

Multiscale Visual Analysis of Dynamic Networks

**Dissertation zur Erlangung des
akademischen Grades eines Doktors der
Naturwissenschaften (Dr.rer.nat.)**

vorgelegt von

Eren Cakmak

an der

Universität
Konstanz



Mathematisch-Naturwissenschaftliche Sektion
Informatik und Informationswissenschaft

Konstanz, 2022

Tag der mündlichen Prüfung: 16. Dezember 2022

1. Referent/Referentin: Prof. Dr. Daniel A. Keim

2. Referent/Referentin: Prof. Dr. Tobias Schreck

Abstract

Networks are a universal language for modeling the underlying structure of real-world systems, such as computer networks, social networks, or financial networks. Many of these modeled real-world systems are dynamic, meaning the relationships between the entities change over time. A central goal in dynamic (temporal) network analysis is to discover similar network structures and retrace structural changes over time. However, visually analyzing dynamic networks remains challenging due to large-scale data often evolving over long periods.

This thesis presents studies for the multiscale visual analysis of dynamic networks. The presented studies introduce multiscale dynamic network visualizations for identifying, comparing, tracing, and interpreting similar network structures over time. The proposed visualizations combine automated analysis methods with interactive visualizations to reveal evolving network structures across multiple abstraction scales (multiscale analysis). The presented multiscale visualizations scale to large-scale dynamic networks and enable analysts to relate high-level overviews with low-level details to reveal structural changes and similar network structures over time. The presented studies are showcased by prototype implementations using real-world datasets and are validated with domain experts, quantitative evaluations, and use cases. Moreover, the thesis systematically discusses the benefits and limitations of the presented studies and outlines future research perspectives.

Zusammenfassung

Mithilfe von Netzwerken lässt sich die zugrunde liegende Struktur von realen Systemen wie Computernetzwerken, sozialen Netzwerken oder Finanznetzwerken modellieren. Viele dieser modellierten realen Systeme sind dynamisch. Das bedeutet, dass die Beziehungen zwischen den Objekten sich im Laufe der Zeit ändern. Ein zentrales Ziel der dynamischen (zeitlichen) Netzwerkanalyse ist es, ähnliche Netzwerkstrukturen zu entdecken und strukturelle Veränderungen über die Zeit zu verstehen. Die visuelle Analyse von dynamischen Netzwerken ist jedoch, aufgrund der sich oft über lange Zeiträume entwickelnden Datenmengen, eine Herausforderung.

In dieser Thesis werden Arbeiten für die multiskalige visuelle Analyse von dynamischen Netzwerken präsentiert. Die vorgestellten multiskaligen dynamischen Netzwerkvisualisierungen ermöglichen es, ähnliche Netzwerkstrukturen über die Zeit zu identifizieren, zu vergleichen, nachzuverfolgen und zu interpretieren. Die Visualisierungen kombinieren automatisierte Analysemethoden mit interaktiven Visualisierungen, um sich ändernde Netzwerkstrukturen über mehrere Abstraktionsstufen hinweg darzustellen (Multiskalenanalyse). Die vorgestellten multiskaligen Visualisierungen skalieren auf große dynamische Netzwerke und ermöglichen es Analysten, high-level Übersichten und low-level Detailansichten zu verknüpfen, um strukturelle Veränderungen und ähnliche Netzwerkstrukturen über die Zeit sichtbar zu machen. Die präsentierten Arbeiten werden mithilfe von Prototyp-Implementierungen unter Verwendung realer Datensätzen vorgestellt und mit Domänenexperten, quantitativen Auswertungen und Anwendungsbeispielen validiert. Des Weiteren diskutiert die Thesis systematisch die Vorteile und Schwächen der vorgestellten Arbeiten und erläutert zukünftige Forschungsperspektiven.

Acknowledgement

First and foremost, I want to thank my supervisor, Daniel Keim, for giving me the opportunity to work in his innovative research group. I also like to thank my secondary advisor, Tobias Schreck, who supported my research early on with fruitful collaborations and discussions. Moreover, I want to thank all my colleagues in the DBVIS research group. I like to give a special thanks to Juri Buchmüller, Dominik Jäckle, Johannes Fuchs, Udo Schlegel, Sabine Kuhr, Matthew Sharinghousen, Benjamin Bäumle, and the outstanding DBVIS support team. It was a great and inspiring time working with all of you.

Finally, I want to thank my family and friends. Thank you for being part of my life.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research Objectives	3
1.3	Thesis Outline & Contributions	4
1.4	Terminology	7
1.5	Publications	8
2	Multiscale Visualization: A Structured Literature Analysis	15
2.1	Introduction	16
2.2	Background	17
2.3	Methodology	20
2.4	Multiscale Visualization Taxonomy	23
2.5	Research Challenges	35
2.6	Discussion	37
2.7	Conclusion	37
3	Visual Abstraction of Spatio-Temporal Networks	39
3.1	Introduction	40
3.2	Related Work & Application Background	41
3.3	Visual Design	48
3.4	Evaluation	55
3.5	Spatio-Temporal Clustering Benchmark	60
3.6	Discussion	66
3.7	Conclusion	67
4	Pixel-Based Visual Analysis of Dynamic Networks	69
4.1	Introduction	70
4.2	Related Work	71
4.3	Dynamic Network to Pixel-Based Visualization	74
4.4	Visual Interpretation	79
4.5	Evaluation	83
4.6	Discussion	88
4.7	Conclusion	90

5 Motif-Based Dynamic Network Visualization	91
5.1 Introduction	92
5.2 Related Work	93
5.3 Structure-Based Visual Abstraction	98
5.4 Motif-Based Visual Analysis	101
5.5 Visual Patterns	104
5.6 Evaluation	106
5.7 Discussion	114
5.8 Conclusion	116
6 Multiscale Visual Analysis of Temporal Summaries in Dynamic Networks	117
6.1 Introduction	118
6.2 Related Work & Application Background	119
6.3 Multiscale Snapshots	124
6.4 Evaluation	133
6.5 Discussion	140
6.6 Conclusion	142
7 Conclusions and Future Work	143
7.1 Conclusion	143
7.2 Future Perspectives	145
List of Figures	149
List of Tables	151
Bibliography	153

Introduction

1.1 Motivation

Networks are useful for modeling and understanding relationships of real-world systems, such as financial networks, ecological networks, computer networks, neural networks, or social networks. Analyzing such networks (graphs) is crucial for understanding relationships (edges) between real-world entities (nodes). For example, in biology, physics, or computer science, typical network analysis tasks are to provide insight into node connectivities, motifs, or communities. Moreover, many real-world networks are dynamic (temporal), meaning that the modeled systems' structure changes over time [149]. Analyzing such dynamic networks enable analysts to understand evolving relationships, processes, and changes in real-world systems. For instance, analysts use dynamic networks to study information diffusion in social networks [173], analyze communication networks [96], examine protein interactions in biological systems [135], or study viruses' spreading in societies [72].

A central task in dynamic network analysis is understanding where in the network, when in time, and what has happened [24]. For example, gaining insight into stable, growing, or shrinking sub-networks and clusters helps to understand structural changes in dynamic networks. Yet, the complexity of dynamic network analysis rises with a growing number of nodes, edges, and especially time steps due to the increasing amount of temporal changes. Automatically quantifying every structural change in a dynamic network remains difficult since no algorithm or metric can capture all evolving network properties. Moreover, examining aggregated network metrics or the result of fully automated algorithms only helps answer precise hypotheses. In many instances, however, analysts first need to explore the dynamic network data to generate such hypotheses. Thus, analysts frequently utilize interactive visualizations to explore and reveal unknown patterns in dynamic networks [103]. Visually exploring datasets to discover such unknown patterns and generate new hypotheses is known as exploratory data analysis [294]. However, visualizing dynamic networks in a readable and scalable manner remains challenging due to the underlying large-scale data, making the visual analysis of dynamic networks a non-trivial task.

Previous dynamic network visualizations, therefore, often utilize abstraction methods to present relevant summaries of the evolving data. For example, visualization approaches utilize dimensionality reduction methods to provide an overview of higher-level network structures over time [103]. However, selecting useful abstraction methods for large-scale dynamic networks remains challenging for multiple reasons. First, the abstraction method's usefulness depends on many factors, such as the application domain, the user task, the network size, and the frequency of changes. Moreover, such abstraction methods must consider and integrate two important data aspects, the relational network structure and the temporal dimension. There is a considerable trade-off between visualizing the detailed network structure for each time step, and presenting the evolving networks over time [36]. Finally, selecting an appropriate abstraction scale poses another challenge for both the relational and temporal data aspects. For instance, a fine-grained temporal aggregation scale will produce numerous summaries with little information being unable to provide an overview of the dynamic network. In contrast, a coarse-grained temporal aggregation will produce only a few summaries, making it impossible to retrace network changes over time [36]. The same is true for the relational aspect since the network structure can be abstracted and analyzed at numerous scales (e.g., nodes, paths, and clusters) [196]. Thus, selecting an appropriate abstraction scale is crucial for visually exploring and gaining insight into large-scale dynamic networks. Yet, to date, many dynamic network visualizations lack methods for visually analyzing evolving network structures at multiple abstraction scales (multiscale analysis).

This thesis proposes multiscale dynamic network visualizations to overcome these challenges. Multiscale dynamic network visualizations are one way to make dynamic networks manageable and understandable. Multiscale visualizations are useful for exploring data at multiple abstraction levels to relate high-level overviews with low-level details. Thus, the following thesis presents studies for visually analyzing relational and temporal patterns in dynamic networks at multiple abstraction scales. This thesis contributes to information visualization research by presenting novel multiscale visualizations and visual analytics approaches for dynamic networks. The presented studies combine automated analysis methods with interactive visualizations to display the large-scale dynamic network data in a readable and scalable manner. The proposed multiscale dynamic network visualizations scale to large-scale dynamic networks, produce less clutter, reveal the emergence of patterns at different abstraction scales, and helps to reveal useful abstraction methods and scales.

1.2 Research Objectives

The central thesis goal is to improve and advance the multiscale visualization of dynamic networks. Thus, this thesis addresses the general research question:

- *How can we enhance the multiscale visual analysis of dynamic networks?*

This thesis answers the research question by addressing the following research challenges. First, developing multiscale visualizations remains challenging due to the ambiguity and usage of "multiscale visualization" in information visualization research. Thereby, defining the terminology and understanding common design practices is crucial for developing multiscale dynamic network visualizations. Second, gaining an overview of a dynamic network is crucial for revealing potentially useful analysis and visualization methods, including identifying proper abstraction methods and scales. Yet, the large-scale data and the limited display space pose a significant challenge for visualizing and tracing structural changes in dynamic network visualizations. Thus, providing an uncluttered overview visualization of the temporal dimensions is necessary to identify structural changes and similar temporal states. Moreover, there is a trade-off between visualizing the detailed network structure for each time step and the evolving data over time. Third, a central challenge for exploring evolving network structures is the visualization, navigation, and interpretation of patterns. The interactive navigation across scales is essential, including the simultaneous exploration of relational and temporal data. Furthermore, it is not enough to display the data at different abstraction scales but also to enable semi-automated exploration across scales. Another challenge is visualizing the relational and temporal network structure using novel visual metaphors to reveal dynamic patterns, such as structural changes, trends, states, and outlier network structures. In this context, distinct application domains require tailored multiscale visual metaphors based on a domain-specific requirements analysis. Finally, multi-scale dynamic network visualizations must display the evolving data in a readable and scalable manner.

This thesis presents multiple studies addressing the previously described challenges. The presented studies and the developed visualization prototypes are based on real-world use cases in different application domains. The proposed visualization approaches are not limited to the presented application domain but are generalizable to other fields with similar dynamic network analysis tasks.

1.3 Thesis Outline & Contributions

The following section outlines the primary thesis contributions that address the aforementioned research question and challenges. In summary, this thesis contributes visualizations to enhance the interactive multiscale visual analysis of dynamic networks. The thesis consists of seven chapters (see Figure 1.1). Chapter 1 motivates the tackled research challenges and introduces relevant terminology. Chapter 2 surveys and provides background information about common design factors in multiscale visualizations. Next, Chapter 3 introduces a design study highlighting the benefits of multiscale visualizations in the application domain of collective animal behavior. The Chapters 4-5 present scalable pixel-based visualization techniques to provide an overview of changes in dynamic networks based on unsupervised graph embeddings and motif analysis. Chapter 6 introduces a multiscale visual analytics approach to seamlessly integrate graph analysis methods with visualization techniques to interactively analyze evolving network structures. Chapter 7 summarizes the thesis contributions and discusses open research challenges.

The following descriptions summarize each chapter's content and scientific contributions. Moreover, each chapter's project code or results are openly accessible online using the specified URLs.

Chapter 1 The introduction motivates and presents the research in this thesis. Moreover, the chapter introduces research challenges, contributions, terminology, and reused publications, including contribution clarifications.

Chapter 2 The second chapter provides an overview of design practices in multiscale visualization research. The structured literature analysis reviews and categorizes 122 multiscale visualizations to understand common design practices. The primary contributions of the chapter are: (1) a unified terminology of multiscale visualization, (2) a taxonomy of multiscale visualization design practices, (3) a collection of design considerations, and (4) the discussion of open research challenges. Researchers and practitioners can use the structured literature analysis to explore existing techniques to develop novel multiscale navigation and visualization methods. The list of the reviewed papers, paper codings, and taxonomy are accessible online at <https://multiscale-vis.dbvis.de>.

Chapter 3 The third chapter presents a multiscale visualization application that enables domain experts in collective animal behavior to explore spatio-temporal networks and group structures at multiple aggregation scales. In the interdisciplinary design study, we addressed biologists' needs and proposed various glyph

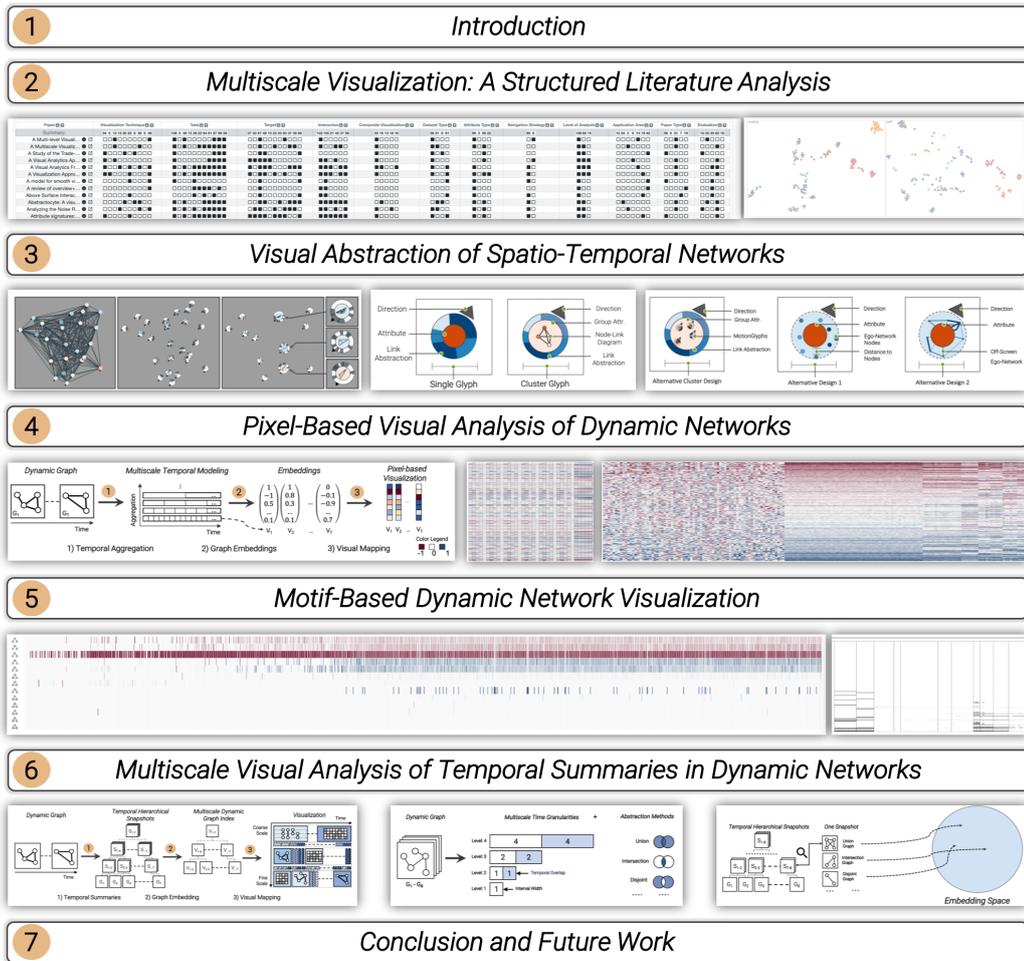


Fig. 1.1 The thesis consists of an introduction, five core chapters, and a conclusion. The figure provides an overview of all chapters.

designs to reveal similar movement behavior and emergent group properties at multiple scales. The main contributions of the chapter are: (1) a design study within the domain of collective animal behavior, (2) a glyph design for summarizing and encoding spatio-temporal networks, (3) a visualization prototype to enable domain experts to analyze local and global network properties over time, and (4) a spatio-temporal clustering benchmark for the field of collective animal behavior. The resulting glyph design enables domain experts to display and explore visual summaries of dense spatio-temporal networks in collective animal behavior. The developed prototype is online at <https://glyph.dbvis.de> and the code is accessible at <https://github.com/eren-ck/MotionGlyphs>. Moreover, spatio-temporal clustering benchmark is accessible online at <https://github.com/eren-ck/spatio-temporal-clustering-benchmark>.

Chapter 4 The fourth chapter presents *dg2pix*, a multiscale pixel-based visualization technique to explore temporal and structural properties in large-scale dynamic networks. The technique consists of multiscale temporal modeling, unsupervised graph embeddings, and dense pixel visualization. The scientific contributions are the following: (1) *dg2pix*, a time-scalable visual metaphor to highlight changes and similar temporal states in dynamic networks, (2) interpretation strategies for the visual patterns in *dg2pix*, and (3) a prototype that allows exploring dynamic networks at multiple temporal scales. Overall, *dg2pix* provides a scalable overview of temporal and structural changes in a dynamic network. The code for the developed prototype is accessible online at <https://github.com/eren-ck/dg2pix>.

Chapter 5 The fifth chapter presents two complementary scalable pixel visualizations to provide an overview of changing motif structures in large-scale dynamic networks. The pixel visualizations reveal structural changes, trends, states, and outliers in dynamic networks. The main contributions of the chapter are: (1) the usage of motif analysis to provide an overview of significant topological changes in a dynamic network, (2) the description of visual patterns and reordering strategies for both pixel visualizations, (3) and a prototype implementation with extensive use case scenarios that analyze synthetic and real-world dynamic networks. The linked pixel-based visualizations allow exploring static and dynamic network summaries to search for temporal patterns and visually analyze the underlying network structure. The code for the developed prototype is accessible online at <https://github.com/eren-ck/motif-pixel-vis>.

Chapter 6 The sixth chapter presents Multiscale Snapshots, a multiscale visual analytics approach for analyzing temporal summaries in dynamic networks. The approach combines temporal hierarchical abstraction with unsupervised graph learning methods to semi-automatically visually analyze changing graph properties at multiple scales. The contributions of the work are: (1) the Multiscale Snapshots approach to visually analyze temporal and structural properties in a dynamic network, and (2) a temporal hierarchical abstraction that utilizes unsupervised graph learning methods to reduce the complexity and speed up analytical tasks. Overall, the approach enables the discovery of similar temporal summaries and the exploration of temporal states, trends, and outliers. The prototype code is accessible online at <https://github.com/eren-ck/MultiscaleSnapshots>.

Chapter 7 The last chapter recaps the presented research and summarizes the thesis contributions. Moreover, the chapter discusses future research perspectives.

1.4 Terminology

In visualization research, practitioners often use terms or concepts for which different definitions exist. Therefore, the following section introduces and defines essential terms used in this thesis.

Graph & Network: A graph models relationships between a set of objects. A graph $G = (N, E)$ consists of nodes (vertices) N and edges (links) $E \subseteq N \times N$ which describe the relationship between nodes. A directed graph consists of ordered edges, and in an undirected case, the nodes are mutually connected. In visualization research, the term network typically refers to real-world graphs (e.g., a social network) where nodes or edges have additional attributes [309]. The node and edge attributes can be of any data type, for instance, typical node and edge attributes are person names or categorical linkage values. Overall, the term network typically refers to real-world systems, and a graph describes the mathematical representation of a network. Thus, this thesis will use the terms graph and network interchangeably.

Dynamic Graph & Network: Dynamic graphs or networks model changing relationships between objects over time. In a dynamic network, there can be changing and varying numbers of nodes, edges, and attributes over time. For instance, users (nodes) typically join or leave social networks, new friendships appear (links), and attributes like people's interests change over time. A dynamic graph can be defined as a sequence of static graphs with $DG = (G_1, G_2, \dots, G_n)$ with G_i representing a static graph and i being the temporal index. Alternative terms for a dynamic network are time-varying [150], evolving [108], or temporal networks [149]. Moreover, for a more detailed discussion of different dynamic network models, please refer to the work of Archambault et al. [16].

Information Visualization: The interdisciplinary field of information visualization develops effective visual communication methods to reveal relationships in complex datasets. The InfoVis pipeline [70] outlines the process of information visualization and describes the mapping of data to visual representations to help users make sense of the data. The pipeline consists of the following high-level steps: (1) transforming raw data into data tables, (2) visually mapping the data tables into visual structures, (3) rendering the visual structures to a view, and (4) utilizing the view to solve a user task. Users have to interact and adapt the individual steps of the InfoVis pipeline to handle complex datasets and solve their particular tasks. Information visualization utilizes algorithms and interactive visualizations to display and reveal relationships in complex datasets.

