoding time, network structure, and attributes.

5.1. Scope

The scope of this study is on the techniques that specifically aim to visualize both the dynamic and multivariate nature of networks. We also consider techniques that focus on simplifying the network structure before visualizing it, through mechanisms such as aggregation and clustering, if they include the techniques for visualizing the evolution of simplified network structures. There are certain visualization techniques that address special cases of DMVN visualizations (e.g., dynamic tree comparison problems [GGPPS13] and dynamic graph comparison problems [YDK*18]). Such specializations are not in the scope of this study.

5.2. Corpus

We began our data collection process by identifying candidate papers from the journals and conference proceedings shown in Table 3. We collected all the papers published in the past decade (2011–2021), by manually scanning the title and abstract of each publication. This process led to an initial collection of 78 papers. Next, we filtered out the papers that focused on either dynamic networks or multivariate networks alone, which generated a new set of 52 papers. We then expanded our corpus using the references cited in the candidate papers and the publications that cited the candidate papers according to *Google Scholar*. The final step resulted in a corpus of 104 papers.

Table 3: *List of Journals and Conferences used for collecting our corpus.*

Type	Name
Journals	*Computer Graphics Forum* *IEEE Transactions on Visualization and Computer Graphics* *Information Visualization*
Conferences	*ACM Conference on Human Factors in Computing Systems (CHI)* *Eurographics Conference on Visualization (EuroVis)* *IEEE Visualization Conference (VIS)* *IEEE Pacific Visualization Symposium (PacificVis)* *International Conference on Advanced Visual Interfaces (AVI)*

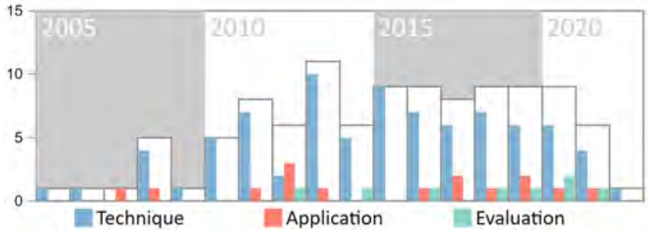


Figure 3: *Number of publications on DMVNs per year in our corpus. White bars indicate total number of publications per year and colored bars represent the counts by type.*

5.3. Data Analysis

Our data analysis had two rounds of codings to organize the literature, as described by Nobre et al. [NMSL19]. In the first round, we focused on the general characteristics of papers — publication year, application area, and evaluation type. Evaluation type was coded using a fixed set of tags (see Table 4) whereas application area was left more open-ended. In the second round, we focused on paper type and the visual encodings used in the proposed techniques, and focused primarily on tagging instead of strict categories so that we could flexibly assign multiple tags to each paper. Here, we only used a subset of publications to derive a reasonable number of tags. Next, we consolidated the tags by merging the similar ones to get a final set of tags to use for the rest of the publications, and grouped the tags into meaningful categories. Finally, we used the categories to organize the design space and present the overview.

Table 4 shows the categories, final set of tags used in each category, and their descriptions. The publication *type* is mainly used for differentiating the collected publications. We distinguish among 3 types: *Technique* papers are the basis for our taxonomy as they form the set of novel visualization approaches. Publications tagged as *application* are used to give the overview of the DMVN applications (Section 7). Publications tagged as *evaluation* are used to discuss the evaluation approaches employed for DMVN techniques (Section 8). Figure 3 shows the breakdown of the corpus by publication *year* with color coded by publication *type*.

Table 4: *Categories and tags used for analyzing the corpus along with the descriptions.*

	Category	Description
Year	year	publication year of the paper
Type	application	application of a DMVN visualization to a specific scenario
	evaluation	evaluation of DMVN visualization techniques through studies
	technique	novel visualization technique
Application Area	biology	biological applications, e.g., protein interactions
	business	applications using business data, e.g., transactions and stocks
	communication	applications of communication networks
	document collections	applications, e.g., interactions between text documents
	security	security network applications, e.g., hacking networks
	social networks	social network applications, e.g., collaboration networks
	software engineering	applications studying evolution of software components
	sports	applications analyzing sports data
	transportation	transportation networks, e.g., flight routes
	other	all other applications including generalized approaches
Evaluation Type	use case	informal demonstration of usage with examples
	case study	evaluation involving domain experts' usage and feedback
	user study	formal study with users, reports quantitative results
	survey	special report on broad survey of related field
	theoretical	arguments based on visualization principles and comparisons with similar and popular tools
	algorithmic	using algorithmic metrics, e.g., performance
Topology	node-link	structure encoded using node-link diagrams
	matrix	structure encoded using adjacency matrices
	list view	structure encoded using adjacency lists or list views
	hybrid	structure encoded using combination of node-link, matrix, and list
Temporal	one time slice	sequentially presenting one time step after the other
	multiple time slices	presenting multiple time steps by mapping time to space
	embedded timeline	embedding the timeline into the structure of the network
Attributes	juxtaposed marks	juxtaposing attributes in a separate view from the network structure
	nested marks	encoding attributes on the visual elements of the network
	attribute-driven layouts	integrating attributes in to the network structure to create the layout

(a) *Single time step* **(b)** *Multiple time steps* **(c)** *Embedded timeline*

Figure 4: *Temporal encoding approaches.*

6. Design Space of Dynamic Multivariate Networks

There are many techniques for visualizing DMVNs. Specifically, our corpus has 76 publications that proposed DMVN visualization techniques. To structurally organize the techniques, we classify them along three dimensions: temporal encoding that deals with encoding the temporal changes; topology encoding that regards encoding the network structure; and attribute encoding that considers encoding the attributes of nodes and/or edges in the network.

Each technique discussed in this section can be described by selecting at least one approach from each of the three dimensions. Hence, we first provide a brief description of each dimension, along with their associated subcategories. Then, we present the taxonomy by classifying the works that are at the intersection of approaches from the three dimensions.

6.1. Temporal Encoding

Temporal encoding is the key aspect of DMVN visualization techniques. While there are other aspects (i.e., topology and attributes), encoding temporal changes is the distinguishing characteristic of DMVN visualizations compared to static network visualizations. Temporal changes can be encoded by mapping time to *time* or to *space*. Beck et al. [BBDW17] categorized techniques that map time to time visually as *animation*-based techniques and those that map time to space as *timeline*-based techniques. The *timeline*-based techniques are more generic and can be further subclassified based on whether the timeline is clearly separated or completely embedded into the network structure. Hence, we classify the temporal encoding dimension into three subcategories: *single time step*, *multiple time steps*, and *embedded timeline*. Figure 4 shows an illustration of them. *Single time step* approaches present users with only one time step at a time. They primarily use transition techniques (e.g., animations and interpolations) to allow users to follow the changes between time steps. *Multiple time steps* approaches present users with multiple time steps arranged in a readable manner. The approaches in the *embedded timeline* embed the time dimension into the network structure and present one single aggregated network visualization. Thus, encoding temporal changes in this category is tightly coupled with technique used for topology encoding. More detailed discussion is provided in Section 6.4.3.

6.2. Topology Encoding

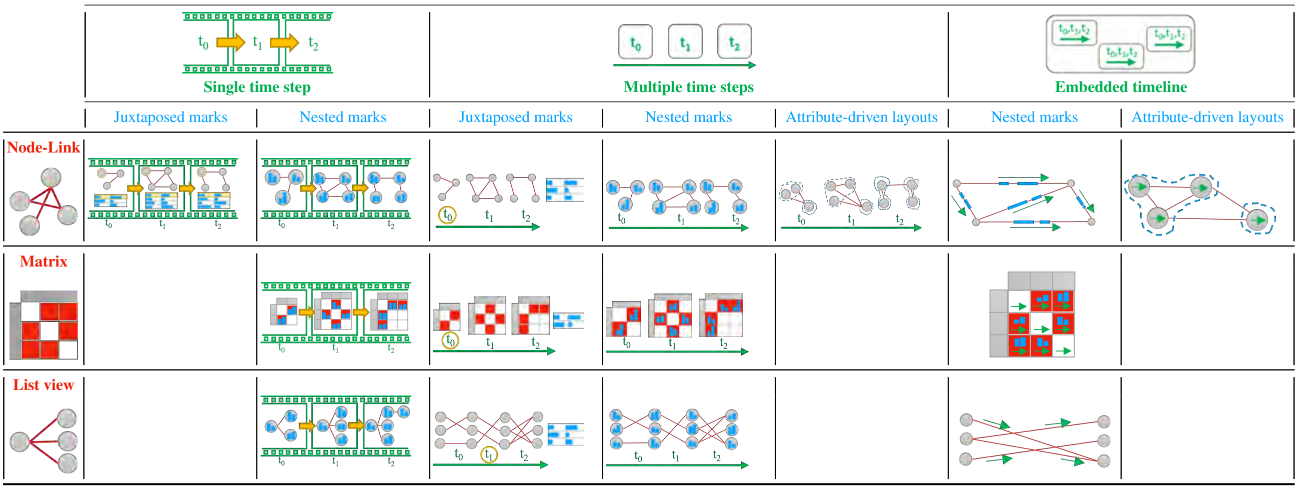
The key challenge of topology encoding is how to show the relations among nodes in a readable manner. The underlying network structure is the primary deciding factor in choosing the encoding technique [VLKS*11]. We distinguish among three approaches for encoding topology: *node-link diagrams*, *matrices*, and *list views*. As shown in Figure 1, *node-link diagrams* represent entities in the network using nodes and relationships using links; *matrices* use rows and columns to show entities and cells to encode relationships (i.e., bipartite relations [SNR14,FSB*13]); *list views* use nodes and links but arrange nodes along parallel axes. Topology encoding is a major research focus in static graph visualizations. In the literature, list view based representations (Figure 1d) are considered as node-link diagrams, but we categorized them separately because they are independent of the layout algorithms, structurally organize the space for nodes and edges, and support more granular tasks.

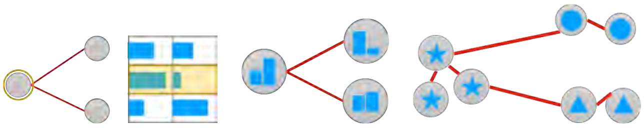
6.3. Attribute Encoding

The key challenge of attribute encoding is how to show attributes associated with nodes and/or edges of the network. This can be considered as designing related multiple-view visualizations [SNK*21, SSAZ21, SKAS22]. Javed and Elmqvist [JE12] proposed generic approaches for organizing multiple coordinated views, which have five categories: *juxtaposition*, *superimposition*, *overloading*, *nesting*, and *integration*. In the context of DMVNs and based on the tags in our corpus, we distinguish between three approaches for encoding attributes: *juxtaposed marks*, *nested marks*, and *attribute-driven layouts*.

Juxtaposed marks (Figure 5a) use a separate view to show attribute data of the network. *Nested marks* (Figure 5b) use the

Table 5: *Overview of the taxonomy of DMVN visualization techniques along with illustrations of the categories.*





(a) *Juxtaposed marks* **(b)** *Nested marks* **(c)** *Attribute-driven layout*

Figure 5: *Attribute encoding approaches.*

(a) *automated* **(b)** *interactive*

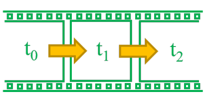
Figure 6: *Navigation mechanisms in 'single time step' approaches.*

on-node/on-edge encoding approach discussed by Nobre et al. [NMSL19]. In this approach, the visual appearance of the nodes and/or edges are modified either by using visual channels (e.g., size and color), or by embedding glyphs. In the last approach, *attribute-driven layouts* (Figure 5c), attributes may not be visualized explicitly but they are implicitly used to compute the network layout. For instance, node positions in the network layout may be calculated based on attribute similarity, or nodes with similar attributes may be grouped to reduce the network size.