Multiscale Visualization: Multiscale visualizations are helpful for visually analyzing multiscale processes and datasets in different application domains, for instance, in molecular biology [221] or geography [119]. Such multiscale visualizations scale to larger datasets and reveal patterns at different abstraction scales while producing less clutter. This thesis will use the following definition as derived in our structured literature analysis of multiscale visualizations in Chapter 2: "Multiscale visualizations allow users to present, navigate, and relate data across multiple abstraction scales." Overall, multiscale visualizations help investigate complex systems by displaying and analyzing small-scale patterns and their effects on larger-scale patterns.

Visual Analytics: Visual analytics is the process of interactive visual data analysis, combining automated data analysis methods with interactive visualizations to help users gain knowledge about datasets [171]. The main goal of visual analytics is to combine the strengths of computers and humans to support the effective understanding and reasoning of large and complex datasets [169]. The research field combines methods from data management, information visualization, data mining, machine learning, and human-computer interactions to solve real-world problems. Overall, visual analytics aims to overcome fully-automated algorithms' limitations by including humans in the analysis process using interactive visualizations [169].

1.5 Publications

In the following, I outline the previously published journal and conference publications included in my thesis. Moreover, I detail the work distribution for each publication and each author's contribution providing transparency about the origins of my thesis. The content was written or revised by myself during the writing process. The publications are sorted according to their appearance in the thesis chapters.

[61] **Eren Cakmak**, Dominik Jäckle, Tobias Schreck, Daniel Keim, Johannes Fuchs. "Multiscale Visualization: A Structured Literature Analysis". IEEE Transactions on Visualization and Computer Graphics, 2021.

(Chapter 2)

Contribution clarification. The publication presents a structured literature analysis of multiscale visualizations to understand existing design practices, design considerations, and research challenges. After designing several multiscale visualizations, I had the idea of analyzing common design factors in multiscale visualizations. I took the lead in the project and was responsible for all sections. Dominik Jäckle labeled 20 randomly selected papers using our

coding schema to validate our coding process. Tobias Schreck and Daniel Keim provided feedback on the general idea and commented on paper drafts, as did all other authors. Moreover, Johannes Fuchs regularly provided feedback on the paper's structure, organization, and content. I was the primary contributor to the paper and wrote all sections. I revised all paper sections several times and implemented the online prototype. Therefore, I reuse the paper text without citation marks in Chapter 2.

- [57] **Eren Cakmak**, Hanna Schäfer, Juri Buchmüller, Johannes Fuchs, Tobias Schreck, Alex Jordan, Daniel Keim. “*MotionGlyphs: Visual Abstraction of Spatio-Temporal Networks in Collective Animal Behavior*“. Computer Graphics Forum Vol. 39. No. 3, 2020.

(Chapter 3)

Contribution clarification. The publication is the result of interdisciplinary collaboration. The design study presents a problem characterization and a glyph design for the multiscale abstraction of spatio-temporal networks in the field of collective animal behavior. I initiated the project and took the project lead. Hanna Schäfer helped conduct the pair analytics session and primarily wrote the evaluation section. Juri Buchmüller investigated similar methods and wrote the gaps in the related approaches subsection. Johannes Fuchs regularly provided feedback on the paper's structure, organization, and content. Tobias Schreck, Alex Jordan, and Daniel Keim provided feedback and commented on paper drafts, as did all other authors. I was the primary contributor to the paper and implemented all designs, including the prototype. I wrote all sections and revised Section 3.2.7 as well as Section 3.4, which Juri Buchmüller and Hanna Schäfer initially wrote. Thus, I reuse the paper text without citation marks in Chapter 3.

- [63] **Eren Cakmak**, Manuel Plank, Daniel Calovi, Alex Jordan, Daniel Keim. “*Spatio-temporal clustering benchmark for collective animal behavior*“. Proceedings of the 1st ACM SIGSPATIAL International Workshop on Animal Movement Ecology and Human Mobility, 2021.

(Chapter 3)

Contribution clarification. The publication follows up the design study [57] by presenting a clustering benchmark in the field of collective animal behavior. This interdisciplinary project proposes a reproducible clustering benchmark, including a diverse set of synthetic datasets with ground truth and scalable implementations of spatio-temporal clustering methods. I was responsible for the project lead. Under my supervision, Manuel Plank helped generate the synthetic datasets, contributed to the experiments section, and implemented

the benchmark, including extending state-of-the-art clustering methods. Daniel Calovi supported the synthetic dataset generation by proposing different agent-based models and helped to write the dataset design subsection. Alex Jordan provided background information on the application domain in the background section. Daniel Keim provided feedback and commented on paper drafts, as did all other authors. I was the primary contributor to the paper. I wrote all sections or revised the sections several times during the writing process. Therefore, I reuse the paper text without citation marks in Chapter 3.

- [62] **Eren Cakmak**, Dominik Jäckle, Tobias Schreck, Daniel Keim. “*dg2pix: Pixel-Based Visual Analysis of Dynamic Graphs*“. Visualization in Data Science (VDS), 2020.

(Chapter 4)

Contribution clarification. The publication presents a pixel-based visualization technique to reveal changes and similar temporal states in a dynamic network. The proposed approach allows exploring temporal and structural properties in long sequences of large-scale networks. Again, I initiated and led the project. Dominik Jäckle regularly provided feedback on the paper’s structure, organization, and content. Tobias Schreck and Daniel Keim provided feedback on the general idea and commented on paper drafts, as did all other authors. I was the primary contributor to the paper and wrote all sections. I revised all paper sections several times and implemented the prototype. Thus, I reuse the paper text without citation marks in Chapter 4.

- [59] **Eren Cakmak**, Johannes Fuchs, Dominik Jäckle, Tobias Schreck, Ulrik Brandes, Daniel Keim. “*Motif-Based Visual Analysis of Dynamic Networks*“. Submitted to Visualization in Data Science (VDS), 2022.

(Chapter 5)

Contribution clarification. The publication presents two complementary pixel-based visualizations based on motif analysis to provide an overview of significant topological changes in dynamic networks. The linked pixel-based visualizations allow exploring motifs in different-sized networks to analyze topological structures within and across dynamic networks. Again, I was responsible for the project lead. Johannes Fuchs regularly provided feedback on the paper’s structure, organization, and content. Dominik Jäckle, Tobias Schreck, Ulrik Brandes, and Daniel Keim provided feedback on the general idea and commented on paper drafts, as did all other authors. I was the primary contributor to the paper and wrote all sections. I revised all paper sections several times and implemented the prototype. Therefore, I reuse the paper text without citation marks in Chapter 5.

- [64] **Eren Cakmak**, Udo Schlegel, Dominik Jäckle, Daniel Keim, Tobias Schreck. “*Multiscale Snapshots: Visual Analysis of Temporal Summaries in Dynamic Graphs*“. IEEE Transactions on Visualization and Computer Graphics 27.2 517-527, 2020.

(Chapter 6)

Contribution clarification. The publication presents a visual analytics approach for visually analyzing temporal summaries of a dynamic network at multiple scales. The paper proposes a hierarchical abstraction using unsupervised graph learning methods to reduce the size of a dynamic network and speed up analytical tasks. I initiated and took the lead in the project. Udo Schlegel helped to implement the experimental evaluation. Dominik Jäckle regularly provided feedback on the paper’s structure, organization, and content. Daniel Keim and Tobias Schreck provided feedback on the general idea and commented on paper drafts, as did all other authors. I was the primary contributor to the paper and wrote all sections. I revised all paper sections several times and implemented the prototype. Thus, I reuse the paper text without citation marks in Chapter 6.

I also authored and contributed to the following chronologically ordered publications, which impacted my research but are not included in this thesis.

- [271] Rita Sevastjanova, **Eren Cakmak**, Shauli Ravfogel, Ryan Cotterell, Mennatallah El-Assady. “*Visual Comparison of Language Model Adaptation*“. IEEE Transactions on Visualization and Computer Graphics, 2022.
- [49] Juri Buchmüller, Udo Schlegel, **Eren Cakmak**, Daniel Keim, Evanthia Dimara. “*SpatialRugs: A compact visualization of space and time for analyzing collective movement data*“. Computers & Graphics, 2021.
- [87] Frederik Dennig, **Eren Cakmak**, Henrik Plate, Daniel Keim. “*VulnEx: Exploring Open-Source Software Vulnerabilities in Large Development Organizations to Understand Risk Exposure*“. IEEE Symposium on Visualization for Cyber Security (VizSec), 2021.
- [214] Tran Hai Dang Mai, Felix Hinderer, **Eren Cakmak**, Udo Schlegel, Juri Buchmüller, Daniel Keim. “*VisMe: Visual and Uncertainty Analysis tool in Object Recognition*“. IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2020 Mini Challenge 2), 2020.
- [106] David Englert, Pablo Martinez-Blasco, **Eren Cakmak**, Udo Schlegel, Juri Buchmüller, Daniel Keim. “*Somewraps - Something with Graphs: Visual*

- Analytics of Network Data*". IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2020 Mini Challenge 1), 2020.
- [265] Udo Schlegel, **Eren Cakmak**, Daniel Keim. "*ModelSpeX: Model Specification Using Explainable Artificial Intelligence Methods*". International Workshop on Machine Learning in Visualisation for Big Data (MLVis), 2020.
- [50] Juri Buchmüller, Udo Schlegel, **Eren Cakmak**, Evanthia Dimara, Daniel Keim. "*SpatialRugs: Enhancing Spatial Awareness of Movement in Dense Pixel Visualizations*". EuroVis Workshop on Visual Analytics (EuroVA), 2020.
- [263] Udo Schlegel, **Eren Cakmak**, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, Daniel Keim. "*Towards visual debugging for multi-target time series classification*". Proceedings of the 25th International Conference on Intelligent User Interfaces (IUI), 2020.
- [47] Juri Buchmüller, **Eren Cakmak**, Natalia Andrienko, Gennady Andrienko, Jolle Jolles, Daniel Keim. "*Moving Together: Towards a Formalization of Collective Movement*". EuroVis Workshop on Visual Analytics (EuroVA), 2019.
- [266] Udo Schlegel, Wolfgang Jentner, Juri Buchmüller, **Eren Cakmak**, Giuliano Castiglia, Renzo Canepa, Simone Petralli, Luca Oneto, Daniel Keim, Davide Anguita. "*Visual Analytics for Supporting Conflict Resolution in Large Railway Networks*". INNS Big Data and Deep Learning conference (Springer Cham), 2019.
- [48] Juri Buchmüller, Dominik Jäckle, **Eren Cakmak**, Ulrik Brandes, Daniel Keim. "*MotionRugs: Visualizing Collective Trends in Space and Time*". IEEE Transactions on Visualization and Computer Graphics, 2019.
- [66] **Eren Cakmak**, Daniel Seebacher, Juri Buchmüller, Daniel Keim. "*Time Series Projection to Highlight Trends and Outliers*". IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC2), 2018.
- [65] **Eren Cakmak**, Udo Schlegel, Matthias Miller, Juri Buchmüller, Wolfgang Jentner, Daniel Keim. "*Interactive Classification Using Spectrograms and Audio Glyphs*". IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC1), 2018.
- [29] Benedikt Bäumle, Ina Bösecke, Raphael Buchmüller, Yannick Metz, Juri Buchmüller, **Eren Cakmak**, Wolfgang Jentner, Daniel Keim. "*Interactive Webtool for Tempospatial Data and Visual Audio Analysis*". IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC1), 2018.

- [246] Isabel Piljek, Giuliana Dehn, Jannik Frauendorf, Ziad Salem, Yertzahn Niyazbayev, Juri Buchmüller, **Eren Cakmak**, Wolfgang Jentner, Florian Stoffel, Daniel Keim. “*Identifying Patterns and Anomalies within Spatiotemporal Water Sampling Data*“. IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC2), 2018.
- [58] **Eren Cakmak**, Giuliano Castiglia, Wolfgang Jentner, Juri Buchmüller, Daniel Keim. “*Visualization For Train Management: Improving Overviews in Safety-critical Control Room Environments*“. 4th International Symposium on Big Data Visual and Immersive Analytics, 2018.
- [264] Udo Schlegel, **Eren Cakmak**, Juri Buchmüller, Daniel Keim. “*G-Rap: interactive text synthesis using recurrent neural network suggestions*“. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2018.
- [60] **Eren Cakmak**, Alexander Gärtner, Thomas Hepp, Juri Buchmüller, Fabian Fischer, Daniel Keim. “*Applying visual analytics to explore and analyze movement data*“. IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2015 MC1), 2015.

Multiscale Visualization: A Structured Literature Analysis

Summary

The visualization community typically employs multiscale visualizations to explore multiscale processes and data in different application domains. However, developing such multiscale visualizations remains challenging due to the plethora of existing work and the expression ambiguity in the visualization research community. The following chapter presents a structured literature analysis to compare and categorize common design practices in multiscale visualization research. In the literature analysis, we reviewed and organized 122 published journal and conference papers between 1995 and 2020. Visualization researchers can use the resulting taxonomy to explore existing techniques, common design practices, research trends, and open research challenges. The primary goal of this chapter is to provide an overview of existing multiscale visualization approaches and help visualization practitioners to develop new multiscale navigation and visualization techniques. The taxonomy, the list of reviewed papers, and the resulting paper codings are accessible online at <https://multiscale-vis.dbvis.de>.

The chapter is based on the following publication. Please refer to Section 1.5 for contribution clarifications. The used icons in this chapter are MaterialDesign icons (Apache License 2.0).

- [61] **Eren Cakmak**, Dominik Jäckle, Tobias Schreck, Daniel Keim, Johannes Fuchs. “*Multiscale Visualization: A Structured Literature Analysis*“. IEEE Transactions on Visualization and Computer Graphics, 2021.

2.1 Introduction

Many multiscale visualizations have been proposed in visualization research. These multiscale visualizations are essential in various application domains to analyze large and high-dimensional datasets, such as in geography [119], physics [241], or biology [221]. For instance, in molecular biology, multiscale visualizations are used to analyze genomes' multiscale hierarchical structure, such as the nucleus with a division into chromosomes, fibers, and, at the lowest scale, atoms [133]. Typically, in contrast to single-scale visualizations, multiscale visualizations scale to larger datasets, produce less clutter, and reveal the emergence of patterns at different levels of scale. For example, aggregation methods can be recursively utilized to promote a top-down or bottom-up hierarchical visual exploration of large datasets [98]. However, designing multiscale visualizations is challenging due to the plethora of existing approaches and different design considerations.

In visualization research, authors regularly used the expression multiscale (multi-scale) visualization in different contexts with often varying meanings. Examples of different contexts include interaction-based multiscale zooming methods [159] or multiscale statistical summary visualizations [295]. Visualization experts know about the expression's ambiguity and typically specify the meaning in their papers. However, the different definitions of what is meant by a multiscale visualization may be confusing for novice readers. For example, selecting a multiscale visualization approach can be challenging for data analysts due to the expression's ambiguity. There has been little work to categorize and compare multiscale visualizations to understand their design practices. To address this challenge, we provide a systematic literature analysis of multiscale visualizations to gain insights into common design factors and improve communication between researchers.

In this chapter, we provide a comprehensive overview of multiscale visualization approaches. We systematically analyzed 122 papers from multiple journals and conferences to understand general design practices for multiscale visualizations. The result is a categorization of multiscale visualization approaches into a taxonomy. We discuss how different multiscale visualizations enable us to analyze and relate information at various scales to gain insight into complex systems, such as in molecular biology [221]. Further, we summarize design considerations and highlight open research challenges for multiscale visualizations. Overall, we provide a basis for the systematic reasoning about multiscale visualizations, and the key contributions are: (1) a unified definition of the terminology, (2) a taxonomy of design practices for multiscale visualizations, (3) a summary of design considerations, and (4) a

collection of crucial open research challenges. An extensive list of the reviewed papers, the resulting paper codings, and the taxonomy are accessible online at <https://multiscale-vis.dbvis.de>.

2.2 Background

In this section, we first examine some definitions and derive a unified consensus on the multiscale visualization terminology. We also search for similar concepts and synonyms in visualization research. The second part discusses commonalities and differences between our literature analysis to related work.

2.2.1 Terminology

Some expressions are often so widely used that people use them without specifying their exact meaning. The term multiscale visualization belongs to these expressions. In visualization research, the potential characteristics and interpretations of multiscale visualizations are quite broad. Therefore, we reviewed existing definitions to derive a consensus on what is meant by multiscale visualization.

In a broader context, multiscale visualizations are a form of multiscale analysis. In many fields, multiscale analysis is widely used to understand the emergent properties of systems in the real world, such as in physics [241] or biology [8]. The essential term "multiscale" has the following dictionary definition: "operating or occurring over different levels" [89]. The dictionary definition highlights the main characteristics of multiscale analysis, analyzing data at various levels of detail. Such a multiscale analysis's primary goal is to investigate complex systems by examining small-scale patterns and their effects on emerging large-scale patterns [8]. For instance, multiscale analysis is useful to analyze local interactions between animals in collective animal behavior to understand individual animals' influence on large-scale swarm behavior [209].

In the following, we examine definitions of multiscale visualizations to derive a more precise definition. First, Furnas and Bederson [118] specify multiscale visualization (multiscale interfaces) as an approach to display data at different magnifications or scales. Next, Stolte et al. [287] provide another perspective. The authors emphasize that multiscale visualizations utilize data and visual abstraction methods to present the data at different abstraction levels. Data abstractions transform and reduce the underlying dataset (e.g., aggregation or filtering), and visual abstractions change

the data point representations (e.g., semantic zooming or distortions). Further, Elmqvist and Fekete [98] propose a multiscale structure and navigation strategies to turn existing approaches into multiscale visualizations and present data at multiple aggregation levels. Ebert et al. [95] describe the need for multiscale interactions to understand scientific data and system-of-systems at multiple problem scales. Viola and Isenberg [306] characterize multiscale visualizations as representations that display and relate abstracted data across various levels of scale.

We want to highlight that the previous definitions include different concepts such as navigating and relating abstracted data (e.g., aggregated data) across scales. These concepts are essential in multiscale analysis in various domains. For example, in the visualization of DNA nanostructures [221], domain experts have to navigate and relate information across different scales to understand complex system-of-systems. Overall, concepts such as the presentation and navigation of different abstraction scales expose patterns and relationships in datasets at varying scales. Therefore, we derive the following definition from the listed previous research: “*Multiscale visualizations allow users to present, navigate and relate data across multiple abstraction scales.*“ Our definition integrates various interpretations to specify the ideal characteristics of multiscale visualizations.

We reviewed the visualization literature to identify similar concepts and notions to the expression *multiscale visualization*. We use these similar expressions in our literature analysis as search terms to identify related papers. We used the IEEE VIS paper keyword search by Isenberg et al. [155] and the derived keyword topics [153] to search for synonyms. Additionally, we scanned the keywords and abstracts of the updated metadata collection about IEEE VIS publications [154]. We reviewed the CHI conference proceedings accessible on the ACM digital library for related expressions. We searched for the author keywords (tags) for “multiscale“ and scanned the resulting 29 papers for related expressions. We identified multiple reoccurring similar expressions such as *multi-scale*, *multiple scales*, *multi-level*, *cross-scale*, *multi-resolution*, and *multiple resolutions* in combination with terms such as *visualization*, *interface*, *representation*, *viewing*, *interaction*, *navigation*, *model*, *design*, and *analysis* are used to describe similar concepts in visualization research. To determine which of the related expressions is most often used in the literature, we investigated the term usage of our literature analysis search results (see Figure 2.1). The term multiscale (multi-scale) is the most commonly used term of the previously listed expressions.

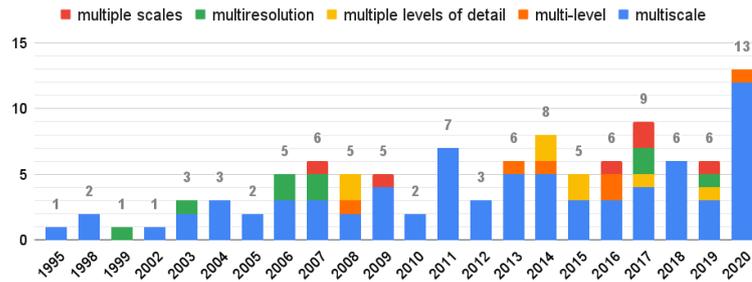


Fig. 2.1 The five most frequently used expressions among the surveyed 122 papers. The most common expression is multiscale visualization.

2.2.2 Related Work

Multiscale visualization has been a part of visualization research for some time. Next, we discuss related theories and surveys that describe multiscale visualizations.

Theoretical Work: Many related theory papers discuss multiscale visualization approaches. Furnas and Bederson [118] provide an analytical framework and space-scale diagrams to understand multiscale interfaces. Stolte et al. [287] formalize multiscale visualizations using abstraction methods for data cubes. Kehrer and Hauser [168] discuss multi-faceted visualization approaches, including a multi-model scenario. Goodwin et al. [119] discuss the modifiable areal unit problem (MAUP) [234] and propose a framework for multivariate visual comparison across multiple geographical scales. Viola and Isenberg [306] examine and formalize the concept of abstraction in visualization research. The authors discuss multiscale visual abstractions for spatial and temporal data. In comparison, we focus less on providing another theoretical framework and concentrate more on presenting an overview of multiscale visualization design practices in visualization research.

Surveys: In visualization research, three surveys investigate multiscale visualizations in specific application domains. Vaquero et al. [298] review the visualization and interaction techniques for multiscale biomedical data, such as anatomy or genomics. Ezzati-Jivan and Dagenais [110] survey multiscale navigation of execution trace data, focussing on multilevel trace abstraction and visualization methods. Miao et al. [220] discuss multiscale visualization techniques for analyzing and manipulating 3D DNA structures in molecular biology. These surveys investigate multiscale visualizations for particular application domains. Furthermore, Ebert et al. [95] describe challenges and opportunities for multiscale scientific visualizations. In contrast, we systematically review design practices for multiscale visualizations in a broader context of visualization research, exceeding the traditional survey scope.

Hierarchical Visualizations: Further related work focus on hierarchical and tree-based visualizations. Yang et al. [325] propose a framework called Interactive Hierarchical Displays (IHD) for the multi-resolution view and navigation (e.g., drill-down) of hierarchies. Elmqvist and Fekete [98] propose a more general framework that presents a multiscale structure and navigation methods to turn existing techniques into multiscale visualizations. Schulz et al. [269] elaborate on the design space of implicit tree visualizations. In contrast, we provide a broader review of visualization research by analyzing design factors in existing multiscale visualizations.