6.4. Taxonomy

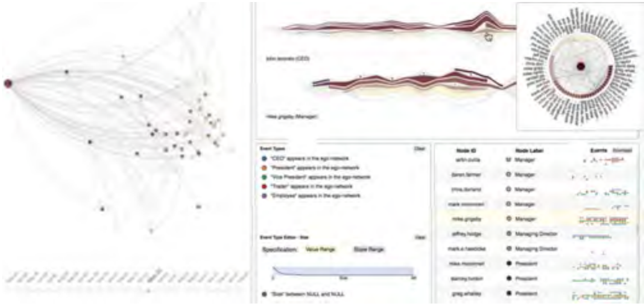
We present a hierarchical taxonomy, with three levels, to organize the categories. The first level, using Roman numerals, shows subcategories in temporal encoding techniques; the second level, represented by lower case alphabets, shows topology encoding techniques; and the third level, represented by numbers, shows attribute encoding techniques. Table 5 gives an overview of the categories in the proposed taxonomy along with illustrations. For the reminder of this section, we provide a description of the categories and techniques at the intersection of the subcategories for each dimension.

6.4.1. I. Single time step

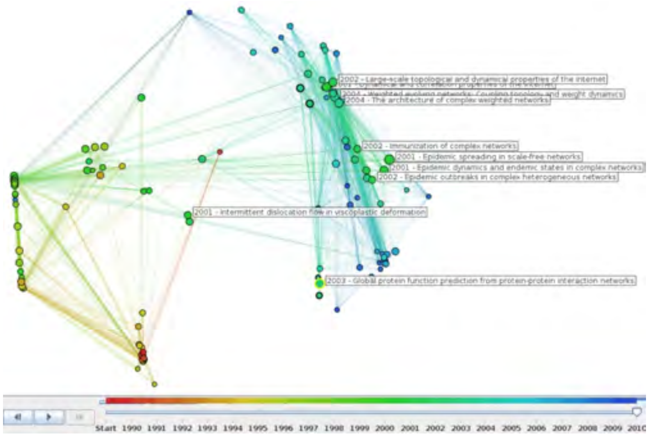


In this category, time steps in the data are mapped to time in the visualization (i.e., time steps are dynamically presented by replacing the current time step with the next one in sequence). These techniques allow users to see a snapshot of the network at one time step only, at any given point of time. Figure 6 shows two time navigation mechanisms employed by the approaches in this category: *automated navigation* and *interactive selection*. *Automated navigation* mechanisms use play/pause buttons to start/stop the animation between the time steps while allowing the transitions by one time step at a time. On the other hand, *interactive selection* mechanisms give more finer control to users by allowing them to quickly move to a time step of interest.

6.4.1.1. ***I.a Node-link diagrams:*** Node-link diagrams provide intuitive layouts for visualizing networks. However, special care must be given to readability and to computational aspects of the layouts, as these can, and do, become unstable when the structure of the network changes. If the changes are not minimal during transitions, maintaining a user's mental model becomes difficult (i.e., requires the user to remember previous time steps rather than focus on the transitions). Hence, existing works to date focus on complex computations involved in developing layouts that can adapt to the network structure with minimal changes during time step transitions. The first network layout adaption algorithm was introduced by Misue et al. [MELS95]. Later, Frishman and Tal [FT08] improved adaptive-force-directed layouts by using a fast, GPU-based algorithm. Another set of approaches tried to preserve node positions across the whole timeline [DGK00] or define initial node placement strategy to minimize transitions [HMHU13]. Sorger et al. [SWKA19] explored ad hoc immersive visual analytics approaches for dynamic networks using 3D force-directed layouts in virtual reality headsets.



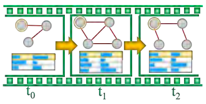
(a)



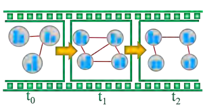
(b)

Figure 7: *Examples of encoding network structure and attributes with a single time step approach. (a) Segue uses interactive timeline to choose a time step and multiple juxtaposed views for network structure and attributes [LWB18]. (b) Encoding few attributes on nodes and edges using color and size visual channels [ABPdO12].*

I.a.1 Juxtaposed marks (Single time step + Node-link): Encoding of topology and temporal changes in DMVNs restrict the possibilities for showing attributes using visual channels on nodes and links in network layouts. Using a juxtaposed view for attributes and employing mechanisms to coordinate interactions between the views offers a good alternative approach. Ahn et al. [ATMS*11] created NodeXL as an extension to spreadsheet software, supporting dedicated views for dynamic network topology and node attributes. Law et al. [LWB18] studied spatial layouts and created *Seque* (Figure 7a) where attributes representing different events in the whole sequence are presented in a juxtaposed view.

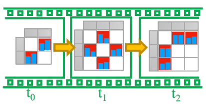


I.a.2 Nested marks (Single time step + Node-link): In this category, typical approaches to encode attributes use available visual channels on nodes and/or links. As it is already difficult to follow the changes in topology due to the animation, encoding diverse attributes using complex markers on nodes and/or edges increases the user's cognitive load and makes the visualization more challenging to understand. For nodes, using shapes for categorical attributes, size for numerical attributes, and color for both numerical and categorical attributes are common practices. Feng et al. [FWSL11] use color to encode node importance, Alencar et al.'s *Time-Aware visualization* [ABPdO12] technique uses color to encode temporal attributes on nodes in the supergraph (Figure 7b), and Portenoy & West [PW16] use both color and size of nodes to encode domain specific attributes on nodes. For edges, using width for numerical attributes, and color for numerical and categorical attributes are common choices. Hurter et al. [HEF*13] use color to encode dynamics in the numerical attributes on edge bundles. Few works use markers to embed more attribute information on nodes. For example, Abello et al. [AHSS13] designed marks to encode both size and density of a subnetwork and embedded it in aggregated nodes to represent the overview of the corresponding subnetworks. Extra care must be taken in designing approaches that use visual channels to emphasize topology changes. For instance, color is used during transitions to highlight the new nodes/edges being added to the network and existing nodes/edges being deleted from the network [FT08, HMHU13, CSW21].



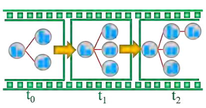
6.4.1.2. ***I.b Matrices:*** While always theoretically possible, using matrix diagrams for laying out network topology in conjunction with animated transitions is a relatively recent practice. Rufiange & Melançon [RM14] created *AniMatrix* by animating matrix representations of network snapshots in time. They studied software evolution by using staged transitions that show changes related to various types of entities and relationships in a certain order that help in preserving a user's mental map.

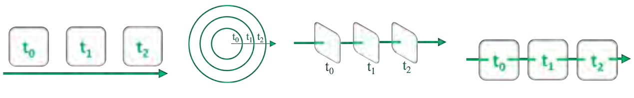
I.b.1 Nested marks (Single time step + Matrix): Color coding matrix cells, row headers, and column headers is heavily used by Rufiange & Melançon [RM14] in the *AniMatrix*. Temporal changes in topology such as addition (progressing from gray to green) and deletion (progressing from gray to red) of relationships are emphasized using color transitions in matrix cells. Domain-specific attribute information (e.g., software component types) is encoded on row and column headers.



6.4.1.3. ***I.c List views:*** A list view is another type of node-link diagram but the arrangement of the nodes and links are more systematic than traditional layouts (e.g., a force-directed layout). Typically, in a list view, all the partitions of nodes in the network are arranged along one axis and the nodes in each partition along another axis. This organized layout helps in improving the readability of dynamics in the network structure by easily maintaining the node positions during transitions [SMNR15, ZSCC20]. List views have become increasingly popular in network visualization compared to other layout approaches. Clyde et al. [CKS*21] created *ChemoGraph* using list views for visualizing chemical networks. In *ChemoGraph*, the transitions are triggered by user actions, leading to the dynamic expansion of the network by adding new nodes and edges.

I.c.1 Nested marks (Single time step + List view): List views has the ability to segregate space for visualizing nodes and edges in the network. This arrangement eliminates occlusion from overlapping nodes and also reduces clutter due to





(a) *small multiples* **(b)** *Radial* **(c)** *New dimension* **(d)** *Integration*

Figure 8: *Approaches for arranging multiple time steps.*

edges crossing over nodes, thus allowing the usage of marks to encode both complex and more attributes, especially on nodes. In *ChemoGraph* [CKS*21], the glyph design facilitates encoding of chemical structure of a compound along with various numerical attributes on a node in the network.

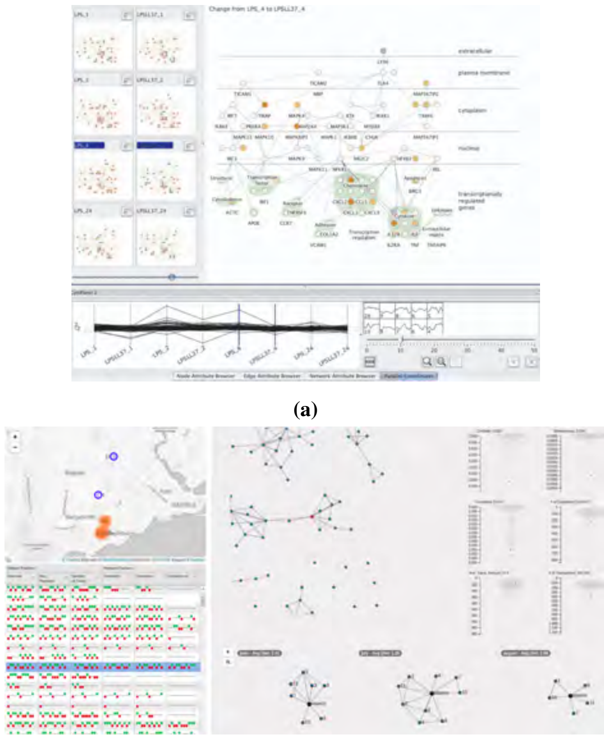
6.4.2. II. Multiple time steps

In this category, time steps are mapped to space (i.e., multiple time steps are statically presented together by arranging them in a readable manner). Hence, these techniques offer better overviews of the temporal changes and help in comparing different time steps. However, due to the limited space for each time step, visual scalability is the main challenge to *multiple time steps* approaches. Figure 8 illustrates four common approaches used in the existing works for arranging time steps along a timeline.

6.4.2.1. ***II.a Node-link diagrams:*** A simple approach presented by Ahmed et al. [AFH*09] uses a radial layout (Figure 8b) for juxtaposing time steps. Node-link diagrams suffer from clutter even with networks of moderate size. Using domain-specific information in designing node-link layouts help in such scenarios. *Cerebral* [BMGK08] follows a small-multiples approach (Figure 8a) where each snapshot uses an optimized node-link layout by using vertical position to facet nodes as per the positions of their membranes. Liu et al. [LHS*15] created a fast and constrained graph layout algorithm to simplify the topology in each time step. In *MobilityGraphs* [VLBR*15], Von Landesberger et al. uses a calendar view for juxtaposing node-link diagrams where representation of each time step is optimized by using graph-based spatial clustering to control the visual clutter. Bach et al. [BKH*16] created *graph comics*, combining comics and node-link diagrams, to present temporal changes in networks. Their studies prove that *graph comics* are useful for reaching wider audiences due to their expressiveness. Pham et al. [PND*20] use force-directed layout for each time step while restricting horizontal position to represent time. This force-directed layout arranges nodes in the vertical orientation resulting in a compact view that visually highlights temporal trends.

Itoh & Akaishi [IA12] follow a 3D approach (Figure 8c) by visualizing each time step as a node-link diagram on a 2D plane while the 2D planes are placed along a third dimension. Users can interactively drag the 2D plane along the time dimension to visualize the changes in the network. Gohnert et al. [GZD*15] followed a similar approach but instead of interactively allowing a user to drag the 2D plane to present the structure in the time step, all the time steps are presented statically while maintaining the node positions across the time steps. Gohnert et al. also used straight lines to connect nodes along the time dimension to easily track a node's lifeline.