In summary, all these previous approaches present significant contributions by introducing frameworks, techniques, or domain-specific surveys. However, none of the previous work explored the broader visualization literature for existing multiscale visualization. Such a literature analysis is essential to understand common practices (e.g., interaction methods and targets). To the best of our knowledge, our literature analysis is the first analysis of design practices for multiscale visualizations.

2.3 Methodology

Our literature analysis's primary goal is to give a comprehensive overview of multiscale visualizations. The guidelines for qualitative literature analysis [124] inspired our methodological approach. We focus on papers that use the expression *multiscale visualizations* or the identified related expressions (see Section 2.2.1). Moreover, our literature analysis cannot include all possible multiscale data models in visualization research, as this would go far beyond the scope of our work. We did not explore visualization approaches that only employ hierarchical or tree-based models. Specifically, we omitted all papers that only utilize multiscale models (e.g., hierarchical clustering) without any multiscale visualization. In the following, we describe our literature search and analysis procedure.

2.3.1 Selection of Literature

First, we used multiple search engines to identify relevant papers from various conferences and journals. We used the search term *visualization* and the identified related expressions (see Section 2.2.1) for online keyword search. We used the following search engines, which led to the results: IEEE Xplore digital library (327 results), ACM digital library (651 results), EG digital library (129 results), and DBLP computer science bibliography (781 results).

The automatically identified papers were refined in three steps. In the first step, we only included peer-reviewed full papers published in journals or conferences. The step reduced the number of papers from 1888 to 1312. In the second step, we manually excluded papers unrelated to multiscale visualizations. We excluded papers that only use multiscale models (e.g., hierarchical clustering) without any multiscale visualization. As a result, the papers were further filtered from 1312 to 75. As for the last step, we recursively scanned the paper references and followed the citations in both directions on Google Scholar. Hence, the number of papers increased again from 75 to 122.

2.3.2 Coding Scheme

We developed a coding scheme and tagged the 122 papers with labels. The coding scheme is designed to capture multiscale visualization characteristics and is based on existing taxonomies. We combined some labels in more abstract categories to keep our coding scheme focused and manageable. Thus, some details might get lost, like the distinction between line charts and scatterplots, which have been summarized as statistical graphics. A paper can have multiple labels of a specific coding category (e.g., multiple target labels). The authors coded the papers. We randomly selected and encoded 20 papers redundantly to validate our coding process. For the redundantly encoded papers, Cohen's kappa coefficient for inter-rater reliability reached a substantial agreement with $\kappa = 0.61$ (83% overall agreement). We tagged the 122 papers with the following coding scheme (see Table 2.1).

Journal: We labeled the papers with the year and journal or conference to identify trends and the leading paper outlets.

Visualization Idioms: Munzner [229, Chapter 7-9] describes various categories of visualization techniques for different dataset types, such as spatial or network data. We selected ten prominent visualization idioms from the described visualization techniques to label the respective multiscale visualizations. We also added an extra category “other“ describing unique visualizations that do not fall into any defined visualization idioms category. The following list summarizes the labels.

- *statistical graphics*: traditional charts, such as line charts, bar charts, or scatterplots.
- *parallel coordinates*: display multivariate datasets as lines between parallel axes.

Category	Labels	Multi-Label	Source
Journal	name of the journal or conference and year		Source of Publication
Visualization Idiom	statistical graphics, parallel coordinates, dense layouts, glyph, 3D, geographic, spatial fields, graph, stacked charts, other	✓	Munzner [229, Chapter 7-9]
Target	trends, outliers, features, distributions, extremes, dependency, correlation, similarity, topology, paths, shape	✓	Munzner [229, Chapter 3]
Interaction	select, navigate, arrange, change, filter, aggregate	✓	Brehmer and Munzner [46]
Composite Visualization	juxtaposed, superimposed, overloaded, nested, and integrated views	✓	Javed and Elmquist [158]
Dataset Type	tables, networks & trees, fields, geometry	✓	Munzner [229, Chapter 2]
Attribute Type	categorical, ordinal, quantitative, hierarchical	✓	Munzner [229, Chapter 2]
Navigation Strategy	top-down and bottom-up exploration strategies		Battle and Heer [28]
Level of Analysis	microscale, mesoscale, macroscale	✓	Shi et al. [274] and Xu et al. [324]
Application Area	CompSystems (computing systems), LifeBio (life sciences, biology), MLStatsModel (machine learning, statistics), ScienceEngr (physical science, engineering), SocHum (social science, humanities), OtherApp (other application areas), NAAApp (domain agnostic)		IEEE VIS Paper Submission Keywords [307]
Paper Type	technique (technique & algorithm), system, design study (application & design study), evaluation (empirical study), or model (theory & model)		IEEE VIS Paper Types [152]
Evaluation	computational benchmark, qualitative evaluation, quantitative evaluation, usage scenario, and no evaluation	✓	visualization evaluation strategies [9, 71, 191]

Tab. 2.1 The applied coding scheme with tags. In case a category is multi-label, then several labels can be assigned to one paper.

- *dense layouts*: pixel-oriented visualization techniques display data records' values as colored pixels.
- *glyph*: multivariate data records are mapped to glyph, icon, and symbol representations.
- *3D*: three-dimensional geometric visualizations.
- *geographic*: geographic visualizations for spatial data, such as choropleth maps.
- *spatial fields*: visualizations of scalar-, vector-, and tensor fields.
- *graph*: graph (network) and tree visualizations.
- *stacked charts*: present data in multiple stacked layers, such as streamgraph visualizations.
- *other*: visualization techniques not fitting into any of the categories above.

Target: We labeled the target of the visualization using the suggested detailed targets by Munzner [229, Chapter 3].

Interaction: We used the manipulation methods for visualizations of Brehmer and Munzner [46] to tag the supported interaction methods. The listed manipulation methods [46] unify interaction and visual encodings as both are closely related to each other.

Composite Visualization: We used the composite visualization design space [158] to label the combination of different visual representations in the same view.

Dataset Type: We facilitated four basic dataset types (*tables, networks & trees, fields, geometry*) described by Munzner [229, Chapter 2] to tag each paper.

Attribute Type: Munzner [229, Chapter 2] described the four attribute types *categorical, ordinal, quantitative, and hierarchical*. We labeled the papers using these attribute types.

Navigation Strategy: We consider two navigation strategies top-down and bottom-up exploration strategies [28]. The strategies can be described by drill-down (top-down) and roll-up (bottom-up) operations.

Level of Analysis: We consider the three levels of analysis scale: micro-, meso-, and macroscale. These analysis levels are often used to describe the analysis scale (e.g., in Shi et al. [274] and Xu et al. [324]). Microscale analysis is the smallest level of scale that displays individual data points, such as examining nodes and edges in a graph. The mesoscale analysis is in-between and investigates structural properties, for instance, analyzing motifs and communities in a graph. Macroscale analysis focuses on the dataset's global properties, such as the number of nodes and edges in a graph. Ideally, a multiscale visualization visualizes all three analysis scales to enable users to relate abstracted data across scales.

Application Area: We utilized the IEEE VIS application areas keywords (see Table 2.1) to tag the application domain [307].

Paper Type: We categorized the papers according to the five IEEE VIS paper types (see Table 2.1) to point out popular paper types in the research field [152].

Evaluation: We investigated common evaluation strategies in visualization research [9, 191] and on quantitative evaluation studies [71]. We used five tags to label the evaluation strategies: *computational benchmark, qualitative evaluation, quantitative evaluation, usage scenario, and no evaluation*.

2.4 Multiscale Visualization Taxonomy

The following section outlines prevalent coding labels and a taxonomy of similar multiscale visualization contributions. Furthermore, we derived design considerations based on our structured literature analysis. Researchers can explore the complete paper codings and the taxonomy online at <https://multiscale-vis.dbvis.de>.

2.4.1 Coding Results

First, we provide a high-level overview of the coding labels based on the coding scheme categories (see Table 2.1). The following percentages always refer to the 122 papers and do not necessarily add up to 100%, as a paper can have multiple category labels at the same time.

Publication Venues & Paper Types In recent years, an increasing number of multiscale visualization papers have been published (see Figure 2.2). The top three publication venues are IEEE TVCG (44/122), Computer Graphics Forum (15/122), and ACM CHI (14/122). The remaining 49 papers were published in related journals or conferences. The most common paper types are technique (46%), design study (25%), and model (16%) papers. The two other paper types, system (7%) and model (6%), rarely appear over the years. From 2015 to 2020, the proportion of paper types has remained constant, except for a fluctuating number of design study papers. Recently, the IEEE TVCG publications reached an all-time high with nine papers in 2020 as the topic is gaining popularity for visualizing large-scale datasets.

Visualization Idioms The following labels, statistical graphics, geographic, 3D, and graph, occur separately in 25-28% of all papers. The previous four labels, considered altogether, appear in about 80% of all papers. The number of 3D, geographic, and graph idioms has steadily increased since 2008 due to a growing number of multi-scale visualizations in social sciences and biology (e.g., 3D DNA visualization [220]). Each of the remaining idioms occurs as follows: 10% dense layouts, 10% glyph, 7% parallel coordinates, and 5% spatial fields, as well as stacked charts. Interestingly, our label "other", representing unique visualization techniques and tailored design studies, appears in 40% of all papers.

Target The commonly assigned target labels for visualizations are with 80% features, 57% shape, 41% similarity, 39% distribution, and 30% for topology as well as trends. The remaining target labels appear in 28% paths, 27% correlation, 27% outliers, and 18% dependency of all papers. The target label "extremes" occurred only ten times, which is rare considering the number of analyzed tabular datasets (46%).

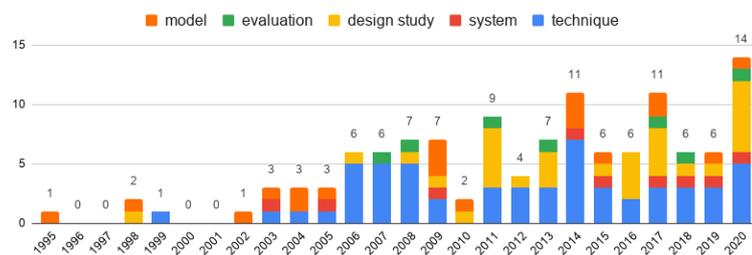


Fig. 2.2 The chart presents the paper types for the years 1995-2020.

Interaction In almost all papers, essential interaction methods are the navigation (85%) and selection (83%) of abstraction scales. Papers without those two labels discuss more theoretical contributions, such as frameworks or workflows. The proportion of the label aggregation (47%) and change (34%) has remained constant since 2006. The interaction methods filter (30%) and arrange (17%) have slightly increased after 2014. Notably, the interaction methods select and navigate likewise aggregate and change tend to appear together as navigation across scales often includes selecting an appropriate scale, and aggregation involves changing the data abstraction scale.

Composite Visualization An overall 45% of all papers received the label juxtaposed. Each remaining label nested, superimposed, integrated views, and overloaded appeared overall in 13-15% of all papers. In terms of temporal shifts, we observed 13 superimposed views from 2013 to 2017, contrasting the only four previously superimposed views from 2003 to 2013. Furthermore, 40 papers did not describe any composite views, as the approaches proposed only visualization techniques or discussed theoretical work.

Dataset Type The utilized types are 50% geometry, 46% table, 25% network & tree, and 7% field datasets. We want to highlight that tabular dataset appear nearly 13% in conjunction with geographic or network & tree datasets, i.e., geographic attributes. Multiscale analyses of field datasets first appeared in 2014 and are overall underrepresented with only eight papers.

Attribute Type The analyzed attribute types are in 81% of the cases categorical, 46% quantitative, and only 2% ordinal. Furthermore, 18% of papers analyze hierarchical data attributes. We want to highlight that there are no dedicated multiscale visualizations for only ordinal data attributes.

Navigation Strategy Overall, 74% of papers utilize top-down approaches, with only eight papers applying bottom-up approaches. Seven of the eight bottom-up approaches were proposed after 2013. There are only three approaches that describe only the bottom-up navigation strategies.

Level of Analysis For the next category, the label occurrences are as follows: 89% microscale, 75% mesoscale, and 12% macroscale. For 65% of all papers, the labels microscale and mesoscale occur together. We noticed that most multiscale visualizations are not displaying macroscopic information, which is essential for relating the abstracted structures to the overall global dataset properties.

Application Area The labels appear with the following frequencies: 28% LifeBio, 11% SocHum, 8% CompSystems, 5% ScienceEngr, and 2% MLStatsModel. Further, 34% of all papers are domain agnostic, and 15% are in other application areas. The following trends have emerged. The number of life science and biology papers has steadily increased from one in 2013 to six papers in 2020. Multiscale visualizations

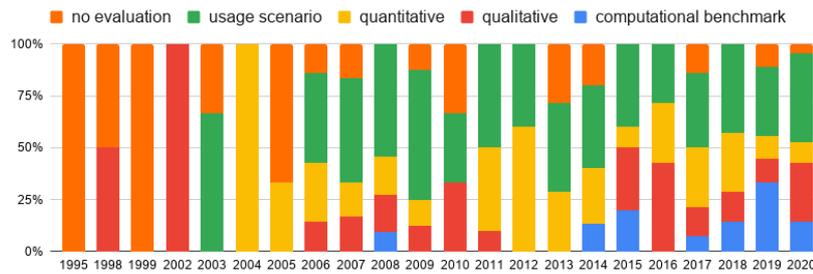


Fig. 2.3 The proportion of paper evaluations for the years 1995-2020, showing a positive trend towards more extensive paper evaluations.

for machine learning applications appeared after 2017 and will inevitably increase in the future, as multiscale visualizations are suited to display deep learning architectures at varying scales, such as network layers and their underlying neurons.

Paper Evaluation The number of utilized evaluation approaches are 51% usage scenario, 29% quantitative as well as 20% qualitative user studies, 20% no evaluation, and 11% computational benchmark. We also examined the proportion of evaluation methods over the years (see Figure 2.3). The analysis indicates an increase in quantitative and qualitative user studies, including a slight decrease in usage scenarios. Additionally, since 2014, there has been an increase in computational benchmarks in paper evaluations. Overall, there is a positive trend towards more detailed evaluations with benchmarks and user studies.

2.4.2 Multiscale Visualization Taxonomy

Next, we introduce prevalent classes of contributions in multiscale visualization research. Our taxonomy consists of six main classes of paper contributions with multiple sub-classes (see Figure 2.4). We outline how we derived the taxonomy based on several clustering iterations and the refinement of the clusters. First, we encoded the labels using one-hot encoding and applied k-means clustering using the cosine similarity to identify similar multiscale visualization papers. Considering the input parameters, we used the silhouette coefficient and the elbow method to identify a decent number of k -clusters. We decided to select $k = 6$ after we examined k between two and twenty. We manually analyzed the clusters in the second step and chose appropriate class names for each cluster. We also refined and reassigned 22 borderline papers to more suitable classes. Finally, we recursively applied the previously described steps to the resulting six classes to identify similar sub-classes of papers. We assigned each of the 122 reviewed papers to exactly one sub-class. Next, we describe the common design factors of each sub-class.



Fig. 2.4 The chart shows the resulting taxonomy, including the number of papers. The six classes are *Multiscale Visual Representations*, *Multiscale Visualization Applications*, *Multiscale Visual Analytics*, *Multiscale Interaction & Navigation*, *Theoretical Work*, and *Multiscale Visualization Systems*.

Multiscale Visual Representations

Multiscale visual representations are listed as a primary contribution across the reviewed papers. The class contains multiscale visualization technique papers, including two design studies that list visualization techniques as part of their contribution. The class is further divided based on visualization idioms (see Figure 2.5) into the six sub-classes: statistical graphic, 3D, geographic, graph & tree, dense, and miscellaneous representations. The remaining visualization idioms did not occur often enough to form sub-classes.

 **Statistical Graphics (5/36)**: All sub-class papers utilize juxtaposed statistical graphics to analyze temporal patterns at multiple scales. The primary targets are exploring temporal data (e.g., time series) to discover similar features (5), including identifying trends (4). The sub-class papers provide the following interaction methods to change (5), navigate (4), and aggregate (4) data. The datasets are tabular (5), examining mainly quantitative data attributes (4). For example, a unique paper is the work of Mao et al. [216], which depicts multiscale statistical trends in text documents, including low-level semantics (e.g., topic shifts) and high-level characteristics (e.g., general trends), as a smooth curve.

 **3D (8/36)**: The second sub-class employs 3D multiscale visual representations tailored for biological applications (8) to explore 3D hierarchical datasets (3). The sub-class consists of visualizations for geometric (7) and field datasets (4). The central targets are to identify distributions of geometric shapes (8) and similar features (7). The proposed interaction methods are to select (7) and navigate (6) 3D spaces in a top-down manner. The analysis level is mainly mesoscale (7), including interactive aggregations methods to locate and compare geometric shapes (3). An example paper is ClearView [186], an interactive focus+context visualization method for complex volumetric data.

 **Geographic (5/36)**: The next sub-class summarizes geographic visual representations that provide insight into spatial phenomena. The targets are to discover in

	statistical graphics	parallel coordinates	dense layouts	glyph	3D	geographic	spatial fields	graph	stacked charts	other
	5	0	0	0	0	1	0	0	1	4
	0	0	0	2	8	1	4	0	0	1
	2	0	0	1	0	5	0	2	1	1
	2	0	0	1	0	0	0	8	1	5
	1	0	5	1	1	0	0	1	0	1
	0	2	0	0	0	0	0	0	0	3
	10	2	5	5	9	7	4	11	3	15

Fig. 2.5 The Figure presents how often visualization idioms appear in the six multiscale visual representation sub-classes. Papers usually utilize multiple visualization idioms in their approaches (e.g., ZAME [97] depicts a matrix visualization with glyphs).

all papers spatial distributions, trends, outliers, and features, such as shapes. The interaction methods are to select, navigate, filter, aggregate, and change spatial scales (4). The navigation strategy is top-down from coarse to fine granular (5) and depicts micro-, and mesoscale (3). For example, the TopoGroups [331] technique provides an overview and navigation means to explore geographical distributions across different aggregation scales.

 **Graph & Tree (8/36):** The following sub-class is about abstracting and visually exploring graph data such as networks and trees. The sub-class papers summarize graph structures into a hierarchy of strongly connected sub-graphs, for example, recursively into a multiscale visualization of small world networks [22]. The regular targets are to explore similar aggregated graph topologies (8), paths (7), and features (5). Nearly all approaches enable users to select, navigate and aggregate the graphs in a top-down fashion to examine nodes (microscale) and meta-nodes (mesoscale). Interestingly, paper evaluations only report usage scenarios (7), except for some computational benchmarks (2). Lately, for instance, Pezzotti et al. [243] proposed a technique to explore large bipartite graphs (social networks) to reveal a hierarchy of clusters.

 **Dense Layout (5/36):** The next sub-class papers present temporal events with dense layouts, also known as pixel-based visualizations. The primary targets are to compare similar features (5), including identifying temporal trends (3) in large datasets. All sub-class papers combine navigation and aggregation interaction methods for tabular datasets at micro-, and mesoscale. All evaluations are primarily usage scenarios. For instance, dg2pix [62] provides an overview of large dynamic graphs using a dense pixel-based visualization to explore graph embeddings at multiple temporal scales. A notable paper is Pálenik et al. [236] that proposes a pixelmap to analyze spatio-temporal particle simulations at multiple temporal and spatial scales.

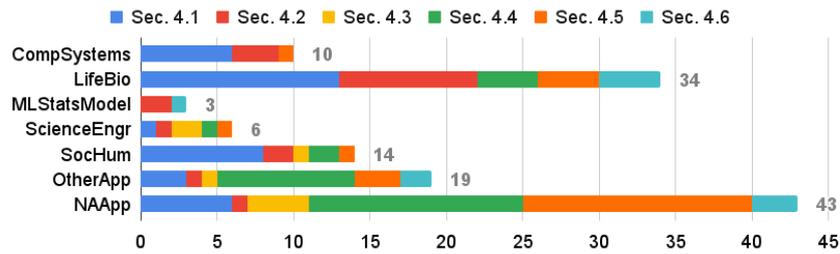


Fig. 2.6 The chart presents the application areas distribution for the corresponding taxonomy classes Section 4.2.

Miscellaneous (5/36): The last sub-class contains rarely occurring visualizations idioms. Like the two parallel coordinate approaches that combine aggregations with navigation methods to summarize features, trends, and outliers [115, 252]. The remaining three papers propose distinct techniques. For example, Veras and Collins [302] propose a display-optimized tree cut algorithm to reduce clutter for multiscale visualizations, such as treemap or sunburst diagrams. Since the sub-class contains different approaches, describing common design factors is pointless.

Summary: The central element of the class papers is to visually explore and compare similar features (32), distributions (23), shapes (16), network topologies as well as paths (16) of data across multiple scales. However, relating data across scales is challenging and often overwhelming for users due to the cognitive and interaction overload [331].

Multiscale Visualization Applications

The second class encompasses design study papers that describe and solve application-focused challenges using multiscale visualizations. Figure 2.6 provides a general overview of the surveyed 122 papers' application areas.

Biological Applications (8/19): The sub-class papers appeared in biology and life sciences. The papers commonly utilize juxtaposed visualizations (4), such as 3D and graph representations. Typical targets are to explore and summarize similar network (5) and geometric (4) datasets features (8) and distributions (4), such as 3D shapes (5), network topologies (5), and paths (5). The interaction methods are selecting (8) and navigating (6) in a top-down fashion to filter and change categorical data (8) attributes across micro- and mesoscale. The paper evaluations are usage scenarios (6), including some qualitative user studies (4). For instance, Abstractocyte [227] enables exploring 3D mesh and node-link representation of astrocytes and neurons.