Integrating time using explicit links across time steps (Figure

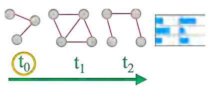


(a)

(b)

Figure 9: *Examples of juxtaposed attribute views on multiple time steps of node-link diagrams. (a) Cerebral using domain information to drive the layout in each time step with a juxtaposed parallel coordinates view for attributes [BMGK08]. (b) Juxtaposed pixel display to encode temporal changes in attributes [BBS*20].*

8d) is also studied in the existing works. Cui et al. [CWL*10] employed a summarizing approach where they used information entropy measures to estimate each time step and plot it on a time-series line plot. Time steps of interest can be visualized in detail using an optimized force-directed layout in the form of a word cloud. *EgoSlider* [WPZ*15] and *DyEgoVis* [FMW*21] visualize projections of ego networks in each time step and integrate them to study the distribution and evolution of similar ego networks.



II.a.1 Juxtaposed marks (Multiple time steps + Node-link): Multiple time step approaches already use a majority of the available space for encoding time step sequences. On top of that, juxtaposing attributes in separate coordinated views is highly challenging and add to existing visual scalability issues. Prior works show that combining a small-multiples arrangement of time steps with juxtaposed attributes works well with a limited number of time steps and compact visualizations for attribute encoding. Cerebral [BMGK08] (Figure 9a) uses a juxtaposed parallel-coordinates view to show associated attributes. Boz et al. [BBS*20] (Figure 9b) use a juxtaposed pixel display to present both domain and network attributes. Fu et al. [FMW*21] use multiple juxtaposed views to emphasize various network attributes of ego net-

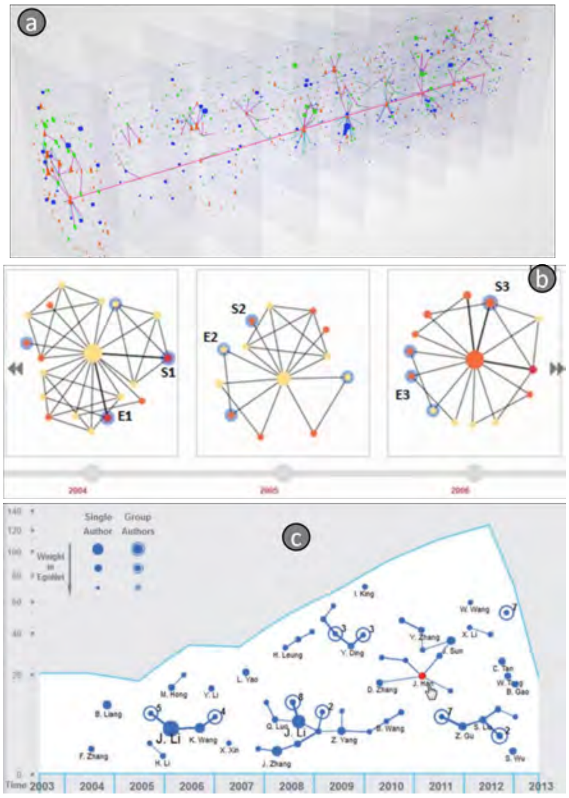
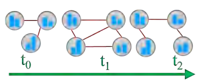
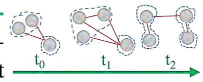


Figure 10: *Examples of nested attributes on multiple time steps of node-link diagrams. (a) Time steps are arranged along a third dimension and attributes are encoded with color and size visual channels on nodes and edges [GZD*15]. (b) [HZL*16] & (c) [LHS*15] Simple markers are embedded on nodes to encode attributes.*

works and their evolution. One juxtaposed view encodes similarities between ego networks per time step and another encodes evolution states of an ego network. Cui et al. [CWL*10] represents each time step using a word cloud, where color is used to encode the appearing behavior and importance of words.

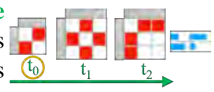


II.a.2 Nested marks (Multiple time steps + Node-link): Multiple time steps with embedded marks for encoding attributes is one of the most highly used categories of DMVN visualization techniques. Ahmed et al. [AFH*09] encode images on nodes to present the country the node represents, and size to encode network attributes (e.g., centrality). *Egonetcloud* [LHS*15] (Figure 10c) and *MENA* [HZL*16] (Figure 10b) use a combination of visual properties and simple glyphs to represent various data and network attributes. *DualNetView* [PND*20] embed glyph inside nodes to encode multiple data attributes. Itoh & Akaishi's [IA12] and Gohnert et al.'s [GZD*15] (Figure 10a) 3D approach also encode multiple attributes on nodes and edges using visual properties (e.g., color and size on nodes; and color, width, and length on edges).

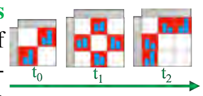


II.a.3 Attribute-driven layouts (Multiple time steps + Node-link): For large networks, traditional node-link layouts are not effective due to visual clutter. To address such scalability issues, aggregating nodes and links based on topology is a commonly used approach. In DMVNs, attributes also play an important role. Hence, aggregating nodes based on both structural and attribute similarities [ZCY09] helps in designing effective layouts. Instead of using raw attribute values to encode on nodes and edges, several studies proposed using them to drive the layout creation and simplify the visual representation [VLBR*15,SLC*17]. Xu et al. [XTYL18] proposed node embeddings to capture syntactic and semantic information in large texts to represent nodes in a vector space.

6.4.2.2. ***II.b Matrices:*** Matrices offer more strict layouts and are less flexible than node-link diagrams. However, due to their advantages in providing readable visualizations for dense and large networks, prior works use them to design new visualization techniques for DMVNs. A straightforward approach is to use a small-multiples arrangement of time steps (Figure 8a) where each time step is represented by a matrix [ZGC*16, ZSCC19]. With the increase in network size and number of time steps, this technique quickly gets ineffective due to limited space. Hence, advanced techniques to radially juxtapose the matrices (Figure 8b) are introduced [BD08, BHW11, VBSW13]. Bach et al. [BPF14] proposed a 3D approach and designed *Matrix Cube*. In *Matrix Cube*, each time step is represented as an adjacency matrix and the matrices are stacked along a third dimension (Figure 8c) to create a cube representation. Following the same space-time cube metaphor, Schneider et al. [STSB16] created *CuboidMatrix*. Another way to arrange multiple time steps is to integrate multiple time steps with explicit links to connect same entities across time steps (Figure 8d). Vehlow et al. [VBW15] proposed a novel technique called matrix of adjacency matrices where the timeline of an entity is emphasized via flow lines connecting time steps.



II.b.1 Juxtaposed marks (Multiple time steps + Matrix): *NetVisia* [GGK*11] uses heat maps to display node attribute changes over time. Instead of representing network structure, each time step shows the distribution of attribute values (Figure 11a). The small-multiples view of multiple time steps shows the changes in the distribution. To improve the readability with large datasets, *NetVisia* uses attribute-based clustering to aggregate similar entities. Juxtaposed tables from two different time steps are used to compare time steps of interest.



II.b.2 Nested marks (Multiple time steps + Matrix): Small-multiples arrangements of time steps limits the available space for embedding enriched marks into matrix cells, so encoding complex attributes is challenging. Existing approaches in the corpus rely on color coding and hence are limited in the number of attributes they can encode [BD08, BHW11, VBSW13, ZGC*16] (Figure 11b). Vehlow et al. [VBW15] integrated multiple time steps while using different colors to highlight hierarchy among groups of entities, changes in network structure, and similarity metrics between different groupings across time steps (Figure 11c). The cells

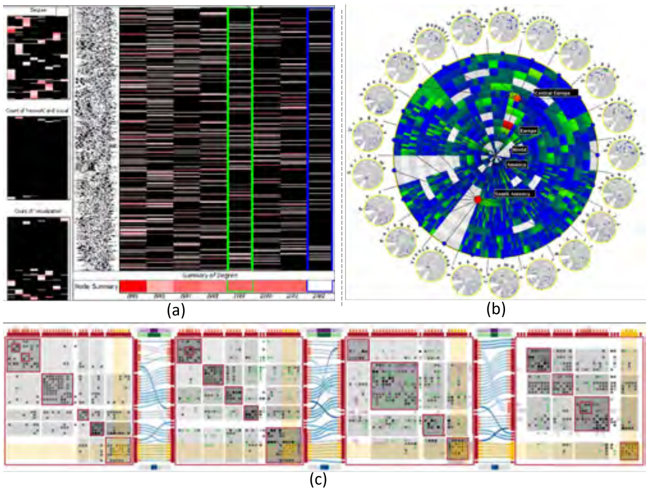


Figure 11: *Examples of attribute encoding on multiple time steps of matrices. (a) Juxtaposed previews of attribute changes using compact pixel displays [GGK*11]. (b) Time Radar Trees with radial arrangement of time steps while using color to encode edge attributes inside cells [BD08]. (c) Integrating time steps using explicit flow lines to track the timeline while matrix cells encode edge attributes using color [VBW15].*

in *Matrix Cube* [BPF14] and *CuboidMatrix* [STSB16] are 3D cubes whose color and size are used to encode edge attributes in the data.

6.4.2.3. ***II.c List views:*** In a list view, nodes are placed along one dimension. This makes small-multiples arrangements of time steps (Figure 8a) easy and creates a compact visualization of a DMVN. Burch et al. [BVB*11] created *parallel edge splatting* as a scalable technique for dynamic networks. Nodes are arranged vertically as lists in parallel axes, and edges are splatted. Edges in each time step are encoded between consecutive pairs of parallel axes. John et al. [JSS*13] also employed list views for chemical reaction networks. The reactions are shown on a timeline and user-selected time steps are represented as bipartite networks using list views. Linhares et al. [LTPR17] studied approaches to visually highlight different aspects of network evolution—arranging nodes based on frequency of connections in time and revealing node activity over time using a temporal activity map. Their approaches align the nodes in the network across time steps to follow their timelines easily. Valdivia et al. [VBP*19] created a visualization of dynamic hypergraphs based on a novel approach called *Parallel Aggregated Ordered Hypergraph*, where nodes are represented using parallel horizontal bars and hyperedges using vertical lines. Riegler et al. [RWDHP19] conducted a study to evaluate effectiveness of list views for dynamic networks using a small dynamic call graph.

List views are intuitive to integrate time (Figure 8d) using explicit links in the layout due to the linear arrangement of nodes. This arrangement allows easy tracking of the lifetime of each entity, so it is ideal for egocentric analysis tasks. *Sven* [AB14] and *DMNEVis* [PTLZ18] use list views to explicitly represent the timeline of individual entities using horizontal lines. Reda et al. [RTJ*11] introduced a technique, applying a time series metaphor, where time

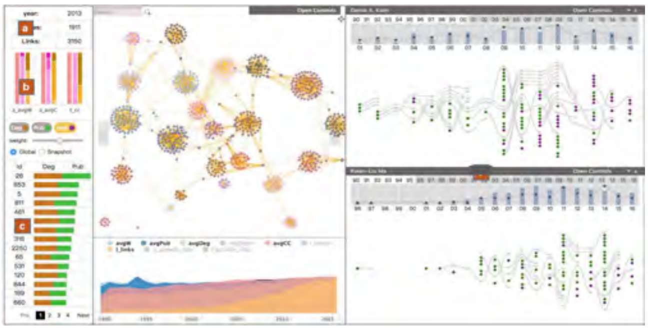
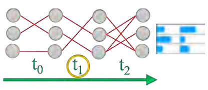
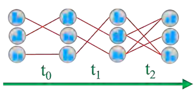


Figure 12: *DMNEVis showing multiple juxtaposed views encoding various domain and structural attributes. Node-link view shows supergraph across all time steps while list views present ego networks of selected egos. [PTLZ18].*

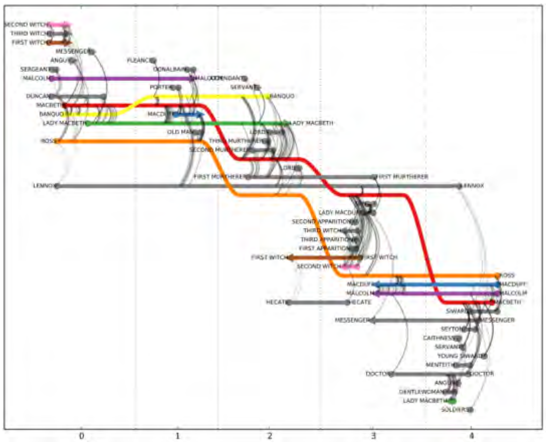
is encoded along the horizontal axis and individual entities are positioned along the vertical axis. The vertical position of nodes is based on the communities in each time step. This generates a compact representation highlighting when communities are disbanding and when new entities are joining a community. Van den Elzen et al. [vdEHBvW13] proposed *Extended Massive Sequence Views*, where nodes are put in vertical lists arranged along a timeline. Parallel vertical lines are used to reduce clutter while showing edges. To make the visualization further compact, they extended the work to create *Circular Massive Sequence Views* [vdEHBvW13]. Dang et al.'s *TimeArcs* [DPF16] draws on force-directed layouts and adds additional forces to maintain the same vertical position for entities of the same type, pull vertices horizontally based on the time step, and maintain close proximity vertically for nodes forming clusters.



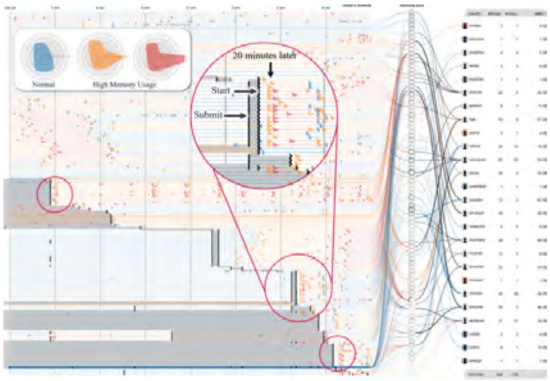
II.c.1 Juxtaposed marks (Multiple time steps + List view): Peng et al. [PTLZ18] created *DMNEVis* to study evolution of ego networks in DMVNs. *DMNEVis* supports multiple juxtaposed views to encode attributes of interest and highlight their temporal changes (Figure 12). Stoiber et al. [SRG*19] conducted a design study and created *netflower* focusing on understanding quantitative flows in dynamic networks. In *netflower*, a sankey diagram is used to present the network structure at a given time step, while the nodes are juxtaposed by visualizations that highlight temporal changes in multiple node attributes.



II.c.2 Nested marks (Multiple time steps + List view): List views organize the space into two parts, one for nodes and the other for edges, so that it is possible to encode attributes on both nodes and edges without worrying about overlap. In *parallel edge splatting* [BVB*11], edges are color coded with attribute data. Many existing techniques depend on color to encode node attributes [WCPB12, DPF16, LTPR17, VBP*19, WSM*18] and edge attributes [RTJ*11, AB14, ZGC*16, PXN*19, SKZ*19]. Several techniques also use markers designed to encode more and diverse attributes [WPZ*15]. Nguyen et al. [NHC*20] embed radar charts on nodes in the network to encode 8 numerical attributes (Figure 13b).



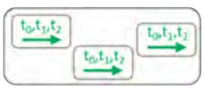
(a)



(b)

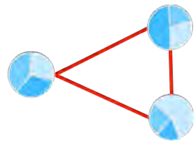
Figure 13: *Examples of nested attributes on multiple time steps using list views. (a) SVEN uses story line visualizations with attributes encoded by color [AB14]. (b) RadarViewer uses radar charts to encode 8 numerical attributes on nodes in the list view [NHC*20].*

6.4.3. III. Embedded timeline

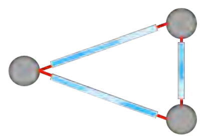


In the *embedded timeline* category, approaches embed the temporal changes within the existing visual elements used for the network structure. This makes them rely heavily on the topology encoding techniques. For instance, small time-series charts or other marks can be designed to track temporal changes and encode them on the network elements (e.g., cells in case of matrices, nodes and edges in case of node-link diagrams and list views). Thus, techniques in this category focus on showing a timeline of individual entities and their relations in a DMVN.

6.4.3.1. ***III.a Node-link diagrams:*** The existing approaches to embed temporal changes in the structure using node-link diagrams include embedding representative marks on the visual elements of the network. Multiple encodings have been designed by customizing these marks for some specific data types and tasks to address the challenges in showing changes in structure and attributes together. Figure 14 shows two common approaches: *On-node encoding* [PGFL05, APBG19] and *On-edge encoding* [LL16].

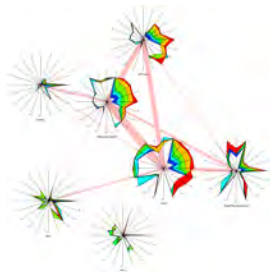


(a) *on-node encoding*



(b) *on-edge encoding*

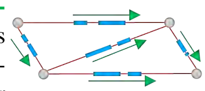
Figure 14: *Techniques for embedding temporal changes in node-link diagrams.*



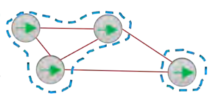
(a)

(b)

Figure 15: *Embedding temporal changes into the network structure. (a) Kiviat graph of 7 components in the Mozilla web browser across 7 versions where each component encodes 20 metrics per version [PGFL05]. (b) Link segmentation algorithm to divide an edge into segments based on time steps while each segment encodes multiple edge attributes [LL16].*



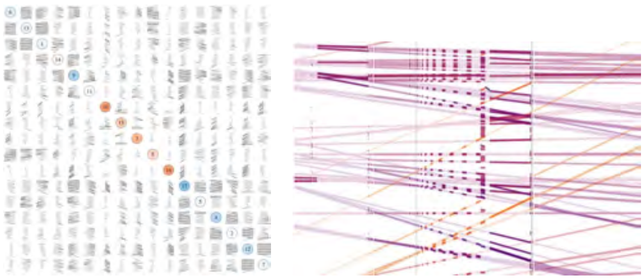
III.a.1 Nested marks (Embedded timeline + Node-link): The key challenge of this category of approaches is resolving the conflicts between embedded encodings used for capturing changes in both the network structure and attributes. Prinzger et al. [PGFL05] introduced the *Kiviat graph* (Figure 15a) inspired by the design of radar charts. *Kiviat graphs* can encode multiple attributes on nodes while integrating multiple releases of software into a single compact visualization. Li & Liao's [LL16] representation uses edges in the network to embed both the structural and attribute changes with time (Figure 15b). Each edge is divided into k segments, where k is the number of time steps. Each segment is further divided to encode categorical or numerical attributes of the relation. If an edge is not present in a time step t_i, the i^{th} segment of the edge will be grayed out. Alexandru et al.'s *Evo-Clocks* [APBG19] also uses a glyph to embed multiple attributes on nodes in the DMVN representing evolution of a project software.



III.a.2 Attribute driven layouts (Embedded timeline + Node-link): Hadlak et al. [HSCW13] use trends in temporal attributes of nodes in the network to group the entities with similar trends and visualize large DMVNs. Each group in the aggregated network is revealed as a node and edges reflect the connections between the groups. Each node is also embedded with a

(a) *Bar chart* **(b)** *Gestalt lines* **(c)** *Pixel-based* **(d)** *Time series*

Figure 16: *Techniques for embedding temporal changes in matrix cells.*

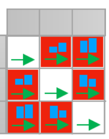


(a) **(b)**

Figure 17: *Embedding temporal changes into the network structure. (a) Gestaltmatrix shows evolution of group structures and rankings over 15 time steps [BN11]. (b) Time aligned edge plots presents the dynamics in a software call graph of an open source Java software using a link segmentation algorithm [ALHW20].*

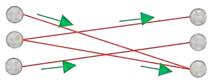
small time-series chart highlighting the temporal trend of the group. Alexandru et al. [APBG19] use hierarchies present in the software components to drive the node positions in the network layout.

6.4.3.2. ***III.b Matrices:*** With a structured layout, matrices offer better ways to embed the timelines of both structure and attributes inside the cells. Existing approaches include bar charts (Figure 16a), gestalt lines (Figure 16b), pixel-oriented visualizations (Figure 16c), and time-series plots (Figure 16d). The marks along the diagonal reveal temporal changes in a node attribute, and the marks in remaining cells represent temporal changes in the relationships between the corresponding entities.



III.b.1 Nested marks (Embedded timeline + Matrix): Stein et al. investigated pixel-oriented visualizations to embed timelines into matrix cells [SWS10]. *TimeMatrix* [YEL10] and Burch et al.'s [BSW13] techniques use bar charts in matrix cells. *TimeMatrix* allows dynamic aggregation of nodes to show aggregated timelines and improves the readability of overviews. Brandes & Nick [BN11] designed *gestaltmatrix* (Figure 17a), where each matrix cell is encoded using *gestaltlines*. *Gestaltlines* are created by combining Tufte's *Sparklines* [Tuf06] with Gestalt principles; and they can encode attributes using visual channels such as color, orientation, length, and thickness.

6.4.3.3. ***III.c List views:*** List views can also embed timelines on edges using link-segmentation-based algorithms, while this has only been tried recently. Similar to edge-based encodings in node-link diagrams, *time-aligned edge plots* [ALHW20] use a time- and edge-scalable visualization technique to segment each edge for encoding temporal changes.



III.c.1 Nested marks (Embedded timeline + List view): In *time-aligned edge plots* [ALHW20] (Figure 17b), the relationship between two nodes over a period of time is encoded on a single line using segments. Each segment is color coded to show an edge attribute during the corresponding time step. Each segment can be further divided to encode multiple edge attributes.

7. Applications

DMVN applications span across diverse domains. Our corpus include areas such as document collections [WSL*18], transportation [KAW*14, SGB19], and security [HL12], but the majority are from social networks, biology, and software engineering (Figure 18). In this section, we briefly discuss DMVN visualizations across three major application domains using examples from the collected literature. We do not claim that our work is a comprehensive review of the domains and applications of DMVNs, as the range of conferences and journals covering representative examples from the published work is too broad for the scope of this paper.

7.1. Social Networks

Social network data is prevalent in today's world due to the increasing popularity of digital communications and social media. The increasing availability and popularity of the digital world makes social network data highly dynamic. An important aspect of social network analysis (SNA) is to analyze the structural properties and attributes of the network to identify the interaction patterns among actors [WF*94, CM11]. The challenge in the case of DMVNs is that the visualization techniques should capture the network structure and attributes as well as the dynamics. Initial applications of DMVN visualizations leverage transition techniques to encode dynamic nature using node-link diagrams [MMBd05, BdM06]. Several applications of DMVNs in social networks that were studied in the literature include visualizing dynamics in online communities [ATMS*11], evolving collaborations in co-authorship networks [BCD*10, LZH*17, ZSC*21], and co-citation networks [Che06]. Due to the large number of attributes (i.e., attributes in the data and those derived from SNA algorithms), DMVN visualizations often combine juxtaposed marks and nested marks. For instance, TMN-Vis [LZH*17] uses multiple coordinated views for showing different aspects of the network, where the main view reveals the network topology with nested attributes and others present different sets of attributes using juxtaposed marks.