 **Computing Applications (6/19):** The second sub-class contains design study papers in computing, including machine learning applications (2). The used visualization idioms are graph (4) and other juxtaposed (5) domain-specific visual representations for categorical and quantitative data attributes. The targets are to investigate features (6), network topology (5), and outliers (4) in tabular (5) and network datasets (3). The utilized interaction methods are selecting (6), navigating (5), filtering (5), and changing (5) the data granularity using aggregation methods. All papers present usage scenarios as a central part of their evaluation. A recent sub-class paper is, for instance, Cao et al. [69] river-based visualization to explore adversarial examples in deep neural networks at multiple levels.

 **Spatio-Temporal Applications (3/19):** The third application sub-class is about multiscale spatio-temporal analysis. The sub-class consists of papers focusing on visually analyzing spatio-temporal data across multiple spatial scales. For example, Biswas et al. [40] propose a workflow to examine the uncertainty of multiple weather ensemble models across varying spatial resolutions. Given that the sub-class consists of only three papers, the description of common design factors is excessive.

 **Miscellaneous Applications (2/19):** The last sub-class contains papers that did not fit into the previously listed sub-classes. One paper describes the multi-level visualization design for poetry [226], and the other paper the interactive analysis of social tag networks and hierarchies [121]. The discussion of common design factors for this sub-class is again challenging, considering the number of papers.

Summary: Visualization researchers proposed biological (8), computing (6), and spatio-temporal (3) design study papers. However, the proposed application-specific solutions are often challenging to transfer and generalize to similar issues in other application areas.

Multiscale Visual Analytics

The third class contains multiscale Visual Analytics (VA) approaches for temporal, geospatial, and graph datasets. Since the class contains only eight papers, we will briefly discuss some design factors for the whole class. The targets are exploring quantitative (7) and categorical (5) data attributes to identify overall distinct features (7), outliers (7), and trends (6). The papers implement a rich set of interaction methods, including selecting (8), navigating (8), filtering (7), and changing (6), as well as aggregating (6) data.

 **Temporal VA Approaches (3/8)**: The first sub-class encompasses three papers to explore time-series data, utilizing the Visual Analytics mantra [172]. The papers allow exploring features, extremes, trends, and outliers in time-series data. For instance, Sips et al. [280] proposed a rare bottom-up navigation approach that utilizes a matrix-like visualization for the multiscale exploration time-series patterns.

 **Geospatial VA Approaches (3/8)**: The second sub-class consists of approaches for geospatial datasets that provide extensive systems for the multiscale analysis of geospatial features, such as shapes. For example, Wang et al. [310] presented a multi-resolution VA approach for weather simulation ensembles, comprising nested parallel coordinate plots. The paper combines juxtaposed, superimposed, and nested composite visualizations with set operations and range queries to highlight parameter correlations for the weather simulations.

 **Graph-Based VA Approaches (2/8)**: The last sub-class contains VA approaches for graph datasets. The approaches enable users to analyze relationships and clusters across scales to identify similar network topologies. For example, Multiscale Snapshots [64] utilizes graph embeddings with multiple visual metaphors to semi-automatically analyze temporal states and trends in dynamic graphs. The two approaches display various temporal scales using different visual representations at all analysis levels.

Summary: Visual Analytics aims to overcome the information overload of large-scale datasets by interactive semi-automated means which involve the user in the visual exploration process [172].

Multiscale Interaction & Navigation

Multiscale interaction techniques are often reported contributions across the reviewed papers. We divided the papers into four sub-class that encompass similar multiscale interaction techniques for visualizations, display devices, virtual environments, and some empirical user studies. A unique characteristic is that most papers (23) in this class contribute quantitative user study (see Figure 2.7).

 **Interaction Techniques (14/27)**: The first sub-class comprises interaction and navigation techniques for multiscale interfaces. The sub-class interaction methods are useful for locating and identifying features (12), such as shapes (9), in multiscale spaces. The papers utilize a wide range of composite visualizations, with integrated (7) and juxtaposed (5) views. Typically, authors present interaction methods on geometric (8) and tabular (5) datasets, using categorical data attributes (12). Many

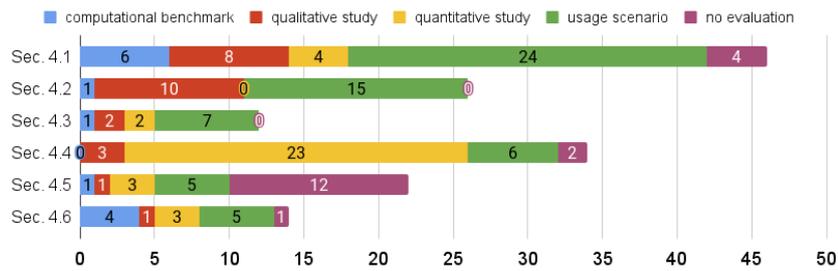


Fig. 2.7 The chart displays for the corresponding taxonomy classes the presented evaluations. Papers regularly contain multiple evaluation methods, for instance, a usage scenario and a benchmark.

sub-class papers utilize top-down navigation strategies (13). For instance, Javed et al. [161] present the PolyZoom technique to progressively build a hierarchy of focus regions that enables users to backtrack and relate multiple magnification scales.

Interaction Techniques for Display Devices (3/27): The sub-class contains multiscale interaction techniques for different display devices. The targets are to lookup geometric datasets using top-down navigation strategies. A unique paper is FingerGlass [166] which allows navigating between locations at multiple scales using multitouch screens.

Interaction in Visualization Environments (6/27): The next sub-class contains papers that describe interaction techniques for multiscale virtual environments. The sub-class targets are the identification of categorical data attributes (microscale) in geometric datasets. For example, HyperLabels [183] proposed navigational aids (labels) for the simultaneous top-down and bottom-up exploration of hierarchical molecular 3D models.

Empirical Studies (4/27): The last sub-class includes evaluation papers that assess multiscale navigation techniques. For example, Pietriga et al. [245] compare four multiscale interaction techniques (e.g., pan-zoom and constrained distortion lenses) for searching tasks. The main target is to identify and locate geometric shapes in a top-down manner. Two sub-class papers [256, 157] investigate the effect of display size in multiscale navigation. The results indicate no apparent benefit for larger display sizes [157].

Summary: The class encompasses multiscale interaction techniques, which pose new challenges as users are often lost in the multiscale information space, also known as the desert fog problem [162].

Theoretical Work

Theoretical visualization research (e.g., framework and workflows) is a central part of the contribution (22). The following class primarily contains such theory and model papers. We divided the papers into four sub-classes: multiscale visualization theory, multiscale navigation theory, frameworks, and related surveys. We only outline some outstanding papers for these theory sub-classes as there is no substantial overlap between the respective coding labels.

 **Multiscale Visualization Theory (6/22)**: The first sub-class includes theoretical papers describing multiscale information visualization's characteristics and challenges. For instance, Viola and Isenberg [306] analyze the concept of abstraction used in visualization research and emphasize the importance of multiscale visual abstractions for presenting multiscale processes in particular application domains. We included the work of Cui et al. [83] in the sub-class as the authors propose quality measurements for data abstractions. Such abstraction quality metrics are useful to assess how much the abstracted data differs from the initial data.

 **Multiscale Navigation Theory (4/22)**: The second class involves model papers about multiscale navigation theory. For example, Jul and Furnas [162] introduce the desert fog problem and further extend view navigation theory. The authors propose the critical zones concept using navigational aids to reduce and overcome the desert fog problem. Further, Guiard et al. [128] discuss the concept of multiscale pointing and introduce a framework for Fitt's law in multiscale navigation.

 **Frameworks for Multiscale Visualizations (5/22)**: The sub-class contains frameworks for multiscale information visualization. For example, Elmqvist and Fekete [98] presented a hierarchical aggregation model to turn existing visualization techniques into multiscale visualizations that scale to large datasets. The authors also describe interaction methods to analyze the aggregated hierarchy, such as drill-down and roll-up operations. In another work, Goodwin et al. [119] propose a theoretical framework for visual comparison across scale and geography. The framework allows users to explore local and global variations, including sensitivities and correlation across multiple spatial scales.

 **Surveys (7/22)**: The last sub-class includes related surveys and reviews. For instance, Cockburn et al. [79] survey interfaces for both focused and contextual viewing (e.g., overview+detail or focus+context). Such interfaces are exceptional cases of multiscale visualizations as the views display two varying magnification scales. Another recent example is the preliminary study of multiscale maps by Dumont et al. [93] that investigates how the map scale influences the displayed map

content. For example, the authors discuss how the visual complexity varies across scales, such as abstracting buildings and roads.

Summary: The class contains frameworks and workflows that propose solutions for particular multiscale visualization challenges (e.g., desert fog problem [162]).

Multiscale Visualization Systems

The last class contains technical multiscale visualization systems, which we did not further subdivide into sub-classes as there was no further plausible distinction.

 **Systems (10/10):** The papers describe scalable systems, toolkits, and architectures to enable multiscale visual analysis of large datasets. Stolte et al. [287] presents a system for multiscale visualizations using zoom graphs and data cubes operations. The targets are to identify (8) explicit target features (8), such as geometric shapes (6). The approaches utilize the statistical graphics visualization idiom (5). The systems allow selecting (8), navigating (8), and aggregating (5) tabular (7) and hierarchical (4) geometric (6) datasets. The application areas are either biological (3) or domain agnostic applications that focus on micro-, and mesoscale analysis. The system papers report a broad set of evaluation methods among usage scenarios (5). Representative papers are, for example, the Kyrix [290] and Kyrix-S [289] toolkits that provide a declarative model and grammar to create and manage pan/zoom visualization scales for large-scale datasets.

Summary: The class contains research introducing novel architectures and software solutions for multiscale visualizations (e.g., Kyrix-S [289] declarative grammar).

2.4.3 Design Considerations

We extracted seven essential design considerations based on our literature analysis. **Multiscale structures enhance visual scalability.** Researchers utilized multiscale structures with easily distinguishable and interpretable visual summaries to reduce clutter and increase the visual scalability [98]. Moreover, Visual Analytics approaches can increase visual scalability [253], such as dynamic analysis pipelines [332].

Understand relations across different scales. Users can relate data across multiple scales by using either various juxtaposed views [219] or interactive lenses [292]. Using different scales, users can progressively build multiscale hierarchies of focus regions [161] or employ space-distortion techniques to highlight multiple focus regions [100]. Noteworthy in the context are space-scale diagrams [118], which

support the understanding of multiscale interfaces.

Guide users during multiscale navigation. Researchers display context information across multiple scales to alleviate interaction overload in multiscale visualizations [331]. For example, residual landmarks across scales can be used to guide and navigate users toward interesting patterns [162]. We identified the following approaches for guiding users in multiscale environments through visual cues [162], topology-aware interaction methods [160], overview visualization [243], animations [297], and navigation viewport optimization [298].

Visualize abstraction measurements across scales. Displaying data abstraction measurements helps to assess the effects of abstraction methods and uncertainty across scales [83, 75]. For instance, comparing scale-independent aggregation measurements enables quantifying the abstraction quality across geographic scales [329].

Combining data and visual abstraction methods. The exploration of both data and visual abstraction methods reveals trade-offs and insight into sensitive abstraction parameters [287]. For example, exploring the trade-off between reducing precision and resolution reveals useful analysis scales [145].

Recursively abstract data features. Typically, abstraction methods are recursively utilized to condense information (e.g., hierarchical clustering [321]) and gradually explore data features (e.g., drill-down and roll-up operations [18]). A representative technique is ZAME [97], which uses a hierarchy to abstract and explore graph data utilizing multiple alternative visual representations.

Design tailored multiscale domain visualizations. Domain experts benefit from distinct visual encodings and adaptive interaction methods for domain-specific scales [220]. Thus, experts themselves need to select the most appropriate design from a set of abstraction and visualization methods for their tasks [227].

2.5 Research Challenges

The visual analysis of data at multiple scales poses several challenges. Understanding the emergence of data patterns across scales represents a challenge for users due to the amount of data and displayed data scales [331]. Therefore, developing more semi-automated analysis methods is essential to help users identify, compare, and relate useful analysis scales and visual representations. In this context, particularly multiscale visualizations based on dense layouts, glyphs, and spatial field visualization idioms are underrepresented. Further, new data abstraction measurements and dimensionality reduction methods can also reveal similarities and differences across abstraction scales in one view. We believe that such methods are well suited

for the multiscale exploration of the underrepresented field, network, and tree datasets. The development of such methods can likewise enhance uncertainty analysis across scales. In addition, multiscale visualizations can be distinguished into approaches for real-world multiscale environments (e.g., biological data) and large non-spatial spaces (e.g., networks), both requiring dedicated frameworks, visualization techniques, and interaction methods. Another considerable challenge is to evaluate how different composite visualizations for multiscale visualizations affect data exploration. For instance, evaluating how simultaneously displayed juxtaposed, superimposed, or integrated views of different scales influence multiscale analysis.

The interaction and navigation across scales are fundamental in multiscale visualizations, often leading to interaction overload. A unique research gap for multiscale interaction and navigation techniques are novel methods to arrange, filter, and change the data appropriately to multiple displayed scales. Such methods are notably needed if users simultaneously navigate horizontally (e.g., filtering) and vertically (e.g., aggregation). Moreover, only a few multiscale visualizations also visualize the data on a macroscale, which is potentially useful for novel user guidance methods and interactive overview visualizations. In addition, improving multiscale transition and navigation models (e.g., 3D camera management systems) are also of enormous importance for preserving the users' mental map during navigation. Researchers proposed largely top-down navigation in this context, and bottom-up navigation approaches and frameworks are rarely utilized. Further, seamlessly switching between different visual abstractions and technical devices, such as displays, tablets, and smartphones, can further advance the collaborative exploration of large multiscale information spaces.

Multiscale visualizations repeatedly claim to enhance visual scalability. For instance, the visualization of multiple abstraction scales (e.g., aggregation) allows analyzing and extracting knowledge from large datasets [98]. However, multiscale visualization scalability is typically not quantified, and existing approaches generally are not compared against each other. Hence, the comparison of computational and visual scalability of multiscale visualizations is still outstanding. A detailed trade-off analysis between data and visual abstraction methods for multiscale visualization may reveal useful information. For instance, comparing the multiscale data and visual abstraction methods in statistics and engineering will provide new insight into the scalability of the recently proposed approaches. Overall, we believe that more empirical studies are required to assess the scalability of existing multiscale visualizations, especially interaction methods for particular user tasks. Such empirical studies will considerably improve the reusability of multiscale visualizations.

A further examination of paper codings also reveals research gaps that have not been sufficiently studied. For example, multiscale machine learning applications are noticeably underrepresented in the reviewed papers. A potential solution can be to design unique bottom-up interaction methods for machine learning models that combine overview visualizations with navigational aids and annotation methods to analyze and understand the functionality of different layers and neurons in deep learning models. Overall, researchers can utilize the resulting paper codings in our online interface to identify further research gaps. For instance, an analysis of the dataset type labels unveils that "field" data is rarely used, implying that the multiscale visualization approaches for continuous fields (e.g., human magnetic resonance imaging scan) are still largely unexplored.

2.6 Discussion

In our systematic literature review, we used the results of our initial exploration of similar expressions (see Section 2.2.1) as keywords to query the search engines. However, multiscale visualization approaches might not necessarily explicitly use one of the listed keywords. We tried to resolve the issue by recursively scanning paper references and citations in both directions. Consequently, some reviewed papers do not necessarily list the expression, although the authors describe similar concepts. Further, we did not include all multiscale modeling approaches (e.g., hierarchical clustering) in visualization research since such a survey requires several additional categories (e.g., type of model construction) that reflect multiscale data models' characteristics, which is far beyond one paper's scope. Moreover, the derived taxonomy highlights only the most important design practices and research challenges. For instance, more research challenges for multiscale visualizations are reported than previously discussed. Despite all those limitations, we hope our resulting taxonomy will stimulate new multiscale visualization approaches, including new multiscale visualization theory, interaction methods, and evaluations.

2.7 Conclusion

In this chapter, we contribute a structured literature analysis of design practices in multiscale visualization research. We reviewed 122 papers with an extensive coding scheme to reveal general multiscale visualization designs, such as typical visualization idioms, targets, and interaction methods. Based on this systematic

review, we derived a taxonomy for multiscale visualizations, which describes distinct design factors and design considerations to help identify trends and gaps in research. Our results help researchers and practitioners design, present, and analyze datasets at multiple abstraction scales.

Visual Abstraction of Spatio-Temporal Networks

Summary

Collective animal behavior experts often analyze spatio-temporal networks to understand the relationships between moving animals over time. However, visualizing such spatio-temporal networks remains challenging due to the often large-scale data with fixed spatial positions and the temporal dimension, resulting in node overlap and edge clutter. In this chapter, we present MotionGlyphs, a design study that allows domain experts to visually abstract and explore dense spatio-temporal networks in the field of collective animal behavior. The proposed glyph designs enable domain experts to summarize and explore spatio-temporal network data at multiple aggregation scales. We validated the proposed design with an expert evaluation, highlighting the design's benefits and how experts can reveal patterns in the spatio-temporal network data. Moreover, the chapter presents a spatio-temporal clustering benchmark in collective animal behavior. Overall, the main goal of the proposed design is to enable multiscale visual exploration by reducing visual clutter through the abstraction of the spatio-temporal network data to glyphs.

The chapter is based on the following publications. Please refer to Section 1.5 for contribution clarifications.

- [57] **Eren Cakmak**, Hanna Schäfer, Juri Buchmüller, Johannes Fuchs, Tobias Schreck, Alex Jordan, Daniel Keim. “*MotionGlyphs: Visual Abstraction of Spatio-Temporal Networks in Collective Animal Behavior*“. Computer Graphics Forum Vol. 39. No. 3, 2020.
- [63] **Eren Cakmak**, Manuel Plank, Daniel Calovi, Alex Jordan, Daniel Keim. “*Spatio-temporal clustering benchmark for collective animal behavior*“. Proceedings of the 1st ACM SIGSPATIAL International Workshop on Animal Movement Ecology and Human Mobility, 2021.

3.1 Introduction

Collective animal behavior is an intriguing phenomenon appearing in nature in many forms. Prominent examples are the collective movement of fish schools, insect swarms, or flocks of birds [120]. Research in biology and other fields aims to explain the mechanisms by which group motion patterns emerge in natural and social sciences [81]. Such patterns can be, for instance, relationships among multiple animals (e.g., social influences), temporal trends (e.g., migrations), and sub-group behavior of animals (e.g., group of leaders). These group patterns are yet not fully understood since the movement depends strongly on influences and interactions between possibly many animals (movers) [81]. Recent research has modeled collective behavior as spatio-temporal network data to analyze the emergent properties of groups [111]. For example, Rosenthal et al. [257] analyze evolving interaction networks in which they map movers to nodes and the sensory information of a mover to weighted links (edges). A purely statistical analysis of such spatio-temporal networks (e.g., networks metrics) should be avoided as the interpretation in the context of collective animal behavior remains challenging [111]. The field, therefore, requires tailored visual metaphors to analyze the evolving network structure and highlight correlations between movers [111].

Spatio-temporal network data is a particularly challenging as it consists of evolving relationships between spatially positioned entities (attribute-driven layout) [233]. Real-world applications are, for instance, traffic [242], network security [276], and migration analysis [267]. Visualizing such data promotes identifying spatial and topological patterns over time. However, two main challenges limit the visual exploration of such evolving patterns. First, the fixed network topology of spatial networks often leads to node overlaps as well as edge crossings in dense areas [319]. Therefore, Nobre et al. [233] recommend displaying spatial networks only for small and sparse networks. Second, the additional temporal dimension poses a challenge in presenting the data in a readable, scalable, and expressive manner [31]. Visualization techniques for multivariate [233] as well as dynamic networks [31] aim to reduce the complexity of such data (e.g., aggregation [315, 94, 176] or filtering [244, 104]). Yet, such methods either change the positions of movers or reduce data characteristics (e.g., filtering), which has to be avoided in collective animal behavior analysis as it can hide potential sub-patterns and consequently decrease the interpretability [111]. An uncluttered overview visualization of spatio-temporal networks in collective animal behavior, therefore, can help domain experts analyze single movers (ego-centric) and groups of movers (socio-centric).

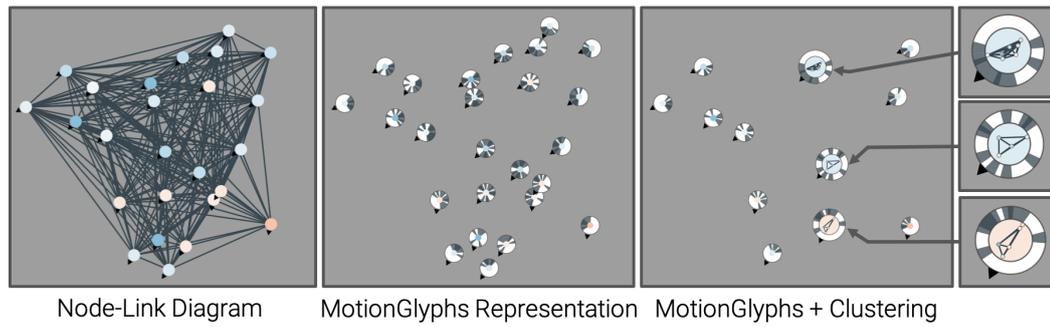


Fig. 3.1 MotionGlyphs allows biologists to visually explore and abstract dense spatio-temporal network data in collective animal behavior. The figure presents the same time instance of golden shiner fish data in a node-link diagram (left), MotionGlyphs representation (middle), and with additional clustering (right). The color of the movers displays the speed (blue to red), and the links (light blue to dark blue) encode the similarity between movement properties (direction, speed, distance to each other). The example above shows how MotionGlyphs abstract relationships and aggregate movers into groups to reduce visual clutter and highlight different group structures.

In contrast to earlier work, our MotionGlyphs prototype (see Figure 3.1) focuses on reducing visual clutter by abstracting a spatio-temporal network to glyphs. We demonstrate the usefulness of our approach by conducting expert interviews and pair analytics sessions [20]. In summary, the primary contributions of this chapter are: (1) a design study with problem characterization, findings, and lessons learned within the domain of collective animal behavior, (2) a glyph design for the summarization and depiction of spatio-temporal networks at multiple levels of granularity, (3) a visualization prototype for experts to explore local as well as global network properties over time, and (4) a spatio-temporal clustering benchmark for collective animal behavior.