Several systems leverage matrix visualizations to explore large, locally dense and highly connected DMVNs [SWS10, YEL10]. In both applications, the network is displayed as an adjacency matrix, while temporal changes are embedded into cells and headers using pixel-oriented visualizations in PixelMatrix [SWS10] and bar charts in TimeMatrix [YEL10]. To support users in exploring the temporal patterns, they offer interactive features such as semantic zooming, hierarchical aggregation, and dynamic filters. Due to the trade-offs between intuitively showing topology using node-link diagrams and reducing clutter using matrices for SNA visualizations,

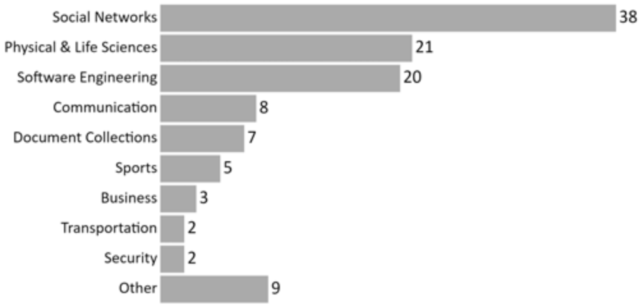


Figure 18: *Number of publications by application area in our corpus.*

several DMVN visualizations implemented both approaches and conducted comparative studies [LAN19, BTBC*21, NWHL20].

List views partition the space based on the time steps and follow a flow metaphor to link nodes across time steps using a time line. This makes it easy to follow the timeline of a node, so they are well-suited for egocentric network analysis tasks [LZH*17, PTLZ18]. Nested marks are used to encode the multivariate nature directly on nodes and edges.

7.2. Physical & Life Sciences

Physical sciences study natural but non-living objects that include physics, chemistry, and astronomy; while life sciences study living organisms that include biology and medicine fields. This is another vast area for DMVNs. Gehlenborg et al. surveyed the applications in systems biology [GOB*10] and concluded that node-link diagrams dominate visualizations in this field because of their ability to easily perform path-related tasks while encoding dynamic relationships. Common examples of applications in this space include biological networks [SDMW09], evolution of metabolic pathways [RUK*10], protein interactions [WPPW14, FH21], and chemical reaction networks [GV20]. Complex attributes on both nodes and edges is a common characteristic of these networks, which are typically encoded using either juxtaposed or nested marks.

Cerebral [BMGK08] uses small-multiples to show the network at multiple time steps, while attributes are revealed on nodes with a juxtaposed parallel coordinates plot. Rohrschneider et al. [RUK*10] analyze the structure and attributes of dynamic pathways by visualizing all time steps in a single view using a node-link diagram. The nodes are color coded based on their first appearance. Despite the heavy reliance on node-link layouts, few works also leveraged matrices. Burch and Diehl [BD08] study the evolution of dependencies in gene sequences using a radial arrangement to stack time steps in the data. In MultiPiles, Bach et al. [BHRD*15] investigate brain connectivity networks by piling hundreds of time steps using matrix representation.

7.3. Software Engineering

Dynamic network visualizations were applied to software-related data to visualize the complex dependencies among software components [KLRZ94]. Dynamic call graphs [TV08, BMR*12], which describe the dynamics in the control-flow between subroutines in a program, is one of the main applications in this field. These graphs comprise many static and dynamic attributes (i.e., component types and various software quality metrics). For instance, annotated semantic graphs for C++ programs [TV08] use node-link diagrams and juxtaposed attribute views containing more than 100 different attribute types.

A software typically involves multiple versions and releases. In this context, a more generic application is studying the evolution of software by visualizing the dynamics in the relationships between modules across different versions. Rufiange and Melancon [RM14] combined matrix visualization and animations to study the evolution patterns in software design. There are also other works that combine node-link diagrams and matrix visualizations. For instance, Burch proposed dynamic graph wall [Bur17] to study the evolution of call graphs from an open source software system, that uses multiple visual metaphors. The visualization of a snapshot of network in time using a suitable representation is based on the sparsity of the snapshot. The survey by Archambault et al. [AAK*14] gives an overview of DMVN applications in software engineering.

8. Evaluation

The tags used for identifying evaluation techniques that researchers employ are summarized in Table 4. We tagged all the papers in our corpus for the evaluation methods using the listed closed set of tags. We identified that most of the papers are evaluated using up to two approaches in various scenarios with *use cases* being one of them. Overall, our analysis found that *use cases* (58%) is a highly employed evaluation method, followed by *user studies* (24%), *case studies* (10%), *surveys* (8%), *algorithmic evaluations* (6%), and *theoretical evaluations* (4%). While most papers perform some type of evaluation, some papers focus only on the evaluation aspect. Our corpus also has few such papers identified using paper type tag (see 'Evaluation' category in Figure 3). In this section, we discuss insights gained from understanding what is being evaluated across multiple evaluation studies. In the current literature on DMVNs, evaluation studies are mainly focused on node-link diagrams [AP12, FM16, LAN20], while very few studied matrix-based visualizations [VABK20] and compared both techniques [FAM21]. List views are not yet evaluated for their effectiveness in visual exploration and analysis of DMVNs.

8.1. Use cases

Use cases are considered a lightweight evaluation technique. Papers in our corpus mainly used them for walking through examples on how a visualization technique or a tool can be used. When a novel technique is introduced, it is easy to explain the usefulness of the technique with the help of well-known datasets in the visualization community such as interactions between characters in Les Miserables, movies and their actors, and collaboration and co-citation networks from popular conferences. Also, these readily available datasets do not require extensive domain knowledge. In our corpus, 73% percent of *application* papers and 66% of *technique* papers rely on use cases for the evaluation. We believe their heavy usage is likely due to the simplicity and easy management in conducting the studies using existing real-world datasets. A major disadvantage with *use cases* is the identification of proper datasets that can demonstrate the diverse applicability, complexity, and scalability of the technique.

8.2. User studies

While different evaluation methods provide different insights into the proposed techniques and tools, evaluation of the usability requires users. *User studies* allow researchers to formulate specific research questions and recruit users from diverse backgrounds to use the tool and understand from their experience. These studies help in quantitative assessment of the proposed technique either by comparing it with existing techniques or by changing the parameters to study different aspects of a single technique. While conducting *user studies* is complex, they are popular because of their ability to bring diverse perspectives into the evaluation and usually result in quantitative metrics. We observed that completion time and error rate are the most widely used metrics for quantitative assessment.

Preserving mental map is considered the most important factor for the techniques under *Single time step* category. Many studies were conducted to study the importance of mental map preservation on task performance [AP12]. They categorized the tasks in these studies into readability tasks (i.e., reading the structural information in a graph) and memorability tasks (i.e., remembering the structural dynamics in a graph). Interestingly, among the two studies conducted by Archambault and Purchase, one did not find significance of mental map preservation on task performance [AP12], while the other claims that preserving mental map resulted in a significantly faster and accurate response for path-related tasks [AP13]. This indicates that the aspect of mental map relies on the tasks used for a study. Also, these studies focused only on temporal and topology aspects of DMVNs. Specifically, they studied effectiveness of node-link visualizations for showing dynamics in topology without considering alternative layouts and multivariate nature of networks.

Another key idea driving the majority of the evaluation studies is comparing different techniques for performing the same tasks, such as comparing node-link diagrams with matrices [OJK18, FAM21] and comparing *single time step* (e.g., animating time steps) with *multiple time steps* (e.g., small-multiples arrangements of time steps) [FM16, LAN20]. While multiple studies proved that node-link diagrams are preferred for topology-related tasks [OJK18], Filipov et al. [FAM21] argue that matrix visualizations are preferred for temporal navigation tasks in terms of completion time, error rate, and participant preference of technique. When comparing animation with small multiples, Bach et al. [BPF13] showed that staged animations reduce the error rate significantly with a slight increase in time, especially for tasks related to searching nodes and edges. Another study by Lee et al. [LAN20] claims that animation-based approaches are better for comparing consecutive time steps whereas small multiples are better for analyzing distant time steps.

Several specific approaches are also evaluated using user studies. Zhao et al. [ZGC*16], evaluated their adjacency matrix approach with a integrated timeline for egocentric tasks against node-link diagrams. They found their approach works better for both temporal and topology tasks, and for tasks involving focus on an individual node. Other studies [WPZ*15, WCB16] show that list-view-based visualizations with integrated timelines work better than node-link visualizations regarding completion time and error rate for exploratory tasks, but no significant difference for navigation tasks.

8.3. Case studies

When a visualization system is designed to address some specific problem in a scientific domain, collaboration with domain experts is critical. These systems are generally complex and unique. *Case studies* are used in these scenarios to solicit feedback from the domain experts as background knowledge in the domain is important to use these systems. Sometimes these studies generate quantitative metrics but interviewing the experts is the most common procedure used [SWKA19, SRG*19, FAS*20]. For instance, Bach et al. [BHRD*15] evaluated their technique of piling matrices in time in the context of studying brain connectivity data from people with Parkinson's disease. The medical experts used the tool for several days in close collaboration with the authors where various parameters in the visualization system are statistically studied for their practical use.

8.4. Algorithmic evaluations

This is the most technical method for evaluating systems, especially when novel algorithms are proposed. In case of DMVNs, these are heavily used for evaluating layout algorithms in node-link visualizations [FT08, FWSL11, HMHU13, CSW21] and reordering algorithms in matrix visualizations [CSJ*20]. Feng et al. [FWSL11] used GPU implementation of the proposed force-directed layout algorithm for visualizing dynamic networks and studied various aspects of layout stabilization during temporal navigation. Cheong et al. [CSW21] compared their proposed variant of a force-direct layout to the existing variants for studying different aesthetic criteria of the resulting node-link visualizations. They showed that the proposed initial positioning method is most effective when visualizing structural dynamics in the networks.

8.5. Theoretical evaluations

Theoretical evaluations are the most uncommon and challenging type of evaluation method mainly for two reasons: 1) they require deep theoretical background in visualization principles; and 2) convincing readers with theoretical arguments is difficult. Hence this method is usually accompanied by another approach such as a *use case* [HEF*13, SLC*17]. For instance, Hurter et al. [HEF*13] studied the effectiveness of edge bundling in encoding dynamics in relationships while giving theoretical arguments and practical use cases using real-world datasets across multiple domains.

9. Recommendations

Here, we present recommendations while designing DMVN visualizations. Table 6 summarizes pros and cons with respect to different categories along each of the three dimensions in our taxonomy.

Our corpus indicates that the *multiple time steps* category is the most popular for visualizing DMVNs (Figure 19). This is mainly

Table 6: *Summary of pros and cons corresponding to the techniques along each of the three dimensions in our taxonomy.*

Taxonomy Category	Pros	Cons
Single time step	1. Ideal for comparing consecutive time steps. 2. Good for eye catching presentations.	1. Preserving mental map is challenging 2. Less suitable for exploratory tasks.
Multiple time steps	1. Ideal for comparing distant time steps. 2. Provides whole summary of the dataset. 3. Ideal for exploratory tasks.	1. Visual scalability is a major concern. 2. Difficult to see rate of change across time steps.
Embedded timeline	1. Ideal for generating most compact summary visualizations. 2. Temporal trends between connected entities is evident.	1. Visualizations quickly become overwhelming.
Node-link	1. Highly intuitive for network representation. 2. Effective for path tracing tasks.	1. Unstable layouts. 2. Visual clutter hinders readability.
Matrix	1. Ideal for dense and highly connected networks.	1. Not intuitive for path tracing tasks.
List view	1. Stable layout. 2. Intuitive for tracking timeline of a node across time steps.	1.Poor visual scalability.
Juxtaposed marks	1. Scale well for large networks with many and heterogenous attribute types.	1. Not ideal for tasks that require focusing on topology and attributes simultaneously.
Nested marks	1. Easy to understand network for both topology and attributes. 2. Working well for sparse networks.	1. Working only for limited number of attributes.
Attribute driven layouts	1. Good for presenting attribute trend in the structure. 2. Helps in managing large networks.	1. Using multiple attributes to drive the layout is difficult with out going into high dimensional techniques.

because of their ability to show multiple time steps at a time while supporting exploration and analysis via interaction techniques. This approach suits well if the user tasks involve comparisons across time steps, especially the ones that are distant in time. Techniques in *single time step* category require users to rely on their memory. This results in navigating back and forth between time steps which incurs interaction cost [APP10]. However, for comparing consecutive time steps, *single time step* techniques are preferred as gradual transitions easily highlight changes. *Embedded timeline* techniques are ideal for creating compact visualizations that summarize dynamics in a single view.