3.2 Related Work & Application Background

The visual identification of patterns (e.g., clusters or trends) in spatio-temporal network data remains challenging due to the high dimensionality and the scalability issues in space, time, and network characteristics. We cover related visualization research, addressing these challenges from different perspectives in the fields of spatial, dynamic, as well as spatio-temporal network data.

3.2.1 Spatial Networks Analysis

Spatial networks (also known as geographic networks) are a way to model relationships between spatial locations. Real-world examples include the analysis of air traffic [181] and transportation data [12]. Nobre et al. [233] defined spatial network data as a special type of multivariate network data (attribute-driven layout). Multivariate network visualization can be applied to spatial networks such as Pivot-Graphs [315], Semantic Substrates [278], GraphDice [38], or dimensionality reduction [92]. However, the listed approaches focus on either node or edge (link) attribute comparisons or abstract the spatial positions. Matrix visualizations using geographical embeddings (e.g., Yang et al. [326]) are not suited for the application domain as the approaches do not scale to many time steps, and the matrices grow quadratically with the number of movers. Other visualization approaches for spatial networks aim to reduce the complexity and visual clutter by either filtering [244, 104], aggregation [315, 94, 176], clustering [97], edge bundling [202], deriving new attributes [92] (e.g., node degree), or converting edges to nodes [163]. Filtering, aggregation, clustering, and edge bundling techniques reduce the number of displayed nodes or links. However, this results in information loss, which may lead to misinterpretations in the application domain [111]. Furthermore, deriving new attributes (e.g., node metrics) can lead to misleading information [111], and the conversion of edges to nodes is not applicable in our application domain as it would produce additional movers. For spatial network visualization, Ko et al. [181] analyzed flight journeys as origin-destination data and introduced a petal glyph that displays multivariate network features. The glyph enables us to assess, for example, the number of flight or security delays for airports. However, the proposed glyph does not scale for dense areas. Zou and Brooks [336] present a visualization system to aggregate nodes into hubs, which enables to display local and global information. The authors propose a dynamic circular layout with new edge curving and node positioning algorithms. The approach is, however, unsuited for our application as the method does not allow displaying the exact spatial position or adapting the applied aggregation method.

3.2.2 Dynamic Network Visualization

Visualizing dynamic (temporal) networks has gained research interest [31]. The automatic analysis of such data enables the examination of structural properties of the network, for example, the temporal analysis of static network metrics (e.g., node degree, centrality [45]) as well as dynamic network metrics (e.g., change

centrality [113]). However, only analyzing such automatically extracted structural properties in collective animal behavior might hide specific local dynamic patterns and how such local changes affected the overall dynamic phenomena [111]. Interactive visualizations overcome these challenges by allowing users to visually analyze the changing relationships in their evolving structural context. Beck et al. [31] categorized dynamic network visualization into: animation (time-to-time mapping) [90, 248, 19], timeline (time-to-space mapping) [125, 25, 144] and hybrid visualizations [130, 32, 23]. Timeline mappings map the temporal dimension to a spatial axis (e.g., small multiples), which, however, does not scale to long sequences [31]. Other approaches from this category (e.g., NodeTrix [143]) do not preserve the position of each node (mover) over time. Similarly, the usefulness and effectiveness of animation is still controversial [296, 254]. While animation has been shown to be effective in some domains such as flow visualization [314], it does not scale to large quantities of nodes and links, often higher cognitive load [296]. For further reading, we refer to the survey of Beck et al. [31].

In summary, the current visualizations either change the positions of the movers (timeline mapping) or animate the temporal evolution of the underlying dynamic data. Therefore, the field of collective animal behavior requires new visual metaphors that combine spatial and temporal abstraction methods to reduce the presented data and highlight temporal and structural changes (e.g., cluster splitting).

3.2.3 Spatio-Temporal Network Visualization

Recently, techniques for the analysis of spatio-temporal networks (dynamic geographic networks) have been proposed (e.g., for collective movement in transport [14]). These approaches focus mainly on the study of origin-destination data. Frequently in flow map visualization, movement data is discretized to highlight the direction and magnitude of mobility patterns [12]. Kim et al. [177] propose a heatmap to display origin-destination data, which can, for example, highlight the origins of disease outbreaks. The approach, however, discards the movement (trajectory) data, which is crucial in analyzing collective animal behavior. Zhu and Guo [334] apply a hierarchical clustering method to identify significant and dense flows in the traffic data. The approach scales to large spatial data but does not scale for large time periods. Andrienko et al. [12] proposed a method for spatial and temporal abstraction, including a composite glyph to reduce clutter and occlusion in the origin-destination data. The proposed composite glyph displays for each location the flow angle and the distance between the locations to reveal regional mobility trends. The approach highlights periodic patterns by aggregating overall spatial

events and then clustering the temporal dimension into periods. A limitation is that information is lost due to spatial and temporal aggregation, and with an increasing number of spatial locations, the glyph becomes challenging to interpret.

In summary, the listed approaches for spatio-temporal networks focus on flow visualizations in specific applications, for instance, mobility trends between locations (origin-destination data) [12]. In contrast to these approaches, we focus on the visual exploration of changing relationships in collective animal behavior, for which no design studies have been carried out. In this design study, we address the needs of biologists and propose a design to tackle the challenge of visualizing evolving relationships between single movers and groups of movers.

Application Background

This design study aims to create a visual analysis design supporting the identification of group patterns over time in a large set of moving entities. We interviewed two domain experts (postdoctoral researchers) to clarify the user needs, and understand the workflow and requirements in the targeted domain. The domain experts analyze spatio-temporal networks to discover similar behavior, evolving group structures, and outliers.

3.2.4 Collective Animal Behavior

Collective animal behavior aims to understand the social influence (relations) as well as information flow between individuals and groups [81]. The research field is lately observing and tracking animal groups at larger scales in lab experiments or field studies due to technological advances (e.g., small GPS devices) [185]. Pure statistical approaches are usually used to analyze data generated by such experiments [281]. While they support verifying a single hypothesis, they are typically unable to observe potentially interesting patterns in the data which fall outside the chosen parameters and scope of the selected statistics [86]. In the research field, a lot of effort is put into revealing evolving groups (clusters) of animals that influence individual groups and vice versa how individuals affect internal group characteristics (e.g., through local influences) [81]. The analysis of influences between animals (e.g., interactions) requires methods that display the spatial data accurately and preserve local neighborhoods as this helps to follow and interpret emerging group properties [82]. Clustering local interactions enable, furthermore,

to distinguish movers with similar behavior [240] at the loss of some spatial accuracy and summarize group structures to reduce the complexity of the data. The similarity between all movers for each time step is essentially a weighted network (distance matrix). The visual exploration of such evolving similarities can reveal underlying group characteristics of collective animal behavior [86]. For example, Rosenthal et al. [257] displayed communications networks to study behavioral changes and social influences in collective evasion maneuvers. For instance, we are visually exploring a real-world dataset consisting of 151 Golden Shiner fish swimming through a depthless fish tank (2.1m x 1.2m) for 12 minutes (18000 frames). The two-dimensional dataset consists of 2.7 million data records and 18000 similarity matrices with more than 410 million links. A similarity matrix is computed using the weighted Euclidean distance between the features of a mover (see Section 3.3.1).

3.2.5 Problem Description

During the interviews, we investigated how domain experts analyze data, which tools they use, and what potential high-level problems must be addressed to understand collective animal behavior. We considered movers (nodes) with similar behavior over time in a group, for instance, the aligned movement of multiple movers towards a food source. Analyzing an appropriately constructed distance matrix (based on similarity) for each time step provides a possibility to identify groups of similar behavior and to investigate socio-centric patterns (e.g., group leaders). For the visual analysis of such socio-centric patterns in collective animal behavior, we have to address the following high-level problems (P):

P1. Display the ego-centric relationships In the application domain, it is crucial to investigate the relations of one mover to all other movers (ego-network). For example, to examine if there are similar ego-networks in space or if ego-networks increase and decrease simultaneously over time. The visual analysis of relationships between multiple evolving movers, however, remains challenging due to visual clutter in spatially dense networks [336]. A visualization of the ego-centric network, therefore, should aim to provide an uncluttered overview (summary) of such relations. A compact ego-network visualization can help domain experts to identify similar movers, compare movers, and detect outliers.

P2. Identify groups of movers with similar behavior The visualization of movers is challenging as with a growing number of movers, the clutter and overlap in dense areas increase, which can hide spatio-temporal patterns [86]. For such cases, often visual data aggregation (e.g., clustering, density maps) is applied to reduce

the number of movers [10]. We consider two types of clustering based on the spatial-temporal data and the evolving network structure. The visual analysis of such clustering methods should also involve domain experts in exploring different parameter settings for grouping elements together [12].

P3. Present the socio-centric relations in groups The display of groups of movers, for example, through a meta-node, can help to reduce the number of displayed movers and reduce clutter in dense areas. However, through such a clustering, relevant information within dense areas themselves, such as internal group dynamics, is lost [12]. The visualization of intra-cluster relationships of groups can present underlying socio-centric processes.

3.2.6 Requirements

Slingsby and van Loon [281] held a workshop with multiple animal movement ecologists and described the requirements necessary for the initial visual analysis of movement ecology. The research disciplines of movement ecology and collective animal behavior are related as both disciplines work on the analysis of collective movement [318]. In discussion with our domain experts, we selected and adapted key requirements, which are relevant for identifying group patterns in collective animal behavior, from the proposed requirements of Slingsby and van Loon [281]. We also identify related key properties a technique needs to support to satisfy these requirements, denoted in *italic* for each item.

R1: Display the original data Group patterns in collective animal behavior emerge from local spatio-temporal interactions between movers. Displaying the raw data is, therefore, essential as it helps to interpret and understand the emergent group properties. This means, the *node representation* needs to be explicit and *spatially accurate* to enable *node and neighbor comparability*. Also, since typical use cases range from small to large amounts of movers, *scalability* towards a broad range of network sizes is required.

R2: Relate the time, space, and attribute dimensions Define and present a summary of the multivariate relationships between the dimensions space, time, and attributes of a mover (node). To do so, *node exploration* by attributes and a dynamic *temporal representation* need to be provided.

R3: Enable the aggregation into groups Enable the aggregation into “ecologically-meaningful“ units, which is crucial to abstract and simplify large movement datasets.

Consequently, the technique needs to support the *cluster and subnetwork explorability and comparability*.

R4: Allow the exploration of the spatio-temporal network at different scales

Networks can be observed from an ego-centric (low-level) perspective or a socio-centric (high-level) perspective. The technique needs to support both perspectives, both for the global view and local groups (*subnetwork*).

3.2.7 Gaps in Related Approaches

In Table 3.1, we compare relevant related work using the properties defined in the requirements (see Section 3.2.6) to highlight the research gap we intend to close. We selected the related work based on recursive scanning of references from the following surveys of Vehlow et al. [301], Beck et al. [31], and Nobre et al. [233].

The Table 3.1 reveals several insights. First, approaches that scale to large networks often fail to consider the temporal dimension (**R2**), for instance, Dunn and Shneiderman, [94], or Zou et al. [336]. Second, approaches relying on static timeline representations are, on the other hand, not suited to represent live group dynamics, such as the work of Dork et al. [92], Andrienko and Andrienko. [13], or Park et al. [238]. Third, related work utilizing animation applies node or edge aggregations to reduce the visual complexity and thus introduce the loss of some spatial accuracy (**R1**) between movers, such as the approaches by Scheepens et al. [262], or Andrienko et al. [12]. The drawback of such animations is that identifying temporal trends remains challenging due to the cognitive efforts to remember changes over time [254]. Moreover, many approaches utilize aggregations to abstract and summarize the developing dynamics by disregarding the behavior of individual nodes making the identification of common behavior challenging (**R3**); for example, Andrienko et al. [12], or Yao et al. [327]. Finally, approaches including the temporal dimension often visualize only large or small networks, thus contradicting the requirement **R4**.

The comparison reveals that related work is not suited to display movers' temporal dynamics using accurate spatial node representations. Furthermore, in contrast to related work, we also utilize node aggregation and spatial clustering as crucial features of MotionGlyphs. Overall, MotionGlyphs satisfies the domain requirements at the cost of implicit edge representation and the loss of spatial accuracy to explore group structures.

Publication, Year	Spatial Rep.			Temporal Aspect			Scalability						Explorability & Comparability					Relational Rep.	
	Accurate	Inaccurate	Displaced	Animated	Timeline	None	Small Nodes	Small Edges	Large Nodes	Large Edges	Large Nodes	Large Edges	Nodes	Edges	Neighbors	Clusters	Sub-Networks	Nodes	Edges
MotionGlyphs	•			•			•	•	•	•	•	•	•	•	•	•	•	explicit	implicit
Scheepens et al. [262], 2014	•			•			•		•				•		•	•		explicit	none
Ko et al. [181], 2014	•				•		•	•					•	•	•	•		explicit	explicit
Kim et al. [177], 2017	•				•					•				•		•		implicit	aggregated
Li et al. [203], 2017	•					•			•	•	•	•		•	•	•	•	explicit	explicit
Andrienko et al. [12], 2016		•		•					•	•	•	•	•	•	•	•		aggregated	aggregated
Andrienko & Andrienko [13], 2010		•			•		•	•	•	•				•		•	•	aggregated	aggregated
Park et al. [238], 2019		•			•		•	•					•	•	•	•		explicit	order-based
Gou & Zhu [129], 2014		•			•			•		•		•		•				none	explicit
Dunne & Shneiderman [94], 2013		•				•	•	•	•	•	•	•		•	•	•	•	aggregated	aggregated
Yao et al. [327], 2019		•				•			•	•	•	•	•	•	•			aggregated	aggregated
Dörk et al. [92], 2011			•		•		•	•	•				•	•	•	•	•	explicit	explicit
Huang et al. [151], 2019			•		•		•	•					•	•	•	•	•	explicit	explicit
Zou & Brooks [336], 2019			•			•	•	•	•	•	•	•		•	•	•	•	explicit	explicit

Tab. 3.1 The table shows a comparison of related work. We sorted the rows by spatial representation and presentation of temporal aspects. The **Spatial Representation** defines the spatial accuracy, such as the accurate, inaccurately (e.g., aggregation), or displaced spatial representation. The **Temporal Aspect** defines the visual representation of the temporal dimension. The **Scalability** describes how many nodes and edges the approach can display, including small (<100), medium (<1000), and large (>1000) networks. The **Explorability and Comparability** defines if the network structures are explorable using the technique. The **Relational Representation** describes how the approaches visualize nodes and edges.

3.3 Visual Design

MotionGlyphs was designed over five months in close collaboration with two domain experts from the field of collective animal behavior. We followed the design guidelines by Lloyd and Dykes [208] to make the design process interactive, including real-world data, developed digital sketches, allowing the free exploration of prototypes and think-aloud protocols. MotionGlyphs is a web prototype to visually explore group patterns spatio-temporal network data, consisting of two components for data modeling and visualization. The data model is responsible for feature extraction (e.g., speed of a mover), computation of similarities matrices, and spatio-temporal clustering. The visual interface of the prototype (see Figure 3.2) consists of the navigation panel to change the temporal dimension, the feature panel to adapt the visual variables (e.g., clustering scale), and the glyph panel to display the single and cluster glyphs.

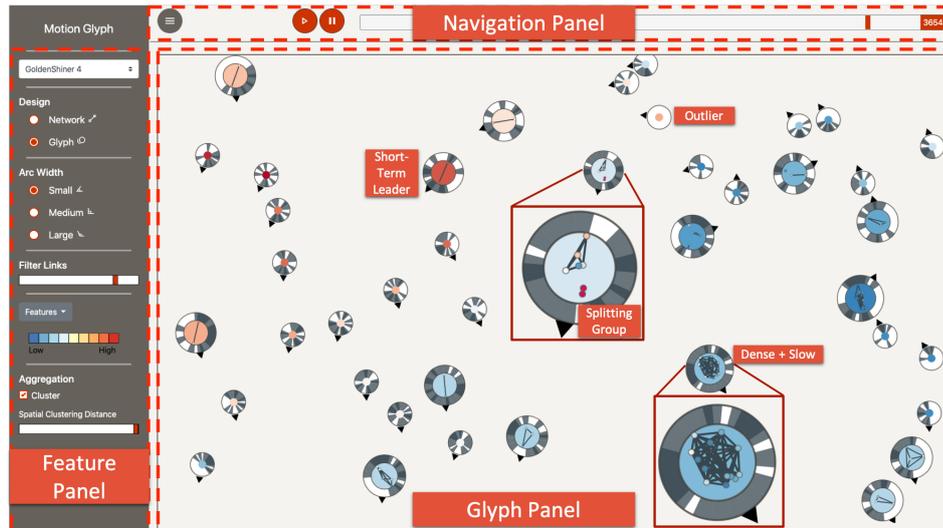


Fig. 3.2 The glyph panel shows a subset of the 151 golden shiner fish school data, the attribute color displays the speed (blue to red), and clustering is applied. The fish school has a fast (red group on the left) and a slower subset (blue groups on the right). Multiple groups with different characteristics are visible (e.g., number of movers, density, number of links), which enables us to analyze them over time. The granularity of the aggregation can also be changed at any time to allow analyzing detailed group structures or abstract movers into larger groups.

3.3.1 Data Model

We briefly describe the functionality and choices we made for the feature extraction, spatio-temporal networks, spatio-temporal clustering. The data model component models interactions between movers by enabling domain experts to compute specific evolving networks and clusters. The input file for the prototype has a standard domain-specific format (time, animal-id, x, y). Domain experts suggested data cleaning methods (e.g., interpolation) and feature extraction (e.g., average speed, direction, and distance to the centroid). For the extraction of features, domain experts have to define the temporal scales (e.g., per second, per minute), which usually depends on the tracking resolution. A network for each time step can be defined by a user-defined similarity metric based on the extracted features (e.g., weighted euclidean distance) or the segmented trajectories of the mover (e.g., Fréchet distance). Such a similarity metric can be, for instance, the weighted euclidean distance between all (or a subset) of the extracted features. Varying combinations of weights in the euclidean distance metric generate different networks, which can be used to highlight specific patterns. For example, using the direction, speed, and acceleration of each mover, the aligned movement of a group towards a particular target can be emphasized. The network for each time step includes

the temporal information as derived features (e.g., average speed) are computed using a rolling window approach. Using temporally smoothed features (e.g., average heading changes per second) improves the interpretation as noise is smoothed out (e.g., small tracking errors).

Domain experts can use either the network weights (distance matrices) or another similarity metric to compute spatio-temporal clusters. The spatio-temporal clustering helps to summarize as well as examine the temporal evolution of relationships and highlight the changes of group properties in the data. We apply the density-based clustering proposed by Peca et al. [239] as the algorithm scales to large datasets. The proposed algorithm has two parameters ϵ_{time} and ϵ_{space} , which we discussed in detail with the domain experts. By default, the ϵ_{time} is set to the temporal scale of the extracted features (e.g., average speed per minute). The clustering is applied several times with a varied ϵ_{space} , which results in clusterings with different spatial densities. The default values of ϵ_{space} are defined by the maximum distance a mover can travel between two consecutive time steps, which is a useful heuristic to determine the possible spatial changes.

3.3.2 MotionGlyphs

MotionGlyphs allows visualizing single (single glyph) and groups of movers (cluster glyph) (see Figure 3.3). The single glyph displays the spatio-temporal network using the spatial positions (geospatial-layout) (**R1**) and abstracts network links by mapping them to a radial representation (outer-ring) of the glyph. The inner-circle of the glyph allows to display characteristics of the mover (e.g., speed), and the glyph arrow depicts the movement direction (**R2**). The outer-ring of a single glyph is essentially a doughnut chart with segments (link abstraction arcs) that aim to summarize the direction and median link weights to other movers that lie in that direction. The segments preserve link characteristics, such as the direction and strength (weight) (**R2**). By default, we segment the outer-ring into 12 segments of 30 degrees. Domain experts can, furthermore, adapt at any time the segment width. The abstraction of links to segments prevents edge crossings and was inspired by the work of Ko et al. [181] in which the authors simplify origin-destination. Two color scales from ColorBrewer [138] are used to encode values: For the inner-circle attributes, a divergent color scale from blue to red is used to highlight low and high attribute values. For example, in some fish schools, the animals are continually moving; therefore, values below and above the mean speed are usually interesting for domain experts. The link weights are mapped to the outer-ring using linear light blue to dark blue color scale.

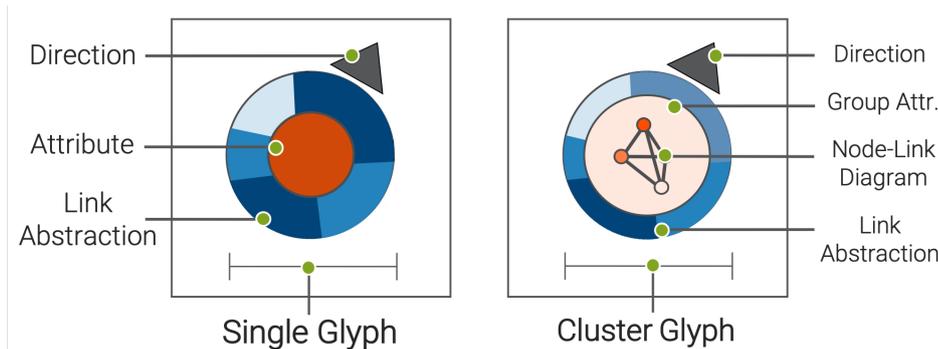


Fig. 3.3 Glyph for single movers (left) and for group of movers (right). The inner-circle display attributes (e.g., speed) and the outer-ring abstracted links to other movers.

MotionGlyphs allows to abstract groups of movers into a cluster glyph (see Figure 3.3) to present the underlying group structure (**R3**). The cluster glyph is a disjoint flat group structure visualization, which is, to the best of our knowledge, the first node glyph proposed for this category [301]. The cluster glyph size is normalized and mapped to the number of nodes in the group. The maximum size (all movers) of the cluster glyph is five times the size of a single glyph. The outer-ring of the cluster glyph displays the abstracted links to all other glyphs. The inner-circle depicts the underlying spatio-temporal network of the group as an animated node-link diagram. We visualize the underlying group structure as an additional level of detail view for cluster interpretation to allow the exploration of the data at different scales (**R4**). The spatial centroid of the group defines the position of a cluster glyph. The inner-circle also enables to display the average attributes of the group (e.g., average speed) as the background color of the inner-circle (**R2**). The node-link diagram in the center of the cluster glyph is also colored and encodes attribute information (e.g., speed) for the nodes and the links (weights) (**R2**). The color encoding in the cluster glyph allows comparing the group nodes with the average attribute values of the spatio-temporal group (**R4**). The cluster glyph also has an arrow, which indicates the average movement direction of the group (**R2**). By default, the prototype only uses the spatial positions for the spatio-temporal clustering [239] due to the preference of domain experts (**R1**). Domain experts can, furthermore, adapt and explore the spatial scale of the clustering as we pre-compute the clustering with varying input parameters (**R4**).

Design Rationale

In the following, we describe our design rationales to facilitate transferability to other domains with similar tasks and requirements.

Why are we using a glyph visualization? The complexity of spatio-temporal networks poses a challenge for the visual exploration of group patterns in collective animal behavior. Typically, methods like clustering [11], which aggregate and abstract the nodes into meta-nodes, and edge-bundling techniques [202], which display flow patterns in dense areas, are used to reduce the complexity of such data. In edge-bundling, the links between pairs of nodes are difficult to perceive [129], and the artifacts produced by such methods often lead to misinterpretations [12]. Glyph-based visualizations depict multivariate data as visual objects to enable the discovery of patterns (e.g., anomalies, clusters) [44]. A glyph maps data characteristics to visual variables providing a compact view of multivariate records and enabling comparison of the data records (e.g., star-glyph [117]). Recent approaches of Scheepens et al. [262] and Andrienko et al. [12] highlight how glyphs can be used to reduce visual clutter for scalable visualizations (e.g., through aggregation). Dunne and Shneiderman [94] also show how different glyphs can be used to improve network readability. Based on these methods, we decided, together with domain experts, to develop sketches and design a glyph [208] to reduce visual clutter and to highlight group structures in collective animal behavior.

Visual variables used: Multiple visual variables (e.g., size, color) can be used to design a glyph. We chose to keep the number of visual variables low to maximize the discriminatory factor between such visual variables. We decided to use a circle for the single glyph design and to display the temporal dimension of the data using animation. We choose not to adapt the shape and size of the single glyph as such distortions modify the spatial positions of movers (nodes) and could be misinterpreted as physical sizes of movers. We also incorporated two other visual variables, an arrow for direction, and color the circle based on movement characteristics (e.g., speed). The visual variable color (hue) is selective and associative [37]. These features are usually used in the visualization of movement data [262] and the application domain [211]. We use color and orientation as visual variables to draw attention to changes in these attributes [44]. We abstract and encode the links in an outer-ring of the glyph to summarize and highlight the relationship characteristics of a node (direction and weight). For the design choice of the outer-ring, we used the design space described by Andrienko et al. [12] and decided to use the combination of a circle and juxtaposition components (CJ flow diagram). A drawback of abstracting the links is that the detailed connection information (e.g., the distance between movers) is lost, which can be incorporated by using multiple outer-rings that also encode the distance to the target node (e.g., CJ glyph [12]). In collective animal behavior, however, showing multiple outer-rings is not useful as movers are usually uniformly distributed and retain similar distances to each other [288, Chapter 2]. We chose to keep the complexity of the glyph low and only display one ring.

Why is a cluster glyph useful? Based on the requirements **R3** and **R4**, we iteratively designed another glyph to allow domain experts to abstract movers into groups. The goal of the cluster glyph is to reduce the number of displayed glyphs, clutter in dense areas, and the cognitive load for the user. The cluster glyph, furthermore, summarize and presents the structural properties of the group, the segments of the nested glyphs, and displays internal links in such groups. We discussed with our collaborators the idea of aggregating and show multiple single glyphs in the inner-circle of a bigger group glyph (nested design) and created a digital sketch as proposed by Lloyd and Dykes [208]. This first alternative cluster design (see Figure 3.4) was complex as the nested glyphs were hardly readable and difficult to interpret since the segments of nested glyphs could be misinterpreted as links to movers outside of the group. Additionally, a minimal amount of space is required to communicate color, which is not given in such a small-sized glyph [116]. We chose to show and animate a simple node-link diagram in the inner-circle of the glyph, which downsizes and displays all the movers of the cluster. For this, we map the spatial extent of the nodes in a cluster to the inner-circle of the cluster glyph. Using such a mapping, we retain the spatial distances between movers (**R1**). The node-link diagram also encodes additional attributes (e.g., speed) and link weight (**R2**). The directional arrows for the internal node-link diagram are not displayed, as they are barely readable after mapping the movers to a smaller scale.

Why do we use animation? We display data by animation, as this is the conventional method to display temporal data in the domain of collective animal behavior (e.g., in Rosenthal et al. [257]). Visualizing the data through animation remains challenging due to change blindness [279] and our limited short-term memory [140]. We aim to overcome these challenges by reducing the number of nodes through clustering, and we highlight merges or splits of movers in groups by coloring the single glyph (merge) or a node in a cluster glyph (split) pink (0.5 seconds). The goal of the highlighting is to help experts to maintain a mental map of the changes. Identifying such group changes, such as split, merge, and swappings between groups remains challenging due to noisy real-world data and the animation speed.

How to interact with the glyph? To further facilitate the visual exploration of group patterns, MotionGlyphs enables a set of interactions. The glyph depicts the abstracted links during a mouseover to investigate the links of a specific node, which was suggested by domain experts in a free exploration of the prototype [208]. The prototype also enables filter links, limit the overall presented number of segments in the outer-ring, and modify the width (in degrees) of the displayed segments. The prototype implements a zoom and the option to adapt the spatial clustering scale using a slider (**R4**).

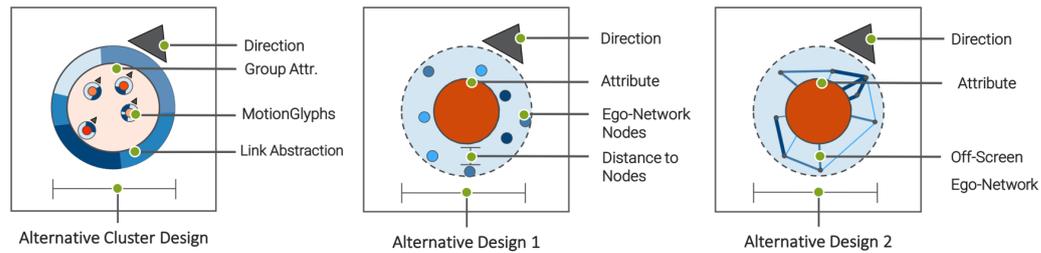


Fig. 3.4 The first design of the cluster glyph (left) and two alternative designs for single glyph (middle, right) which also enables to encode the distance information to other nodes and present the ego-network in an off-screen outer-ring.

3.3.3 Design Alternatives

Many glyph visualization techniques for either spatio-temporal or network data have been proposed. A possible design alternative to simplify the spatio-temporal network is to apply motif simplifications [94]. The approach replaces motifs in the networks (e.g., fan and cliques) with glyphs. The primary problem of motif simplification for collective animal behavior is that the interpretation of such motif glyphs over time is difficult as the approach abstracts structural motifs (e.g., fan motifs). The single glyph has a similar design as the proposed petal glyph [181], rose or sunburst diagrams [101, 272, 12] which are used to present origin-destination data [181]. The design space analysis by Andrienko et al. [12] for origin-destination data provided us with a structured way of thinking about the possibilities of abstracting links. The proposed variants of flow diagram designs examine different glyphs to reveal mobility trends between regions. The usage recommendation for the CJ glyph (circle and juxtaposition), which is similar to the single glyph design, is to highlight details for individual regions [12]. We discussed many alternative sketches and designs with domain experts to encode attributes as visual variables. For example, we explored different background colors, different hues, shapes, and the usage of small multiples. Through the usage of these digital sketches [208] we learned that the domain experts prefer rather simple glyph designs to identify behaviorally similar movers. Two examples of such design alternatives for the encoding of the links can be seen in Figure 3.4. Off-screen visualization techniques inspire the first alternative glyph in which the linked movers are mapped to circles in the outer ring. The design was inspired by the work of Farrugia et al. [112] in which they displayed ego-network neighborhoods in concentric circles, which are mapped to a time step. In contrast to a single glyph, the first design alternative animates and places the ego-network nodes based on the distance and the direction to the linked mover. The color of each node in the outer-ring encodes the weight of the abstracted link. The second design alternative displays the whole ego-network with links. The

two design alternatives allow displaying evolving ego-networks of movers in more detail. However, identifying changes and comparing values in the relatively small and complex outer-rings would have been difficult due to clutter resulting from the detailed information.

3.3.4 Design Process

We conducted contextual interviews to understand the data analysis workflow of our collaborators. During these interviews, our collaborators described examples of challenges, common features, and methods (e.g., spatio-temporal clustering) used in the domain. We identified that the main focus is to verify a single hypothesis with statistical tools, with only a few tools to display spatio-temporal data (e.g., Animal Ecology Explorer [285]). Standard network visualization tools (e.g., Gephi [27]), furthermore, have limited support for dynamic networks and do not support any abstraction methods over time. We did not find any specifically tailored visualization tools to present and analyze spatio-temporal data in the application domain. Afterward, we discussed potential abstractions methods and designs in the form of slides with our collaborators [208]. Based on the feedback we received, we implemented a prototype to visualize the spatio-temporal network by a first simple glyph design. In later iterations, we redesigned the cluster glyph based on the feedback we received and added more features (e.g., filter links) to the prototype. Finally, we conducted a user evaluation to understand how users perform real tasks.

3.4 Evaluation

We conducted audio-recorded interview sessions of 60-90 minutes with five domain experts to showcase the usability of the MotionGlyphs approach. We questioned each domain expert about their background knowledge, expectations, and first impressions of the design. Afterward, the domain experts used the MotionGlyphs prototype in a screen-recorded pair analytics session [164]. The interview concluded with a comparison of the initial expectations and the proposed design.

3.4.1 Participants

The participants (P1-P5) are researchers in collective animal behavior, with four male and one female participant. The educational background and age distribution

were one master's student, three Ph.D. researchers, and one Post-Doc, with four participants between 20-30 years and one between 30-40 years. The participants never used or saw the approach before participating in the study.

3.4.2 Dataset and Tasks

We prepared a real-world 151 golden shiner fish dataset to provide a realistic analysis scenario. During the session, the participants had to solve the following six tasks:

1. Introduction - Familiarize with all interactions using a test dataset
2. Temporal - Identify and analyze an interesting point in time
3. Spatial - Find an outlier fish and analyze its characteristics
4. Network - Find a group and analyze its characteristics
5. Find at least one meaningful single behavior pattern
6. Find at least one meaningful group behavior pattern

The participants were also motivated to use and compare the network view to the glyph view. The tasks were all conducted on the real-world 151 golden shiner dataset. The participants were encouraged to use the animation, the filtering, and the clustering feature to gain insight into real single and group behavior patterns.

3.4.3 Background and Domain Characteristics

The participants had different background knowledge. Three out of five participants studied fish behavior, and the other two participants analyzed insect or mammal behavior. All participants analyzed collective behavior with three (3) experts focusing on collective movement data. The primary tasks of all participants are the following: analyzing interactions (3), finding clusters (2), and identifying distinct groups (2). Four participants utilize custom programs to analyze and solve their analysis tasks. The participants mainly use visualizations to visually explore their experimental data (3) and present their experiments' final results (3). Essential elements of their data analysis were social interactions (5), movement metrics (2), and vision fields (2).

3.4.4 Expectations and First Impressions

The specified goals of participants for explorative visualizations are extracting essential data subsets (5), the interactive filtering of information (3), and the comparison of groups (3). Upon introducing the single and cluster glyphs design, all participants agreed that the design was easy to understand. The raised questions in this context were the glyph interpretability in collectives (2). Moreover, stated important interaction features were zooming (3) and the adaption of the glyph parameters (3). Some participants also emphasized some design similarities to their current approaches for data exploration (3).

3.4.5 Pair Analytics Session

The mentioned crucial features are the animation, the spatio-temporal clustering, and displaying the data as node-link or glyph representations. The temporal exploration using animation was crucial for the analysis. The participants suggested adapting the animation speed (4), automatically following movers or groups during animation (4), and displaying the original video synchronized with the animation (4). Moreover, the participants agreed that the network visualization is cluttered and overloaded (4), appreciating the clean design while also being able to display the edge information on hovering a glyph (2). All participants agreed that the aggregation into the cluster glyph helps identify and explore groups and outliers in dense areas. The following participant's quote highlights this: "Even when proper filtering is applied, there is no way to see the interactions of a fish in the center [of a cluster]. Then the glyph is way better. [...] In high-density formations, the glyphs are awesome. In low-density formations, the network is much more important".

The participants suggested various improvements. First, the scaling of the cluster glyph has to be adapted since the interior network structure was challenging to read (3). In addition, the multiscale clustering scale has to be adapted to include more fine-tuned variations (2). The participants also suggested adding interactive link parameterization (4), adapting the distance metric (4), and the spatial granularity (3). A commonly requested feature was labeling individuals, groups, and the temporal dimension (4). In addition, the participants suggested the automatic selection of sweet spots via distribution information (2), scaling the approach to long sequences (2), and displaying the exact values via a mouseover (2).

3.4.6 Expectation Review and Future Use

The participant's feedback was overall positive, and they solved the given tasks and identified meaningful patterns. The participants were able to detect outliers (4), large groups (3), and group transitions (3). Furthermore, some participants were able to identify leaders and followers (2), milling groups (2), and outlier subgroups re-joining the larger group (2). The participants confirmed the applicability of MotionGlyphs to their projects (5) and were also excited about the web interface (2). Moreover, the participants required additional features such as contextual spatial information (e.g., 3D or geographic maps) and exporting relevant dataset subsets for statistical analysis with other tools (2).

3.4.7 Use Case

The selected use case (see Figure 3.5) highlights the merging process of two fish groups and shows how MotionGlyphs can be used to identify structural and temporal patterns. The use case is adapted from a pair analytics session and shows the 151 Golden Shiner fish data (color mapped to speed). (A-B) display the same time moment as a node-link diagram (A) and as MotionGlyphs with clustering (B). The left group in (A) and (B) reveals how MotionGlyphs helps to reduce clutter and emphasizes movers with different behaviors (see left red box in (B)). Also, in (B), there is an apparent mover (influencer) who is going to initiate the merging process of both groups. The influencer mover leads between (B-C) a subgroup from the main group (right) towards the smaller group (left). The merging process between the two groups is reflected by the movers being added to the left cluster glyph (see merging in (C-E)), which indicates that the in-between subgroup of movers imitates the behavior from the left group. The merged group moves, afterward, towards the larger group on the right (see (E-F)). In (C-F), furthermore, a group of followers trying to catch up with the left cluster glyph is visible. The follower movers in (C-F) group accelerate, and some followers catch-up with the group and merge into the cluster glyph. However, in (F) still, two follower movers, as well as an outlier fish below, are visible, which did not yet manage to catch up and integrate into the merging cluster group. In (C-F), a fish in-between the groups is apparent, and the temporary influences on the in-between mover are visible through the as abstracted links. The in-between fish moves in (D-E) towards the left group and adapts his behavior in (F) towards the direction of the right group. In Figure 3.5 (F), the cluster granularity was also adjusted to aggregate the movers further into groups to reduce overlapping glyphs and present higher-level patterns in the merging fish swarm. The

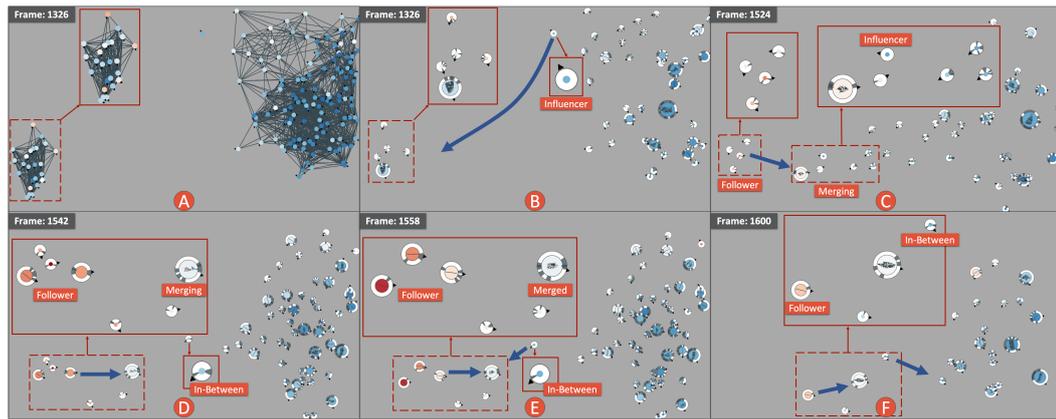


Fig. 3.5 The presented use case in Section 3.4.7 from the 151 golden shiner. The color of the glyph is mapped to the speed of movers. The time steps show how two groups merge initiated by an influencer fish. The example illustrates how the designed glyphs display relations between movers and group structures to identify patterns and generate new insight using the proposed glyphs.

use case shortly describes how MotionGlyphs can be utilized to analyze the temporal evolution of interactions and group structures in collective behavior. In the use case, more patterns are visible (e.g., outlier movers), which allows further detailed analysis to understand the influences among the movers. Experts can perform such an investigation by tracking the movers or groups over time and examining the links between them.

3.4.8 Lessons Learned

Domain experts test hypotheses and apply familiar visualizations (e.g., heatmaps) for presenting statistical results. The interactive aggregation and disaggregation of data help them to unveil behavior processes in space and time. Domain experts, however, need the original video in addition to the animation, as the individual behavioral traits of movers are also dependent on the posture and visual field of movers. The animation rate seems to influence the perceived patterns heavily and should, therefore, automatically adapt to a user-defined metric so that the animation plays faster for intervals in which the change is minimal. There was also an emphasis on including an export functionality for the data subset to verify the identified pattern with statistical tools. This shows that visual exploration and statistical analysis are seen as complementary and require new methodologies combining both perspectives.

3.5 Spatio-Temporal Clustering Benchmark

The following section present a spatio-temporal clustering benchmark for the field of collective animal behavior. The reproducible benchmark follows up on the MotionGlyphs work and compares practical spatio-temporal clustering methods. The benchmark reveals that temporal extensions of clustering algorithms are inherently useful for the detection of moving clusters in collective animal behavior.

3.5.1 Motivation

Spatio-temporal clustering is crucial for analyzing groups of moving objects in various applications, such as in transportation analysis or the study of collective animal behavior. A central task in such domains is to identify moving clusters, a group of objects moving close together in space and time [91]. However, identifying such moving clusters remains challenging due to often large-scale datasets, resulting in a trade-off between computational cost and accuracy. In addition, the performance of spatio-temporal clustering methods is rarely compared against each other, posing a challenge for users to select accurate and scalable methods.

Hence, we present a benchmark of spatio-temporal clustering in the field of collective animal behavior. The benchmark proposes 3600 datasets with varying data characteristics to compare the performance of different common spatio-temporal clustering methods. We believe that our benchmark enhances the experimental reproducibility of spatio-temporal clustering results within animal movement ecology. For the benchmark datasets and implemented methods, please refer to our online project page.¹ In summary, the main contributions of this section are: (1) A diverse set of synthetic collective behavior datasets with ground-truth, (2) a reproducible benchmark of spatio-temporal clustering algorithms, and (3) scalable implementations of spatio-temporal clustering methods.

3.5.2 Background

Collective animal behavior studies the interactions and behaviors of animal groups, exploring how local interaction rules produce behavioral patterns [178]. One of the central goals is to understand the spatio-temporal interaction rules in animal collectives and the resulting behavior, for example, in social insects [42] or fish

¹<https://github.com/eren-ck/spatio-temporal-clustering-benchmark>

schools [312]. Thus, it is essential to understand the interaction frequency and if the animals form loose or stable associations with other individuals [197]. Analyzing movement patterns helps to discover collectives, such as the clustering of flocks [127], swarms [205], or convoys [333]. We use the term moving clusters as described by Dodge et al. [91] instead of domain-specific terms, such as flocks, swarms, or convoys. Spatio-temporal clustering methods can be classified into techniques for trajectory and moving object clustering [205]. Trajectory clustering usually uses specific geometric distance metrics (e.g., dynamic time warping) to compute similarities between the trajectories and utilizes afterward standard clustering techniques (e.g., K-means) [328]. Moving object clustering discovery identifies similar movement behavior directly by adapting classical clustering algorithms to spatio-temporal data, such as the spatio-temporal extension of DBSCAN [39]. For further reading on spatio-temporal clustering, we refer to the surveys of Kisilevich et al. [179], Yuan et al. [328], Atluri et al. [21], and Ansari et al. [15].

In contrast to previous work, we present a spatio-temporal clustering benchmark comparing methods against each other to evaluate their performance. Experimental studies have usually compared spatio-temporal clustering methods against baselines on custom datasets (e.g., Agrawal et al. [3]). As there is no unified and commonly used experimental dataset and protocol, it remains challenging to compare the performance of spatio-temporal clustering methods. Therefore, we propose a benchmark for detecting moving clusters in collective animal behavior to overcome these prevailing shortcomings, focusing on generated synthetic data with ground truth, and presenting state-of-the-art baseline methods.