Static networks are studied extensively for techniques to present topology, especially node-link and matrices. For DMVNs, node-link diagrams are most intuitive but the stability of their layouts pose challenges in understanding the temporal changes, while matrices are still preferred for dense and highly connected networks. List-view-based visualizations offer both intuitive and highly structured layouts for networks. Existing studies so far treated list-view-based network visualizations as node-link diagrams. We categorized them separately because of their ability to be independent of layout algorithms and better support micro-level tasks that involve focusing on an individual node, its neighborhood, and timeline.

In general, visualization techniques for DMVNs do not scale well to large networks. Using attribute driven layouts can be helpful to either reduce the network size by grouping entities with similar attributes [HSCW13, VLBR*15] or by removing visual representation of attributes and integrating them into the layout [SLC*17]. Regarding techniques for showing attributes, juxtaposed attributes scale well to many heterogeneous attributes, but they lack integration between topology and attributes due to separate views. Nested attributes help with such integrated tasks, but they do not scale well due to the availability of limited space.

Overall, *multiple time steps* techniques using node-link diagrams or list- view-based visualizations with nested attributes is the most popular combination used in the literature (Figure 19). While selecting a suitable visualization technique depends on analysis tasks, the summarized pros and cons of encoding techniques help visualization practitioners identify suitable technique along each dimension and create usable DMVN visualization tools.

10. Challenges and Future Directions

While significant progress has been made in designing visualizations for DMVNs, there are still several research challenges to investigate. Our taxonomy allow us to see the areas that were studied in detail and those that are still in early stages (Figure 19). Here we discuss five important open challenges identified based on the surveyed literature. The challenges and future directions reflect our opinions, and they partly overlap with prior studies [AAK*14].

10.1. Visual Scalability

The amount of data displayed should not affect the readability of the visualizations. One of the key challenges with DMVNs is scaling of the techniques to a large number of time steps and attributes. Filtering and aggregation are commonly used approaches in this scenario. Alternatively, this limitation can be avoided by designing visualizations tailored for large displays, which can take advantage of distributed layouts. Designing sophisticated graph layouts has been a major area of study in static graph visualizations and it will continue. Researchers created distributed graph visualization

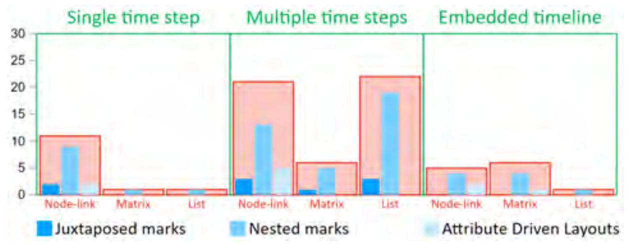


Figure 19: *Number of publications at the intersection of techniques along the three dimensions in our DMVN taxonomy.*

algorithms [MGL06, ADLM18], layouts specialized for visualizing large-scale networks. Employing these sophisticated techniques for designing DMVN visualizations is a promising future direction for improving the scalability.

Prior studies to evaluate scalability of techniques discuss static and multivariate networks in depth. For instance, node-link and matrix diagrams are compared extensively for static networks [OJK18, NWHL20]. Beck et al. [BBD09] argue that for dynamic networks, number of time steps is crucial for studying scalability. For DMVNs, we also need to consider the number of attributes to be encoded as an important factor. To our knowledge, there is still lack of studies investigating existing DMVN techniques for their scalability towards number of time steps and attributes.

10.2. Attribute Dimensionality

Existing visualizations can encode a few attributes using visual properties (e.g., size, shape, and color), while well-designed glyphs can increase the number. However, DMVNs in many domains have large number of attributes to encode, especially in software engineering and biology. Using dimensionality reduction techniques and attribute-based aggregations help in reducing the attribute dimensions, but existing techniques still lack the ability to correlate network structure and attributes while capturing dynamics in both.

10.3. Interactive Visualizations

Interaction is a pervasive element in visualization techniques. Especially with large DMVNs, it is important to navigate the time dimension, manage the network size, and modify the attribute encodings for performing exploratory tasks effectively. Existing systems apply the interactions available for static and multivariate networks that allow exploration along utmost two dimensions to DMVNs. However, DMVNs require focusing on three dimensions, structure, attributes, and dynamics, simultaneously. This calls for research in designing advanced interactions that use both structural and temporal properties during navigation.

10.4. Evaluation Studies

User studies measuring accuracy and completion time of user tasks dominated the literature on DMVNs. Another common measure we encountered is cognitive load on the user, especially when animation is involved. These quantitative studies are based on a few low level user tasks for the specific technique/system being evaluated. However, in general, the evaluation is still lacking in the context of generic tasks for DMVNs.

The majority of evaluation approaches focus on node-link visualizations, while a few study matrix visualizations. The suitability of different DMVN visualization techniques for different tasks still remains unanswered formally. We need elaborate studies for visualization techniques, interaction, qualitative measures, and benchmarks for comparison, such as a well-designed study for comparing node-link diagrams, matrices, and list views, especially for tasks, which explore temporal dynamics in both topology and attributes.

10.5. Collaborative Analysis

For applying analysis techniques to large-scale datasets across various domains and solving complex problems, expert collaboration is necessary [Kee06]. However, DMVN visualizations were not investigated in the context of collaborative analysis even though their applications span across diverse domains. Hence studying integration of collaborative systems and DMVN datasets would be beneficial. In particular, specifics in collaboratively exploring graphs and the temporal changes in them needs to be addressed.

11. Conclusions

In this paper, we present the state of the art in visualizing DMVNs and provide a taxonomy of techniques for visualizing this complex type of data. The taxonomy was generated from a systematic analysis of 104 papers collected from different areas of visualization research and accompanied by a companion website to interactively filter publications at different intersections of the categories in our taxonomy. Moreover, we discuss tasks specific to DMVNs along with examples from different domains. We also provide an overview of common application areas and the evaluation methods used in these published techniques.

Dynamic multivariate data is becoming ubiquitous across diverse domains, and thus the demand for efficient visualizations is growing. While node-link diagrams dominated the field initially, organized layouts like matrices and list views have gained more popularity recently. Designing complex glyphs to embed on visual elements of the network or making use of coordinated views to create separate visualizations still remains the two most popular techniques for encoding attributes. We believe this review makes it easy for visualization practitioners to compare techniques by using our taxonomy and for choosing an appropriate technique for their tasks. We also hope this review will help researchers to identify future directions and areas that require further research.

Acknowledgments

This research was supported by the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility under Contract DE-AC02-06CH11357. We thank Laura Wolf for her help in editing the paper. We also acknowledge the anonymous reviewers for their time and valuable comments on improving the paper.

References

[AAK*14] ARCHAMBAULT D., ABELLO J., KENNEDY J., KOBOUROV S., MA K.-L., MIKSCH S., MUELDER C., TELEA A. C.: Temporal multivariate networks. In *Multivariate Network Visualization*. Springer, 2014, pp. 151–174. 2, 3, 13, 15

[AB14] ARENDT D. L., BLAHA L. M.: Sven: Informative visual representation of complex dynamic structure. *arXiv preprint arXiv:1412.6706* (2014). 10, 11

[ABPdO12] ALENCAR A. B., BÖRNER K., PAULOVICH F. V., DE OLIVEIRA M. C. F.: Time-aware visualization of document collections. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (2012), pp. 997–1004. 7

[ADLM18] ARLEO A., DIDIMO W., LIOTTA G., MONTECCHIANI F.: A distributed multilevel force-directed algorithm. *IEEE Transactions on Parallel and Distributed Systems 30*, 4 (2018), 754–765. 16

[AFH*09] AHMED A., FU X., HONG S.-H., NGUYEN Q. H., XU K.: Visual analysis of history of world cup: A dynamic network with dynamic hierarchy and geographic clustering. In *Visual Information Communication*. Springer, 2009, pp. 25–39. 8, 9

[AHSS13] ABELLO J., HADLAK S., SCHUMANN H., SCHULZ H.-J.: A modular degree-of-interest specification for the visual analysis of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (2013), 337–350. 7

[ALHW20] ABDELAAL M., LHUILLIER A., HLAWATSCH M., WEISKOPF D.: Time-aligned edge plots for dynamic graph visualization. In *2020 24th International Conference Information Visualisation (IV)* (2020), IEEE, pp. 248–257. 12

[AP12] ARCHAMBAULT D., PURCHASE H. C.: The mental map and memorability in dynamic graphs. In *2012 IEEE Pacific Visualization Symposium* (2012), IEEE, pp. 89–96. 13, 14

[AP13] ARCHAMBAULT D., PURCHASE H. C.: Mental map preservation helps user orientation in dynamic graphs. In *International symposium on graph drawing* (2013), Springer, pp. 475–486. 14

[APBG19] ALEXANDRU C. V., PROKSCH S., BEHNAMGHADER P., GALL H. C.: Evo-clocks: Software evolution at a glance. In *2019 Working Conference on Software Visualization (VISSOFT)* (2019), IEEE, pp. 12–22. 11, 12

[APP10] ARCHAMBAULT D., PURCHASE H., PINAUD B.: Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE transactions on visualization and computer graphics 17*, 4 (2010), 539–552. 15

[APS13] AHN J.-W., PLAISANT C., SHNEIDERMAN B.: A task taxonomy for network evolution analysis. *IEEE transactions on visualization and computer graphics 20*, 3 (2013), 365–376. 3, 4

[ATMS*11] AHN J.-W., TAIEB-MAIMON M., SOPAN A., PLAISANT C., SHNEIDERMAN B.: Temporal visualization of social network dynamics: Prototypes for nation of neighbors. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction* (2011), Springer, pp. 309–316. 7, 12

[BBD09] BECK F., BURCH M., DIEHL S.: Towards an aesthetic dimensions framework for dynamic graph visualisations. In *2009 13th international conference information visualisation* (2009), IEEE, pp. 592–597. 16

[BBDW17] BECK F., BURCH M., DIEHL S., WEISKOPF D.: A taxonomy and survey of dynamic graph visualization. In *Computer graphics forum* (2017), vol. 36, Wiley Online Library, pp. 133–159. 2, 5

[BBS*20] BOZ H. A., BAHRAMI M., SUHARA Y., BOZKAYA B., BALCISOY S.: An exploratory visual analytics tool for multivariate dynamic networks. 3, 8

[BCD*10] BEZERIANOS A., CHEVALIER F., DRAGICEVIC P., ELMQVIST N., FEKETE J.-D.: Graphdice: A system for exploring multivariate social networks. In *Computer graphics forum* (2010), vol. 29, Wiley Online Library, pp. 863–872. 12

[BD08] BURCH M., DIEHL S.: Timeradartrees: Visualizing dynamic compound digraphs. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 823–830. 9, 10, 13

[BdM06] BENDER-DEMOLL S., MCFARLAND D. A.: The art and science of dynamic network visualization. *Journal of Social Structure 7*, 2 (2006), 1–38. 12

[BHRD*15] BACH B., HENRY-RICHE N., DWYER T., MADHYASTHA T., FEKETE J.-D., GRABOWSKI T.: Small multipiles: Piling time to explore temporal patterns in dynamic networks. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 31–40. 13, 14

[BHW11] BURCH M., HOFERLIN M., WEISKOPF D.: Layered timeradartrees. In *2011 15th International Conference on Information Visualisation* (2011), IEEE, pp. 18–25. 9

[BKH*16] BACH B., KERRACHER N., HALL K. W., CARPENDALE S., KENNEDY J., HENRY RICHE N.: Telling stories about dynamic networks with graph comics. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), pp. 3670–3682. 8

[BMGK08] BARSKY A., MUNZNER T., GARDY J., KINCAID R.: Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE transactions on visualization and computer graphics 14*, 6 (2008), 1253–1260. 8, 13

[BMR*12] BURCH M., MÜLLER C., REINA G., SCHMAUDER H., GREIS M., WEISKOPF D.: Visualizing dynamic call graphs. In *VMV* (2012), pp. 207–214. 13