3.5.3 Benchmark Design

Problem Statement

Spatio-temporal clustering detects jointly moving objects in space and time without having any labels. Intuitively a moving cluster can be seen as a sequence of static spatial clusters with the objects being spatially close to each other during the whole sequence. Identifying such moving clusters is valuable for various applications in animal movement ecology, such as analyzing migrating bird flocks. In such applications, we cannot cluster the spatio-temporal data with standard clustering methods (e.g., DBSCAN [109]) due to the temporal dimension. Hence, detecting such moving clusters requires adopting clustering methods utilizing similarity metrics that partition both the spatial and temporal data dimensions. Ideally, such spatio-temporal

clustering methods are scalable to large-scale datasets, handle high-dimensional data with additional attributes, and discover arbitrary cluster shapes [3].

Dataset Design

The goal of our dataset design was to generate realistic spatio-temporal datasets with ground truth. We used three collective behavior models to generate synthetic datasets with known ground truth clusters, covering all the main features of existing models. First, Reynolds [250] proposed a model in which agents have a fixed speed and adapt their movement direction based on the separation, flocking, and wandering behavior. Second, Couzin [82] proposed another model based on three zones around an agent: the zone of repulsion, orientation, and attraction. Third, the Gautrais-Calovi model [67] is a data-driven model that describes the movement of agents using the persistent turning walker model (PTW). In this model, agents interact based on the Voronoi neighborhoods and the turning speed of each agent. We refer the reader to the respective publications for a detailed explanation and specification of three used collective behavior models. Based on the specified parameters, the three models produce different phase transitions and polarization (schooling) and vortexing (milling) behavior. The Gautrais-Calovi model [67] investigates and defines parameters for movement patterns (e.g., milling states).

We used data generation models several times with different parameters and later concatenated them to obtain datasets with clusters. We created 30 datasets with a length of 50,000 time steps, varying numbers of movers, clusters, and different input parameters for each model to obtain datasets with distinct characteristics, e.g., different cluster densities. We enclosed the exact input parameter configurations for each dataset with our datasets. Afterward, we randomly sampled from the produced 90 datasets a variety of different sized datasets, with a varying number of moving objects, clusters, and timesteps. We likewise added noise by randomly sampling and attaching single movers from the initial 90 datasets. Furthermore, we also subsampled moving objects from the original moving clusters randomly over time, for instance, we randomly subsampled 20 movers from a cluster with 50 movers. Through the sampling, we generated diverse datasets with different temporal lengths, cluster densities, uncertainty, and noise. We generated 100 datasets for the three collective behavior models with the temporal lengths of 100, 300, 600, 900, 1200, 1500, 1800, 2100, 2400, 2700, 3000, and 3500. Every dataset has up to ten clusters with up to 20 agents per cluster. The dataset sizes range from 1200 up to 520,000 data points. As a result, we obtained 3600 spatio-temporal datasets with a diverse set of data characteristics.

Evaluation Metrics

We compared the performance and execution time of the spatio-temporal clustering methods. We measured the clustering quality using the ground truth with the adjusted mutual information (AMI) score [305]. We employed the adjusted metric to ensure that uniform random label assignments will result in scores close to zero. Moreover, we captured the run-time in seconds of each algorithm for all synthetic datasets. We limited the run-time to 120 seconds, meaning that if an algorithm takes longer than two minutes, we canceled the respective clustering method.

3.5.4 Experiments

Methods

We compare two standard baseline methods against a set of temporal extensions of standard clustering algorithms. The two baselines are ST-DBSCAN [39] and trajectory clustering using Euclidean distance as a similarity measure.²

Our implemented clustering algorithms extend standard algorithms in two ways by (1) processing large-scale datasets by splitting the datasets into frames and thus (2) generalizing them to discover clusters in spatio-temporal clusters. We argue that we can adjust most clustering algorithms to accommodate both spatial and temporal features. Two approaches inspired our temporal extensions of clustering algorithms. First, ST-DBSCAN [39] uses two distance parameters to assess whether a data point is density-reachable from another data point. Second, the splitting and merging process for spatio-temporal data by Peca et al. [239]. Building on these ideas, we (1) split the dataset periodically into overlapping frames, (2) filter the distance matrix of data points using a temporal distance parameter, (3) employ standard clustering algorithms, and afterward (4) merge the results of subsequent frames. Only subsets of the original data are loaded into memory by splitting the temporal dimension into frames using a fixed time window. Two subsequent frames also always have an overlapping period to ensure that we can merge the resulting cluster labels of individual frames. We benchmark the outlined extension for the following scalable clustering methods: DBSCAN [109], agglomerative clustering, K-Means, BIRCH, and HDBSCAN [68] using the following implementations.³ Moreover, we implemented

²Using implementations of: <https://github.com/GISerWang/Spatio-temporal-Clustering> and https://github.com/cshjarry/trajectory_cluster.

³Using implementations of: <https://github.com/scikit-learn/scikit-learn> and <https://github.com/scikit-learn-contrib/hdbscan>.

spatio-temporal extensions of spectral and affinity propagation clustering methods. However, even for smaller datasets, both methods required more than two minutes of execution time.

We perform a simplistic hyperparameter search by grid search for the 3600 datasets and each clustering method. The searched parameter space examines suitable parameter combinations, which we specified based on the generated data characteristics. For instance, we determine the number of clusters for the temporal extension of K-Means using the dataset ground truth. To guarantee the successful merging of cluster labels across frames, we choose an overlap of 10% between two subsequent frames. We compare two baseline approaches against a set of scalable temporal extensions of standard clustering algorithms. We evaluate the clustering quality and execution time in detecting moving clusters.

Experimental Setup

All experiments were computed on a server with 30 CPU cores (Intel Xeon CPU E5-2640 v3 @ 2.60GHz) and 162 GB RAM. The setup with so much main memory is necessary for the baseline methods. For example, ST-DBSCAN calculates a distance matrix between all points and thus has a quadratic memory consumption. We excluded the parameter search in our run-time analysis. If the underlying clustering implementations were parallelized, we employed the parallelization functionality.

Results

Figure 3.6 shows the respective results for grouped dataset sizes. For instance, the first group *800* encompasses all data sets containing 800-3000 data points. The AMI results show that the temporal extensions of the standard algorithms and the ST-DBSCAN baseline method achieve comparable clustering quality for small datasets, for instance, less than 20,000 data points). However, the baseline ST-DBSCAN method does not scale to large datasets due to the quadratic memory consumption. Moreover, the trajectory clustering method using the Euclidean distance fails to detect moving clusters, even for smaller datasets. We identified three scalable clustering methods, specifically the temporal extensions of HDBSCAN, DBSCAN, and agglomerative clustering. For larger datasets, the AMI decreases due to increasing cluster overlaps and consequently growing merging errors for subsequent overlapping frames. Our temporal extension of ST-DBSCAN scales the furthest, up to 200,000 data points. The execution time highlights the performance

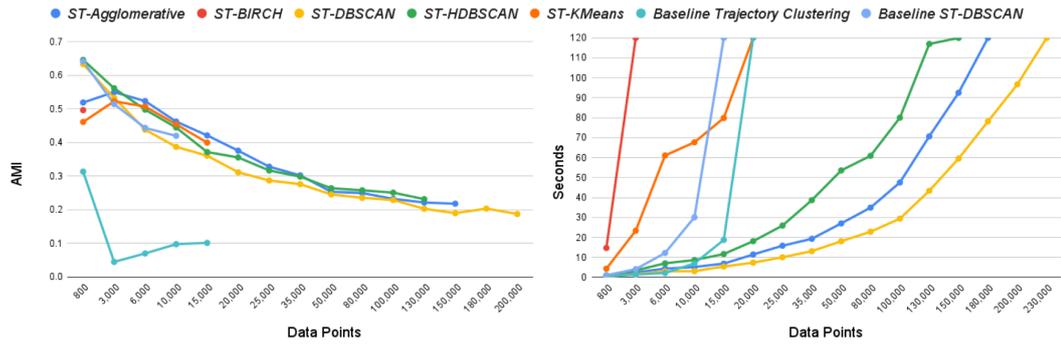


Fig. 3.6 The results of our benchmark. The average adjusted mutual information (AMI) score and the average run-time for different dataset group sizes are displayed. For example, the group 800 includes all datasets containing 800-3000 data points.

of each method. Some methods only scale to small data sets, such as the baseline ST-DBSCAN or the baseline trajectory clustering method. The standard clustering method’s temporal extensions are scaling better considering the datasets are split into subsequent frames and then merged again, requiring no quadratic memory consumption. However, the potential merging process also causes defects due to possible spatial overlaps within the overlapping frames. None of the evaluated methods detected moving clusters in datasets larger than 200,000 points within two minutes of run-time. The benchmark results show the performance of our temporal extensions of standard clustering methods is useful for detecting moving clusters in collective animal behavior datasets. We recommend using our temporal extension of ST-DBSCAN or ST-HDBSCAN to identify moving clusters if the number of clusters is not known in advance. If the number of clusters is available, we recommend agglomerative clustering. Overall, the proposed methods enable analyzing group dynamics of swarms, flocks, and other animal collectives. We believe that the implemented spatio-temporal clustering methods are crucial to detect group and sub-groups of moving clusters in collective animal behavior. For example, one can use the proposed methods to study evolving structures within collectives by exploring temporally stable and changing sub-groups.

3.5.5 Limitations

First, we mainly focused on generating and comparing 2D spatio-temporal datasets. However, our implemented clustering methods can also be used to cluster and benchmark 3D spatio-temporal datasets, such as the 3D movement of a fish shoal. Moreover, the clustering methods have several parameters that need to be specified. We tried to set the hyperparameters for our benchmark with a simple parameter

search. However, such parameters have to be set manually by the user in real-world applications. For example, the frame size strongly influences the run-time and the clustering quality, leading to the computation of larger distance matrices and possibly resulting in fewer merging errors. We also decided to use a run-time constraint of two minutes to limit the execution time of the methods. Likewise, our dataset partitioning into groups (e.g., 800-3000 data points) resulted in differing dataset numbers in each group due to random data generation and sampling. In our benchmark, we did not investigate moving clusters with a varying number of movers over time. As a result, movers cannot switch from one cluster to another. Moreover, the synthetic datasets are not modeling specific animal species but rather capture general movement patterns of collectives.

3.6 Discussion

The cornerstone of our design is the visual abstraction of spatio-temporal network links and group structures. The approach consists of the basic steps, (1) to define a spatio-temporal network based on a similarity metric, (2) the spatio-temporal clustering, and (3) the visual exploration using MotionGlyphs. There are multiple parameters to set for the steps (1-2), for example, choosing what features to use in the similarity metric and the range of spatial densities for the clustering. The meaningfulness of the network and the clustering, therefore, depends on the input parameters and the similarity metric (Euclidean or cosine distance) [249]. Many of these parameter choices must be defined by a domain expert and depend on the data characteristics (e.g., tracking resolution). We consider the flexibility of computing different networks and clusterings an advantage of our approach and a possibility for future work to explore which similarity metric works best for particular patterns (e.g., following a leader).

The choice of encoded attribute poses another challenge, as there are multiple alternative designs possible. The downside of the link abstraction is that the aggregated segments are harder to interpret and that minimal variations and segment changes are hardly readable. However, in the application domain, such minimal variations result from noise, and domain experts' primary focus is to visually identify evolving structural properties (e.g., group changes). Identifying changes (e.g., movers frequently swapping between groups) in the evolving data poses a challenge and requires further visual support (e.g., smoothing of the animation). The cluster glyph aims to reduce clutter and the number of displayed movers, however, the mapping results in a small visual space in which changes are difficult to interpret. Visual

indicators such as highlighting changes (e.g., mover leaving a group) intend to point out evolving structural properties in the group. The cluster glyph placement using the centroid distorts the positions of the individual movers and can create overlaps between groups and single movers. Such an overlap between a group and a single mover indicates that the single mover is a local outlier as the movement characteristics differ from the spatially related neighbors.

We consider two types of scalability: the network size and the number of time steps. The approach is robust to a larger group of movers (e.g., 800 movers) as the proposed glyph designs reduce the number of displayed network links. MotionGlyphs is, however, currently not fully able to cover datasets with different spatial distributions, which can be supported by applying other density-based clusterings (e.g., ST-OPTICS [3]). We also used agent-based models (e.g., Couzin et al. [82]) to investigate the temporal scalability (6000 time steps) of the approach and identified that the current prototype scales up well to 25 million network links. The glyphs are less useful in the application domain if the number of movers and links is below a certain threshold as we rely on the visual abstraction of links as well as groups.

3.7 Conclusion

In this chapter, we presented a design study to visually explore spatio-temporal networks and group structures in collective animal behavior. The result of our iterative design process is a glyph that enables us to display a visual summary of dense spatio-temporal network data, which are typically hard to visualize. MotionGlyphs is iteratively designed by a series of discussions with our collaborators. We validate our design with an expert evaluation, highlighting how the design and prototype can be used to gain insights into the underlying evolving data. We learned that the glyph design is appropriate and can be extended for various analysis use cases (e.g., context analysis). Even though the application domain motivated the design of MotionGlyphs, the design is suitable for visualizing any spatio-temporal networks. We plan to evaluate the designed glyph for similar analysis tasks in other domains, such as identifying attacks in network security. We also plan to combine a semantic zoom with a hierarchical clustering by modifying AGNES [167] to work with spatio-temporal data to split groups interactively during a semantic zoom into smaller subgroups and to adapt the proposed glyph to group sizes. Finally, we also presented a benchmark, an initial step toward practical algorithms for finding spatio-temporal clusters in collective animal behavior. We generated numerous synthetic datasets and compared the performance of different spatio-temporal clustering algorithms.

Based on our benchmark, we report that temporal extensions of standard clustering methods are valuable and scalable methods to detect moving clusters in the field of collective animal behavior.

Pixel-Based Visual Analysis of Dynamic Networks

Summary

Providing a scalable overview of dynamic networks remains challenging due to the often underlying large-scale structural changes. Previously proposed visualization approaches often apply dimensionality reduction methods based on designed feature vectors to reduce the overall complexity of the evolving data. This chapter presents *dg2pix*, a multiscale pixel-based visualization that provides a scalable overview and enables users to visually explore temporal and structural properties in long sequences of large-scale dynamic networks. *dg2pix* utilizes a multiscale temporal model, graph embeddings, and an interactive pixel-based visualization. We demonstrate the technique’s applicability through two use cases that analyze synthetic and real-world large-scale dynamic networks. *dg2pix* effectively uses the display space and presents changes in large-scale dynamic networks without overlap and clutter.

The chapter is based on the following publication. Please refer to Section 1.5 for contribution clarifications.

- [62] **Eren Cakmak**, Dominik Jäckle, Tobias Schreck, Daniel Keim. “*dg2pix: Pixel-Based Visual Analysis of Dynamic Graphs*“. Visualization in Data Science (VDS), 2020.

4.1 Introduction

Dynamic graph visualizations are used in many real-world applications to present evolving relationships between entities, such as in social network analysis. A primary user task in such dynamic graph visualizations is to obtain an overview of the temporal dimension, for instance, to identify temporal states such as trends, outliers, and similar graph structures over time [103]. However, visualizing large-scale dynamic graphs remains challenging as such visualization techniques have to present large amounts of evolving data in a readable and scalable manner [31].

Visualization techniques for dynamic graphs can be distinguished by the following primary categories: animation and timeline visualization [31]. However, both categories do not scale due to a large number of nodes, edges, and time steps [30]. Particularly, the evolving, highly complex data may pose a significant challenge for the visual detection and traceability of changes in dynamic graph visualizations. Therefore, previous approaches for the visual analysis of dynamic graph data often rely on dimensionality reduction methods to provide an overview of higher-level structures over time [103]. Such dimensionality reduction methods reduce the complexity by embedding the evolving topological structures in low-dimensional space. While to date, some dynamic graph visualization techniques leverage dimensionality reduction methods (e.g., 2D embeddings [103]), they still fail to provide a scalable overview of the structural changes as the approaches depend on the temporal analysis scale and the designed feature vector (e.g., graph metrics).

We propose *dg2pix* (dynamic graph to pixel-based visualization), a novel visualization technique for large-scale dynamic graphs based on unsupervised graph learning methods (e.g., graph2vec [230] or FGSD [303]). The main goal is to provide a scalable overview of the temporal dimension and enable the initial exploration of the high-dimensional data to support the identification of temporal changes and similar temporal states. The visualization technique consists of three main steps: multiscale temporal modeling, graph embeddings, and a pixel-based visualization. The graph embedding reduces the dynamic graph to a low-dimensional representation (50-300 dimensions) and learns the similarity between graphs to capture the evolving topology of the high-dimensional data. The compact visualization technique allows users to interactively adapt the temporal analysis scale and compare high-level as well as fine-grained structural changes. We demonstrate the usefulness of our approach through two use cases to show how *dg2pix* can be utilized to identify temporal changes and states in dynamic graphs.

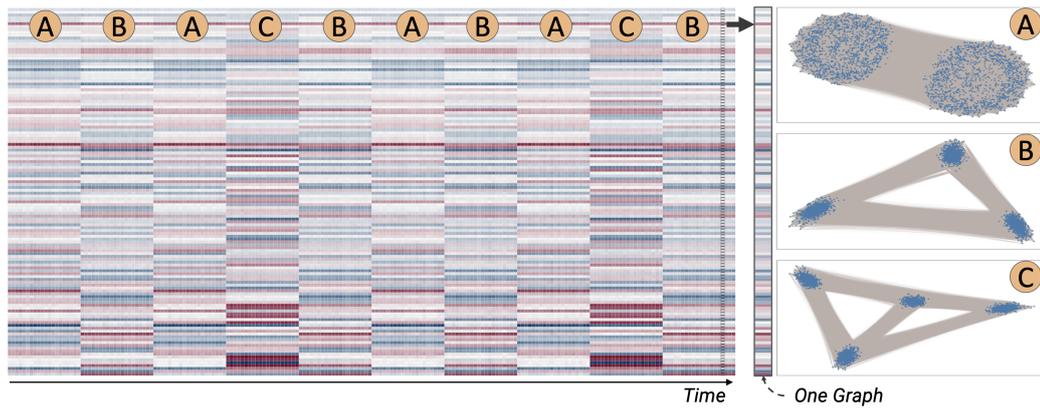


Fig. 4.1 The example presents a synthetic dynamic graph (200-time steps) using `graph2vec` [230]. The x-axis presents the temporal dimension, and the y-axis displays for each time step a graph embedding as a pixel-bar. The reoccurring states (A-C) have for each time step the same amount of nodes (2500) and edges (350000) with a different number of clusters. Each state (20-time steps) was generated with SBM [147] with slight variations for the density of edges between clusters. The graphs (A-C) display a sample graph for each state.

In summary, the contributions of this chapter are the following: (1) The novel *dg2pix* visualization technique, a time-scalable visual metaphor to reveal changes and similar temporal states in a dynamic graph; (2) an interpretation strategy of visual patterns that users can examine in *dg2pix*; and (3) an interactive prototype that allows exploring dynamic graphs at multiple scales.

4.2 Related Work

We briefly discuss related work from dynamic graph visualizations, the visual analysis of dimensionality reduction methods, and pixel-based visualization techniques.

4.2.1 Dynamic Graph Visualization

In many application domains, dynamic graph visualization techniques have recently gained more research attention [31]. Such techniques can be classified into two main categories: animation and timeline visualizations [31]. The animation of large-scale dynamic graphs is often regarded as inadequate due to the cognitive efforts to maintain a mental map [248] and trace changes [296]. On the other hand, timeline visualizations often map the graph to a compact representation to reduce cognitive efforts and enable the comparison of periods. For instance, the parallel

edge splatting technique [55] displays dynamic graphs as a sequence of bipartite graph layouts. However, even in the improved version that uses the interleaving concept [52], the identification of temporal patterns remains challenging due to the over-drawing problems between the individual graphs. Further, Van den Elzen et al. [102] propose to extend massive sequence views and suggest different reordering strategies to minimize block overlaps. Nevertheless, identifying temporal patterns in dense and large-scale graphs remains challenging due to the overlapping edges, making it difficult to trace changes in the linear-ordering. An extensive survey of further dynamic graph visualization techniques can be found in the surveys of Kerracher et al. [174], Beck et al. [31], and Nobre et al.[233].

In summary, dynamic graph visualizations such as animations and timeline mappings do not scale to long sequences of large-scale graphs due to limited display space [31]. Therefore, previous visualization approaches apply dimensionality reduction methods to reduce the complexity of the high-dimensional graph data.

4.2.2 Dimension-Reduced Dynamic Graph Visualization

Visualization approaches based on dimensionality reduction focus on summarizing and abstracting dynamic graphs to highlight temporal and structural changes. For example, Van den Elzen et al. [103] use dimensionality reduction methods (e.g., t-SNE [213]) to reduce the amount of data and provide an overview of high-level temporal states in a dynamic graph. The proposed visual analytics approach, however, depends on the temporal discretization scale and requires feature engineering for embedding the discretized intervals into vectors. Time curves [26] likewise embed the temporal data in a spatial layout to highlight temporal patterns and anomalies. Still, time curves heavily depend on feature engineering of the vectors, the quality of the distance metric, and often produce visual clutter for long sequences due to overlapping issues. For further reading, we refer to the survey Engel et al. [105], Sacha et al. [260], and the work of Vernier et al. [304].

Overall, dimensionality reduction methods reduce the complexity of the dynamic graph data and support the identification of temporal patterns. However, such methods frequently fail to capture structural changes as the methods heavily depend on the designed feature vector (e.g., graph metrics). Furthermore, such methods do not scale to long sequences due to the visual clutter caused by overlaps in the spatial layout. Pixel-based visualization techniques can be utilized to avoid such over-drawing problems.

4.2.3 Pixel-Based Visualization

Pixel-based visualizations effectively use the whole display space and allow us to present large amounts of data without overlap and clutter [170]. For example, Buchmueller et al. [48] highlighted the usefulness of pixel-based visualizations for the visual summarization of changes in spatio-temporal data. There are, however, only a few pixel-based visualizations for dynamic graphs. For instance, a matrix of pixel-based glyphs can be used to highlight temporal patterns in small social networks [286]. Furthermore, space-filling temporal treemap visualizations [182] can be extended to display medium-sized evolving trees in a pixel-based visualization manner. Such temporal treemaps [182] and other hierarchy based visual metaphors [53] are, however, only able to depict evolving hierarchies. Another pixel-based timeline visualization is GraphFlow [84], which displays graph metrics to highlight structural changes in a dynamic graph. The GraphFlow [84] method, however, only works for small graphs with a limited number of time steps, and the energy-based visualization also heavily depends on the used graph metric (e.g., node degree) that can fail to capture the overall dynamic phenomena.