[BN11] BRANDES U., NICK B.: Asymmetric relations in longitudinal social networks. *IEEE transactions on visualization and computer graphics 17*, 12 (2011), 2283–2290. 12

[BPF13] BACH B., PIETRIGA E., FEKETE J.-D.: Graphdiaries: Animated transitions andtemporal navigation for dynamic networks. *IEEE transactions on visualization and computer graphics 20*, 5 (2013), 740–754. 3, 14

[BPF14] BACH B., PIETRIGA E., FEKETE J.-D.: Visualizing dynamic networks with matrix cubes. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (2014), pp. 877–886. 9, 10

[BSW13] BURCH M., SCHMIDT B., WEISKOPF D.: A matrix-based visualization for exploring dynamic compound digraphs. In *2013 17th International Conference on Information Visualisation* (2013), IEEE, pp. 66–73. 12

[BTBC*21] BURCH M., TEN BRINKE K. B., CASTELLA A., PETERS G. K. S., SHTERIYANOV V., VLASVINKEL R.: Dynamic graph exploration by interactively linked node-link diagrams and matrix visualizations. *Visual Computing for Industry, Biomedicine, and Art 4*, 1 (2021), 1–14. 13

[Bur17] BURCH M.: The dynamic graph wall: visualizing evolving graphs with multiple visual metaphors. *Journal of Visualization 20*, 3 (2017), 461–469. 13

[BVB*11] BURCH M., VEHLOW C., BECK F., DIEHL S., WEISKOPF D.: Parallel edge splatting for scalable dynamic graph visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2344–2353. 10

[Che06] CHEN C.: Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for information Science and Technology 57*, 3 (2006), 359–377. 12

[CKS*21] CLYDE A., KALE B., SUN M., PAPKA M., RAMANATHAN A., STEVENS R.: Scaffold embeddings: Learning the structure spanned by chemical fragments, scaffolds and compounds. In *Workshop on Learning Meaningful Representation of Life* (2021). 7, 8

[CM11] CORREA C. D., MA K.-L.: Visualizing social networks. In *Social network data analytics*. Springer, 2011, pp. 307–326. 12

[CSJ*20] CAKMAK E., SCHLEGEL U., JÄCKLE D., KEIM D., SCHRECK T.: Multiscale snapshots: Visual analysis of temporal summaries in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 517–527. 14

[CSW21] CHEONG S.-H., SI Y.-W., WONG R. K.: Online force-directed algorithms for visualization of dynamic graphs. *Information Sciences 556* (2021), 223–255. 7, 14

[CWL*10] CUI W., WU Y., LIU S., WEI F., ZHOU M. X., QU H.: Context preserving dynamic word cloud visualization. In *2010 IEEE Pacific Visualization Symposium (PacificVis)* (2010), IEEE, pp. 121–128. 8, 9

[DGK00] DIEHL S., GÖRG C., KERREN A.: Foresighted graphlayout. 6

[DPF16] DANG T. N., PENDAR N., FORBES A. G.: Timearcs: Visualizing fluctuations in dynamic networks. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 61–69. 10

[FAM21] FILIPOV V., ARLEO A., MIKSCH S.: Exploratory user study on graph temporal encodings. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)* (2021), IEEE, pp. 131–135. 13, 14

[FAS*20] FISCHER M. T., ARYA D., STREEB D., SEEBACHER D., KEIM D. A., WORRING M.: Visual analytics for temporal hypergraph model exploration. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 550–560. 14

[FH21] FU D., HE J.: Dppin: A biological repository of dynamic protein-protein interaction network data. *arXiv preprint arXiv:2107.02168* (2021). 13

[FM16] FEDERICO P., MIKSCH S.: Evaluation of two interaction techniques for visualization of dynamic graphs. In *International Symposium on Graph Drawing and Network Visualization* (2016), Springer, pp. 557–571. 13, 14

[FMW*21] FU K., MAO T., WANG Y., LIN D., ZHANG Y., SUN X.: Dyegovis: Visual exploration of dynamic ego-network evolution. *Applied Sciences 11*, 5 (2021), 2399. 3, 8

[FSB*13] FIAUX P., SUN M., BRADEL L., NORTH C., RAMAKRISHNAN N., ENDERT A.: Bixplorer: Visual analytics with biclusters. *Computer 46*, 8 (2013), 90–94. 5

[FT08] FRISHMAN Y., TAL A.: Online dynamic graph drawing. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (2008), 727–740. 6, 7, 14

[FWSL11] FENG K.-C., WANG C., SHEN H.-W., LEE T.-Y.: Coherent time-varying graph drawing with multifocus+ context interaction. *IEEE Transactions on Visualization and Computer Graphics 18*, 8 (2011), 1330–1342. 7, 14

[GAW*11] GLEICHER M., ALBERS D., WALKER R., JUSUFI I., HANSEN C. D., ROBERTS J. C.: Visual comparison for information visualization. *Information Visualization 10*, 4 (2011), 289–309. 2

[GGK*11] GOVE R., GRAMSKY N., KIRBY R., SEFER E., SOPAN A., DUNNE C., SHNEIDERMAN B., TAIEB-MAIMON M.: Netvisia: Heat map & matrix visualization of dynamic social network statistics & content. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing* (2011), IEEE, pp. 19–26. 3, 9, 10

[GGPPS13] GUERRA-GOMEZ J., PACK M. L., PLAISANT C., SHNEIDERMAN B.: Visualizing change over time using dynamic hierarchies: Treeversity2 and the stemview. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2566–2575. 4

[GOB*10] GEHLENBORG N., O'DONOGHUE S. I., BALIGA N. S., GOESMANN A., HIBBS M. A., KITANO H., KOHLBACHER O., NEUWEGER H., SCHNEIDER R., TENENBAUM D., ET AL.: Visualization of omics data for systems biology. *Nature methods 7*, 3 (2010), S56–S68. 13

[GV20] GUPTA U., VLACHOS D. G.: Reaction network viewer (renview): An open-source framework for reaction path visualization of chemical reaction systems. *SoftwareX 11* (2020), 100442. 13

[GZD*15] GOHNERT T., ZIEBARTH S., DETJEN H., HECKING T., HOPPE H. U.: 3d dynnetvis—a 3d visualization technique for dynamic networks. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2015), IEEE, pp. 737–740. 8, 9

[HEF*13] HURTER C., ERSOY O., FABRIKANT S. I., KLEIN T. R., TELEA A. C.: Bundled visualization of dynamicgraph and trail data. *IEEE transactions on visualization and computer graphics 20*, 8 (2013), 1141–1157. 7, 14

[HL12] HARRISON L., LU A.: The future of security visualization: Lessons from network visualization. *IEEE Network 26*, 6 (2012), 6–11. 12

[HMHU13] HAYASHI A., MATSUBAYASHI T., HOSHIDE T., UCHIYAMA T.: Initial positioning method for online and real-time dynamic graph drawing of time varying data. In *2013 17th International Conference on Information Visualisation* (2013), IEEE, pp. 435–444. 6, 7, 14

[HS12] HOLME P., SARAMÄKI J.: Temporal networks. *Physics reports 519*, 3 (2012), 97–125. 2

[HSCW13] HADLAK S., SCHUMANN H., CAP C. H., WOLLENBERG T.: Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2267–2276. 11, 15

[HSS15] HADLAK S., SCHUMANN H., SCHULZ H.-J.: A survey of multi-faceted graph visualization. In *EuroVis (STARs)* (2015), pp. 1–20. 2

[HZL*16] HE Q., ZHU M., LU B., LIU H., SHEN Q.: Mena: Visual analysis of multivariate egocentric network evolution. In *2016 International Conference on Virtual Reality and Visualization (ICVRV)* (2016), IEEE, pp. 488–496. 3, 9

[IA12] ITOH M., AKAISHI M.: Visualization for changes in relationships between historical figures in chronicles. In *2012 16th International Conference on Information Visualisation* (2012), IEEE, pp. 283–290. 8, 9

[JE12] JAVED W., ELMQVIST N.: Exploring the design space of composite visualization. In *2012 ieee pacific visualization symposium* (2012), IEEE, pp. 1–8. 2, 5

[JSS*13] JOHN M., SCHULZ H.-J., SCHUMANN H., UHRMACHER A. M., UNGER A.: Constructing and visualizing chemical reaction networks from pi-calculus models. *Formal Aspects of Computing 25*, 5 (2013), 723–742. 10

[KAW*14] KO S., AFZAL S., WALTON S., YANG Y., CHAE J., MALIK A., JANG Y., CHEN M., EBERT D.: Analyzing high-dimensional multivariate network links with integrated anomaly detection, highlighting and exploration. In *2014 IEEE conference on visual analytics science and technology (VAST)* (2014), IEEE, pp. 83–92. 12

[Kee06] KEEL P. E.: Collaborative visual analytics: Inferring from the spatial organization and collaborative use of information. In *2006 IEEE Symposium On Visual Analytics Science And Technology* (2006), IEEE, pp. 137–144. 16

[KKC14] KERRACHER N., KENNEDY J., CHALMERS K.: The design space of temporal graph visualisation. In *EuroVis (Short Papers)* (2014). 2

[KKC15] KERRACHER N., KENNEDY J., CHALMERS K.: A task taxonomy for temporal graph visualisation. *IEEE transactions on visualization and computer graphics 21*, 10 (2015), 1160–1172. 2, 3

[KLRZ94] KIMELMAN D., LEBAN B., ROTH T., ZERNIK D.: Reduction of visual complexity in dynamic graphs. In *International Symposium on Graph Drawing* (1994), Springer, pp. 218–225. 13

[LAN19] LEE A., ARCHAMBAULT D., NACENTA M.: Dynamic network plaid: A tool for the analysis of dynamic networks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–14. 13

[LAN20] LEE A., ARCHAMBAULT D., NACENTA M. A.: The effectiveness of interactive visualization techniques for time navigation of dynamic graphs on large displays. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 528–538. 13, 14

[LHS*15] LIU Q., HU Y., SHI L., MU X., ZHANG Y., TANG J.: Egonetcloud: Event-based egocentric dynamic network visualization. In *2015*

IEEE Conference on Visual Analytics Science and Technology (VAST) (2015), IEEE, pp. 65–72. 3, 8, 9

[LL16] LI T., LIAO Q.: Dynamic networks analysis and visualization through spatiotemporal link segmentation. In *2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)* (2016), IEEE, pp. 209–214. 11

[LTPR17] LINHARES C. D., TRAVENÇOLO B. A., PAIVA J. G. S., ROCHA L. E.: Dynetvis: a system for visualization of dynamic networks. In *Proceedings of the symposium on applied computing* (2017), pp. 187–194. 10

[LWB18] LAW P.-M., WU Y., BASOLE R. C.: Segue: Overviewing evolution patterns of egocentric networks by interactive construction of spatial layouts. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2018), IEEE, pp. 72–83. 7

[LZH*17] LU B., ZHU M., HE Q., LI M., JIA R.: Tmnvis: Visual analysis of evolution in temporal multivariate network at multiple granularities. *Journal of Visual Languages & Computing 43* (2017), 30–41. 3, 12, 13

[MELS95] MISUE K., EADES P., LAI W., SUGIYAMA K.: Layout adjustment and the mental map. *Journal of Visual Languages & Computing 6*, 2 (1995), 183–210. 6

[MGL06] MUELLER C., GREGOR D. P., LUMSDAINE A.: Distributed force-directed graph layout and visualization. In *EGPGV@ EuroVis/EGVE* (2006), pp. 83–90. 16

[MMBd05] MOODY J., MCFARLAND D., BENDER-DEMOLL S.: Dynamic network visualization. *American journal of sociology 110*, 4 (2005), 1206–1241. 12

[NHC*20] NGUYEN N., HASS J., CHEN Y., LI J., SILL A., DANG T.: Radarviewer: Visualizing the dynamics of multivariate data. In *Practice and Experience in Advanced Research Computing*. 2020, pp. 555–556. 10, 11

[NMSL19] NOBRE C., MEYER M., STREIT M., LEX A.: The state of the art in visualizing multivariate networks. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 807–832. 2, 3, 4, 6

[NWHL20] NOBRE C., WOOTTON D., HARRISON L., LEX A.: Evaluating multivariate network visualization techniques using a validated design and crowdsourcing approach. In *Proceedings of the 2020 CHI conference on human factors in computing systems* (2020), pp. 1–12. 13, 16

[OJK18] OKOE M., JIANU R., KOBOUROV S.: Node-link or adjacency matrices: Old question, new insights. *IEEE transactions on visualization and computer graphics 25*, 10 (2018), 2940–2952. 14, 16

[PGFL05] PINZGER M., GALL H., FISCHER M., LANZA M.: Visualizing multiple evolution metrics. In *Proceedings of the 2005 ACM symposium on Software visualization* (2005), pp. 67–75. 11

[PND*20] PHAM V., NGUYEN V. N., DANG T., TURKAY C., VROTSOU K.: Dualnetview: Dual views for visualizing the dynamics of networks. In *EuroVA@ Eurographics/EuroVis* (2020), pp. 25–29. 8, 9

[PTLZ18] PENG D., TIAN W., LU B., ZHU M.: Dmnevis: a novel visual approach to explore evolution of dynamic multivariate network. In *2018 IEEE international conference on systems, man, and cybernetics (SMC)* (2018), IEEE, pp. 4304–4311. 3, 10, 13

[PW16] PORTENOY J., WEST J. D.: Dynamic visualization of citation networks showing the influence of scholarly fields over time. In *International Workshop on Semantic, Analytics, Visualization* (2016), Springer, pp. 147–151. 7

[PXN*19] PERRIE J., XIE J., NAYEBI M., FOKAEFS M., LYONS K., STROULIA E.: City on the river: visualizing temporal collaboration. In *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering* (2019), pp. 82–91. 10

[RM14] RUFIANGE S., MELANÇON G.: Animatrix: A matrix-based visualization of software evolution. In *2014 second IEEE working conference on software visualization* (2014), IEEE, pp. 137–146. 7, 13

[RTJ*11] REDA K., TANTIPATHANANANDH C., JOHNSON A., LEIGH J., BERGER-WOLF T.: Visualizing the evolution of community structures in dynamic social networks. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1061–1070. 10

[RUK*10] ROHRSCHNEIDER M., ULLRICH A., KERREN A., STADLER P. F., SCHEUERMANN G.: Visual network analysis of dynamic metabolic pathways. In *International Symposium on Visual Computing* (2010), Springer, pp. 316–327. 13

[RWDHP19] RIEGLER V., WANG L., DOPPLER-HAIDER J., POHL M.: Evaluation of a novel visualization for dynamic social networks. In *Proceedings of the 12th International Symposium on Visual Information Communication and Interaction* (2019), pp. 1–8. 10

[SDMW09] SCHREIBER F., DWYER T., MARRIOTT K., WYBROW M.: A generic algorithm for layout of biological networks. *BMC bioinformatics 10*, 1 (2009), 1–12. 13

[SGB19] SOBRAL T., GALVÃO T., BORGES J.: Visualization of urban mobility data from intelligent transportation systems. *Sensors 19*, 2 (2019), 332. 12

[SKAS22] SHAIKH A. R., KOOP D., ALHOORI H., SUN M.: Toward systematic design considerations of organizing multiple views. In *2022 IEEE Visualization and Visual Analytics (VIS)* (2022), IEEE, pp. 105–109. 5

[SKZ*19] SUN M., KOOP D., ZHAO J., NORTH C., RAMAKRISHNAN N.: Interactive bicluster aggregation in bipartite graphs. In *2019 IEEE Visualization Conference (VIS)* (2019), IEEE, pp. 246–250. 10

[SLC*17] SHAO M., LI J., CHEN F., HUANG H., ZHANG S., CHEN X.: An efficient approach to event detection and forecasting in dynamic multivariate social media networks. In *Proceedings of the 26th International Conference on World Wide Web* (2017), pp. 1631–1639. 9, 14, 15

[SMNR15] SUN M., MI P., NORTH C., RAMAKRISHNAN N.: Biset: Semantic edge bundling with biclusters for sensemaking. *IEEE transactions on visualization and computer graphics 22*, 1 (2015), 310–319. 7

[SNK*21] SUN M., NAMBURI A., KOOP D., ZHAO J., LI T., CHUNG H.: Towards systematic design considerations for visualizing cross-view data relationships. *IEEE Transactions on Visualization and Computer Graphics 28*, 12 (2021), 4741–4756. 5

[SNR14] SUN M., NORTH C., RAMAKRISHNAN N.: A five-level design framework for bicluster visualizations. *IEEE transactions on visualization and computer graphics 20*, 12 (2014), 1713–1722. 5

[SRG*19] STOIBER C., RIND A., GRASSINGER F., GUTOUNIG R., GOLDGRUBER E., SEDLMAIR M., EMRICH Š., AIGNER W.: netflower: Dynamic network visualization for data journalists. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 699–711. 10, 14

[SSAZ21] SUN M., SHAIKH A. R., ALHOORI H., ZHAO J.: Sightbi: Exploring cross-view data relationships with biclusters. *IEEE Transactions on Visualization and Computer Graphics 28*, 1 (2021), 54–64. 5

[STSB16] SCHNEIDER T., TYMCHUK Y., SALGADO R., BERGEL A.: Cuboidmatrix: exploring dynamic structural connections in software components using space-time cube. In *2016 IEEE working conference on software visualization (VISSOFT)* (2016), IEEE, pp. 116–125. 9, 10

[SWKA19] SORGER J., WALDNER M., KNECHT W., ARLEO A.: Immersive analytics of large dynamic networks via overview and detail navigation. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)* (2019), IEEE, pp. 144–1447. 6, 14

[SWS10] STEIN K., WEGENER R., SCHLIEDER C.: Pixel-oriented visualization of change in social networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining* (2010), IEEE, pp. 233–240. 12

[Tuf06] TUFTE E. R.: *Beautiful evidence*. Graphis Pr, 2006. 12

[TV08] TELEA A., VOINEA L.: An interactive reverse engineering environment for large-scale c++ code. In *Proceedings of the 4th ACM symposium on Software visualization* (2008), pp. 67–76. 13

[VABK20] VOGOGIAS A., ARCHAMBAULT D., BACH B., KENNEDY J.: Visual encodings for networks with multiple edge types. In *Proceedings of the International Conference on Advanced Visual Interfaces* (2020), pp. 1–9. 13

[VBP*19] VALDIVIA P., BUONO P., PLAISANT C., DUFOURNAUD N., FEKETE J.-D.: Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization. *IEEE transactions on visualization and computer graphics 27*, 1 (2019), 1–13. 10

[VBSW13] VEHLOW C., BURCH M., SCHMAUDER H., WEISKOPF D.: Radial layered matrix visualization of dynamic graphs. In *2013 17th International Conference on Information Visualisation* (2013), IEEE, pp. 51–58. 9

[VBW15] VEHLOW C., BECK F., WEISKOPF D.: Visualizing dynamic hierarchies in graph sequences. *IEEE transactions on visualization and computer graphics 22*, 10 (2015), 2343–2357. 3, 9, 10

[vdEHBvW13] VAN DEN ELZEN S., HOLTEN D., BLAAS J., VAN WIJK J. J.: Dynamic network visualization withextended massive sequence views. *IEEE transactions on visualization and computer graphics 20*, 8 (2013), 1087–1099. 10

[VLBR*15] VON LANDESBERGER T., BRODKORB F., ROSKOSCH P., ANDRIENKO N., ANDRIENKO G., KERREN A.: Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE transactions on visualization and computer graphics 22*, 1 (2015), 11–20. 8, 9, 15

[VLKS*11] VON LANDESBERGER T., KUIJPER A., SCHRECK T., KOHLHAMMER J., VAN WIJK J. J., FEKETE J.-D., FELLNER D. W.: Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer graphics forum* (2011), vol. 30, Wiley Online Library, pp. 1719–1749. 2, 5

[WCB16] WITTENHAGEN M., CHEREK C., BORCHERS J.: Chronicler: Interactive exploration of source code history. In *Proceedings of the 2016 CHI conference on human factors in computing systems* (2016), pp. 3522–3532. 14

[WCPB12] WIDANAGAMAACHCHI W., CHRISTENSEN C., PASCUCCI V., BREMER P.-T.: Interactive exploration of large-scale time-varying data using dynamic tracking graphs. In *IEEE symposium on large data analysis and visualization (LDAV)* (2012), IEEE, pp. 9–17. 10

[WEF*14] WYBROW M., ELMQVIST N., FEKETE J.-D., LANDESBERGER T. V., WIJK J. J. V., ZIMMER B.: Interaction in the visualization of multivariate networks. In *Multivariate Network Visualization*. Springer, 2014, pp. 97–125. 2

[WF*94] WASSERMAN S., FAUST K., ET AL.: Social network analysis: Methods and applications. 12

[WPPW14] WANG J., PENG X., PENG W., WU F.-X.: Dynamic protein interaction network construction and applications. *Proteomics 14*, 4-5 (2014), 338–352. 13

[WPZ*15] WU Y., PITIPORNVIVAT N., ZHAO J., YANG S., HUANG G., QU H.: egoslider: Visual analysis of egocentric network evolution. *IEEE transactions on visualization and computer graphics 22*, 1 (2015), 260–269. 8, 10, 14

[WSL*18] WANG Y., SHI C., LI L., TONG H., QU H.: Visualizing research impact through citation data. *ACM Transactions on Interactive Intelligent Systems (TiiS) 8*, 1 (2018), 1–24. 12

[WSM*18] WU H., SUN M., MI P., TATTI N., NORTH C., RAMAKRISHNAN N.: Interactive discovery of coordinated relationship chains with maximum entropy models. *ACM Transactions on Knowledge Discovery from Data (TKDD) 12*, 1 (2018), 1–34. 10

[XTYL18] XU J., TAO Y., YAN Y., LIN H.: Exploring evolution of dynamic networks via diachronic node embeddings. *IEEE transactions on visualization and computer graphics 26*, 7 (2018), 2387–2402. 9

[YDK*18] YOGHOURDJIAN V., DWYER T., KLEIN K., MARRIOTT K., WYBROW M.: Graph thumbnails: Identifying and comparing multiple graphs at a glance. *IEEE Transactions on Visualization and Computer Graphics 24*, 12 (2018), 3081–3095. 4

[YEL10] YI J. S., ELMQVIST N., LEE S.: Timematrix: Analyzing temporal social networks using interactive matrix-based visualizations. *Intl. Journal of Human–Computer Interaction 26*, 11-12 (2010), 1031–1051. 12

[ZCY09] ZHOU Y., CHENG H., YU J. X.: Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment 2*, 1 (2009), 718–729. 9

[ZGC*16] ZHAO J., GLUECK M., CHEVALIER F., WU Y., KHAN A.: Egocentric analysis of dynamic networks with egolines. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), pp. 5003–5014. 3, 9, 10, 14

[ZSC*21] ZHAO J., SUN M., CHIU P., CHEN F., LIEW B.: Know-what and know-who: Document searching and exploration using topic-based two-mode networks. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)* (2021), IEEE, pp. 86–90. 12

[ZSCC19] ZHAO J., SUN M., CHEN F., CHIU P.: Missbin: Visual analysis of missing links in bipartite networks. In *2019 IEEE Visualization Conference (VIS)* (2019), IEEE, pp. 71–75. 9

[ZSCC20] ZHAO J., SUN M., CHEN F., CHIU P.: Understanding missing links in bipartite networks with missbin. *IEEE Transactions on Visualization and Computer Graphics 28*, 6 (2020), 2457–2469. 7