4.2.4 Delineation to our Work

This chapter proposes an overlap free multiscale pixel-based visualization that does not require any feature engineering, scales up to long sequences of graphs, and enables us to drill down into aggregated temporal intervals. We utilize unsupervised analysis methods (graph embeddings) from the field of machine learning to automatically learn and embed graph structures in low-dimensional space without requiring any features engineering [330]. Such graph embedding methods stand in contrast to previous analysis methods (e.g., GraphFlow [84]) that typically used static or dynamic graph metrics (e.g., diameter [45] or change centrality [113]). However, there are currently no visualization techniques that leverage graph embeddings for dynamic graphs even though they have shown to be efficient for various tasks (e.g., in link prediction [123]). Our approach was inspired by the stripe-based visualizations of word embeddings that can be used to highlight semantically similar word groups [275]. Contrary to previous approaches, *dg2pix* scales to long and large-scale dynamic graphs, providing an encompassing overview of possible temporal aggregation levels. This chapter also investigates how the visual patterns in *dg2pix* can be interpreted.

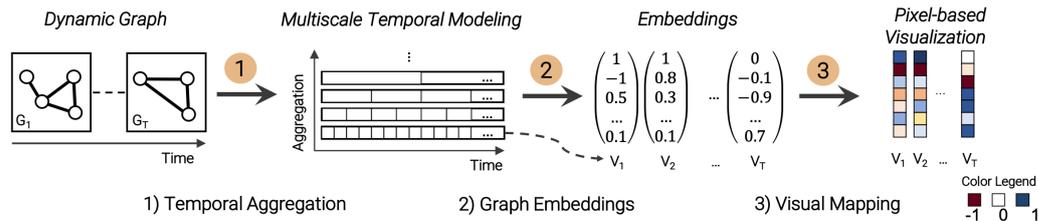


Fig. 4.2 The *dg2pix* enables users to discover similar temporal states. The approach consists of three main adjustable steps (1) temporal aggregations, (2) graph embeddings, and (3) the visual mapping to the pixel-based visualization.

4.3 Dynamic Network to Pixel-Based Visualization

dg2pix is a scalable visualization technique to gain an overview of the temporal dimension in long and large-scale dynamic graphs. The approach combines temporal aggregations with dimensionality reduction methods (graph embeddings) at multiple temporal scales to reveal temporal patterns, for instance, reoccurring states with similar graph structures. With *dg2pix*, we show how graph embeddings can be interactively used to surpass state-of-the-art visualization techniques for dynamic graphs by the amount of displayed information.

The visualization technique consists of three adjustable steps (see Figure 4.2). The technique’s basic idea is to use graph embeddings combined with a pixel-based visualization to present vast amounts of high-dimensional data to support the exploration and summarization of dynamic graphs [46]. The (2-3) transformation steps (see Figure 4.2) are, to the best of our knowledge, not considered in previous literature for dynamic graph visualizations even though graph embeddings outperforming many state-of-the-art unsupervised learning methods [330], and pixel-based visualizations enable to display large amounts of data overlap and clutter-free [170]. Next, we explain the three steps of *dg2pix* and present the implemented prototype.

4.3.1 Multiscale Temporal Modelling

Temporal abstraction methods (e.g., temporal aggregations) are applied to dynamic graphs to reduce the amount of data and summarize the changes over time. Typically, such temporal abstraction methods are based on aggregation, such as the supergraph computation, which summarizes intervals by unifying all nodes and edges of a sequence of graphs [131]. Supergraphs provide an overview of temporal intervals by summarizing a sequence of graphs into only one graph with the cost of discarding temporal information [36]. The computation of supergraphs can be seen as a

discretization of the temporal dimension. However, the usefulness of such a temporal discretization depends on many aspects, for instance, graph size, frequency of topological changes, and the temporal aggregation scale [36]. For example, a fine-grained temporal aggregation into supergraphs results in various intervals with little information and is unable to provide an overview. In contrast, coarse-scale aggregation produces only a few supergraphs, containing a high variance, where important intervals may remain hidden. Moreover, finding the optimal fixed interval length for analysis depends on the user task at hand [88].

We compute supergraphs at different temporal aggregation scales to enable users to explore temporal states at multiple temporal granularities interactively. In contrast to previous discretizations of time that use uniform or non-uniform temporal granularities (time-slices) [311], we propose to recursively partition the data using uniform time-slicing methods and compute for each interval a supergraph. For example, the recursive supergraph generation can be done based on the cyclic division of time, such as the division into a year, months, and days. We propose the following default dynamic graph coarsening approach for domains with no reasonable temporal partitioning. The default approach slices T time steps of the temporal dimension recursively into intervals of length 2^l with l being the level $l \in 1, \dots, \lceil \log(T) \rceil$. The resulting levels contain at the lowest level one intervals of length one and the highest level $\lceil \log(T) \rceil$ a supergraph of all graphs. We compute a supergraph for each of these intervals, which results in $\lceil T/2^{l-1} \rceil$ supergraphs for each the level l . The multiscale temporal modeling computes $\lceil \log(T) \rceil$ levels of granularity having overall $(2 \cdot T) + 1$ supergraphs.

The temporal multiscale modeling essentially recursively coarsens the dynamic graph into supergraphs, which are used in combination with the original evolving graphs in the next step to learn the similarities between graphs in a latent space. Our multiscale temporal modeling was inspired by Elmqvist and Fekete [99] hierarchical aggregation, which enables us to turn visualizations into multiscale (multiresolution) approaches that scale better to large datasets. The multiscale temporal modeling is later used to perform unbalanced drill-down and roll-up operations.

4.3.2 Graph Embedding

In the second step, we apply dimensionality reduction methods (graph embeddings) to all generated snapshots to learn the similarities between graphs and reduce the high-dimensional data to low-dimensional vectors. We apply graph embeddings (e.g., graph2vec [230]) as they are scalable to large-scale dynamic graphs, outperform

state-of-the-art methods in the field of unsupervised learning, do not require feature engineering, and are small enough to fit into main memory for interactive visual analysis [330]. For example, graph2vec outperforms graph kernels, and substructure embedding approaches for classification tasks on large graph datasets [230]. A graph embedding can be seen as a function $f : V \rightarrow \mathbb{R}^d$ that maps a set of vertices (e.g., random walks) to a d dimensional embedding. Typically, the embeddings of the latent space \mathbb{R}^d are used to gain insight into the data and for further standard machine learning tasks. For example, we cluster the embeddings to visualize and gain an overview of similar temporal states. In contrast to previous dynamic graphs visualization approaches (e.g., van Elzen et al. [103]), which depend strongly on the used graph metric, unsupervised graph embeddings do not require any feature engineering, are task agnostic and data-driven. An advantage of graph embeddings is that the methods learn similarities between graphs in the latent space by approximating different graph metrics [43]. Furthermore, we employ graph embeddings instead of node embeddings because graph embeddings only compute one vector for a given time step and therefore scale to large datasets.

We utilize the three recently proposed graph embeddings [258] for our approach: graph2vec [230], GL2Vec [73], and FGSD [303] as the approaches have moderate run-time complexities. We compute embeddings of all $2T + 1$ supergraphs of the multiscale temporal modeling step and embed the graphs, as suggested by Bonner et al. [43], into the range of 50 – 300 dimensions. Per default, we embed each graph to a vector of 128 features and applied L_2 normalization to the embeddings. The normalization maps the vectors to unit length and enables us to use cosine similarity instead of the dot product as a distance measurement [201]. The vectors are later displayed in the pixel-based visualization as pixel-bars to identify changes in dynamic graphs visually. In Figure 4.5, three different graph embeddings of a synthetic dataset are presented, highlighting that the proposed methods capture temporal states in dynamic graphs. In our discussion (see Section 4.6), we elaborate on the input parameters and the scalability of such graph embeddings.

4.3.3 Pixel-based Visualization

In the last step, we visually encode the embeddings into dense pixel-based visualizations to provide an overview of the temporal dimension and reveal similar graphs. We particularly visualize the embeddings as they are compact encodings of the structural information of each graph. The embeddings are displayed as linearized pixel-bars that are basically grid-based columns in which each rectangle (pixel) is a feature of the embedded vector. The technique colors each pixel by the feature's

value by using a diverging color scheme from ColorBrewer [138]. We utilize a global segmented color scheme with two distinct values to support the comparison task [291]. We sequentially order each displayed pixel-bar (graph embedding) by time, creating a dense pixel-based visualization. The y-axis ordering of the colored pixel-bars is per default, based on the linear order of the vector. The challenge of finding a useful linear order to highlight particular patterns visually can be mapped to the linearization problem [34]. We are utilizing different reordering algorithms to improve the global ordering of the embeddings to emphasize different patterns along both axes of the pixel-based visualization. For example, we apply clustering algorithms to all displayed data features to group and arrange similar features over time. We discuss different reordering strategies in Section 4.4.3.

We utilize the computed supergraphs of the multiscale temporal modeling to present the data at multiple user-defined levels of temporal aggregation (see Figure 4.4). Such a multiscale (multiresolution) visualization helps to set detailed abstraction levels into the overall temporal context [99]. For example, the visualization technique presents 1000 supergraphs as pixel-bars instead of several thousand individual graphs and enables users to drill-down into intervals. We limit the number of depicted grid-based columns to the available horizontal pixels of the screen space to address our approach’s visual scalability, which means that the minimum width of a pixel-bar is precisely one pixel. If a user drills down and reaches the limit of screen space pixel, he has to coarsen temporal intervals to reduce the number of overall displayed pixel-bars. Next, we describe our implemented prototype.

4.3.4 Prototype

The *dg2pix* prototype implementation ¹ enables us to explore the temporal changes of large scale dynamic graphs. In the following, we briefly introduce the two main linked views of the prototype.

The ***dg2pix* view** (see Figure 4.3) consists of a *toolbar* (A), a *zoom context bar* (B), and the *pixel-based visualization* (C). The *toolbar* allows selecting and presenting various graph embeddings for particular datasets, including choosing different training epochs and applying automated analysis methods. For example, the x-axis can be reordered based on a clustering of the graph embeddings (see Section 4.4.3). Furthermore, the toolbar enables us to change the temporal granularity of intervals (drill-down and roll-up) and display selected graph embeddings in the graph view. The *zoom context bar* presents additional information for the vertical and horizontal

¹<https://github.com/eren-ck/dg2pix>



Fig. 4.3 The *dg2pix* component consists of three views a toolbar (A), the zoom context bar (B), and the pixel-based visualization (C). In (C) the embeddings (GL2Vec [73]) of a synthetic dynamic graph (SBM[147]) with reoccurring states are depicted. The displayed embeddings of the x-axis are clustered (HDBSCAN [68]), and y-axis ordering is based on the median of each vector attribute. The reordering and clustering of the synthetically created reoccurring states highlight large clusters of similar graphs and outliers in the temporal data.

temporal navigation and provides an overview of the displayed temporal interval and granularities. The view (see also Figure 4.4) displays for each pixel-bar the corresponding temporal granularity as a zoom bar (rectangle). The height is mapped to the zoom level, meaning the zoom bars of low levels of temporal granularity are rather small. The zoom bars are always ordered by time and enable us to relate the potentially reordered pixel-bars to their overall temporal context via brushing and linking. The zoom context bar also allows for selecting and filtering periods of the pixel-based visualization, allowing navigating horizontally along the temporal dimension. The *pixel-based visualization* displays per default the medium zoom level of the graph embeddings ordered by time. The view allows us to select individual and multiple pixel-bars and adapt the temporal granularity by drilling-down a lower temporal granularity or coarsening the temporal dimension (roll-up). The view is also directly linked to the zoom context bar, enabling us to keep an overview and relate the pixel-bars to the temporal dimension. The x- and y-axis of the pixel-based visualization can also be reordered using different reordering strategies to highlight clusters and similarities between the embeddings (see Section 4.4.3). Furthermore, multiple pixel-bars can be selected to display the underlying supergraphs and graphs in the second main view.



Fig. 4.4 The zoom context bar enables us to investigate the zoom level for an individual and multiple pixel bars. Further, it allows filtering time intervals for vertical and horizontal navigation.

The **graph view** allows us to display the underlying graph data of the selected pixel-bars as a supergraph to highlight and compare the intersections and disjoint nodes and edges between the graphs in the temporal data. The supergraph nodes and edges are colored using two graph set operations on all selected time steps to highlight similarities and differences. The applied set operations are intersection (orange) and disjoint (blue) on all nodes and edges of the selected time steps. The goal of the set operation comparisons is to investigate the changes in the temporal graph, which helps to identify and interpret which graph structures were preserved in the latent space. The view uses per default for all time steps one precomputed graph layout (Fruchterman-Reingold [114]) by computing a supergraph for the whole dynamic graph. We facilitate one global layout to preserve the user's mental map [30]. The graph view can also be explored via semantic zooming to explore particular graph structures (e.g., node and link attributes).

4.4 Visual Interpretation

dg2pix provides a scalable overview by emphasizing the underlying changes in a dynamic graph. The main idea of the approach is to learn and display low-dimensional embeddings of graphs that capture the similarity between graphs in a latent space. However, interpreting such embeddings in the latent space remains challenging as the meaning of particular numeric values cannot be directly mapped to topological features of the graph. For example, the specific meaning of a dimension value of 0.3 of an embedding with 128 dimensions remains unanswered. Consequently, the abstractness of what low and high values of each dimension encode poses a challenge to understand and map the patterns in *dg2pix* to topological changes in the evolving graph. In previous work, typically, 2D visualizations are used to interpret and understand such latent space [206]. For instance, the Embedding Projector [282] by Google Brain uses projections (e.g., t-SNE [213]) to present word embeddings as 2D and 3D scatterplots. However, such simple 2D visualizations discard latent space information as the d -dimensional embeddings are again reduced

into 2D embeddings for the visual representation. The following section describes the underlying challenges of visualizing latent spaces, interpreting visual patterns, and different reordering strategies to highlight temporal changes.

4.4.1 Latent Space Visualizations

Recently, the visual analysis of latent spaces (embedding spaces) has gained research interest [206]. For example, ad-hoc dimensionality reduction methods (e.g., PCA, t-SNE [213], or UMAP [218]) are often applied to display neighbors in the latent space in 2D space. The latent space representation's central goal is to provide more insight into the underlying embedded data and enable the qualitative interpretation of the learned embeddings [206]. Heimler and Gleicher [141], for instance, describe tasks for word embeddings and display words in a matrix-based view to highlight co-occurrences between words. Further, Liu et al. [206] describe a set of tasks for exploring latent spaces and present a cartography system to visually investigate relationships between data points and compare attributes of vectors (e.g., word embeddings). The visual analysis of latent spaces currently remains the primary method to investigate and interpret graph embeddings. There has been little theoretical work to prove that such embeddings approximate and learn different graph metrics [43]. For example, EmbeddingVis [204] enables the comparison of different latent spaces of node embeddings to investigate which node metrics are preserved by applying regression.

In contrast to all previous approaches, our primary goal is to generate a visual summary of the temporal dimension that helps to understand and highlight temporal states in the evolving data. We display the embeddings with all their dimensions to visually compare similarities and apply reordering strategies to present changes in the latent space. Our approach also allows us to present the underlying graphs in combination using graph set operations (e.g., union or intersection) to help interpret and compare the latent space with the original evolving graph data. Next, we elaborate on how *dg2pix* can be interpreted, and automatic approaches can be used to find similar temporal states.

4.4.2 Interpretation of Visual Patterns

Graph embeddings are machine learning models that produce abstract low-dimensional vector representations for graphs that are difficult to interpret, as the individual values of the dimension themselves have no exact interpretation [261].

Challenges The reasons for interpretation challenges arise from the stochastic algorithms (e.g., graph2vec [230]), which utilize non-transparent neural networks with hyperparameters [261]. Further, the embeddings can be changed with unitary rotation, completely transforming each dimension's values while preserving the latent space distances. Therefore, the complexity of interpreting graph embedding dimensions can be compared to the efforts to understand activations in neural networks for image classification [261]. Nevertheless, recent experiments [261, 43] indicate that graph embedding methods learn to approximate various topological features of graphs. Therefore, we utilize and visualize graph embeddings to highlight changes in dynamic graphs as the methods have shown to be effective feature spaces for various graph mining tasks, such as classification of graphs [303, 293, 73].

Interpretation The pixel-based visualization enables us to perceive similarities and differences between embeddings to provide an overview of the dynamic graph. Generally, the visualization of embeddings can reveal relationships in the latent space, as shown by Shin et al. [275] for the comparison of semantically similar word embeddings. The graph embeddings can only be interpreted in relation to other embeddings by investigating the pairwise similarity between embeddings. More specifically, if two subsequent graph embeddings in the dynamic graph are, to some extent, similar to each other, then the original graphs are also similar to one another. Also, vice versa, if two successive embeddings are different, then the two embedded graphs are dissimilar to some extent. Therefore, we can use the embeddings to examine and highlight changes and temporal states in a dynamic graph even though we cannot interpret the individual values of particular dimensions.

Visual Comparison of Embeddings Consequently, the human-centric visual analysis of temporal states (e.g., reoccurring graphs) can be mapped to distinguishing similar pixel-bars in the $dg2pix$. For instance, Figure 4.3 displays a large block of similar pixel-bars with an apparent outlier in-between. The visual analysis of pairwise similarities between pixel-bars enables identifying temporal changes and states in the underlying dynamic graph. However, the cognitive efforts to compare multiple pixel-bars are high since the user has to simultaneously relate numerous dimensions of different embeddings. The pairwise similarities between multiple embeddings can also be computed using the cosine similarity. We, therefore, propose to use automatic methods to sort and cluster similar rows and columns in the pixel-based visualization to enable the identification of temporal states (e.g., outliers).

Explainability We also compare the underlying graph structures of embeddings in the graph view against each other, intending to generate new insight into the latent space. For instance, displaying the graph data helps explain graph features' potential

reasons and impacts on particular values for individual dimensions. Overall, both the pixel-based visualization and the graph view can help to understand and explain the semantic meaning of high and low values of particular dimensions to gain new insight into graph embeddings, which are black-box models [261].

4.4.3 Reordering Strategies

dg2pix was designed to scale to large-scale dynamic graphs and provide a visual summary of the temporal data. However, temporal states can remain hidden and difficult to identify due to the sheer amount of visualized data, for instance, if single reoccurring pixel columns correlate with other prominent states. Applying different reordering strategies to the embedding can reveal such otherwise hidden temporal states. For example, clustering reordering the displayed pixel columns (x-axis) will highlight similar graph structures. Therefore, we provide users with the option to apply reordering strategies to reveal similar patterns along both axes.

We provide global reordering strategies for the dimensions of the embeddings (y-axis) and the temporal dimension (x-axis). In general, identifying an optimal ordering for our pixel-based visualization is known to be NP-Hard since the issue can be mapped to the problem of reordering (linearization) of rows and columns in matrices [34]. For the reordering of matrices, various reordering strategies (layouts) have been proposed to highlight different patterns (e.g., block patterns [34]). We provide for the reordering of the embedding dimensions (y-axis) several heuristics based on a statistical metric of each row. For example, before the L_2 normalization, the y-axis can be sorted by the median value for each dimension of the displayed embeddings to highlight block and band patterns [34]. Furthermore, the prototype allows us to reorder the dimensions (rows) of the pixel-based visualization using the mean, minimum, maximum, variance, and standard deviation of the depicted rows.

We also provide two reordering strategies for the temporal dimension (x-axis) to identify similar temporal states by computing clusters and reordering based on the distances to one particular column (similarity search). The clustering uses HDBSCAN [68] for the displayed embeddings facilitating the cosine-similarity as a distance measurement. We employ HDBSCAN [68] as the approach aims to find the result with the best stability over different epsilons parameters and accordingly detect clusters with varying densities. The clustering results are displayed by grouping and highlighting the pixel-bars according to their clusters using a grey bounding box. For instance, the clusters are reordered using the median time of all embeddings, and the underlying embeddings of a cluster are again sorted by time. Second, we

enable to reorder the y-axis based on the distance to a particular embedding. The resorting places an embedding to the first position and afterward ranks the presented embeddings by the distance to the selected embedding (similarity search). This reordering enables us to compare one particular embedding in time with all other graph embeddings in detail.

Overall, using such reordering strategies for both axes can help users understand how the ordering influences the visual patterns, can group, and rank similar temporal states to explore the latent space in more detail.

4.5 Evaluation

In the following section, we apply *dg2pix* to synthetic and a real-world dynamic graph to demonstrate how the approach can be used to gain an overview and provide insight into the temporal changes and reoccurring states in evolving graphs.

4.5.1 Synthetic Dynamic Graphs

We generated synthetic dynamic graphs, with known ground truths, to show the applicability and the usefulness of *dg2pix*. For example, we created different datasets with the Stochastic Block Model (SBM) [147] with a fixed amount of nodes for each time step, a varying number of edges, and multiple temporal states (see Figure 4.1). We elaborate on the results of one dynamic graph to show how the approach can be used to identify states in large-scale graphs. The synthetic dynamic graph consists of 1000 time steps, 1000 nodes, more than 30 million edges, and three reoccurring temporal states. We facilitated the SBM to create three states with different numbers of clusters (blocks), a slightly varying number of nodes (up to 50) per cluster, and minor edge density changes (internal and external). The dynamic graph consists of randomly shuffled data of 500-time steps with two clusters of nodes, 250-time steps with three clusters, and 250-time steps with four clusters. The dynamic graph was embedded with three different graph embeddings with the following parameters:

- **graph2vec** [230]: 1000 epochs, 0.02 learning rate, 2 Weisfeiler-Lehman iterations, and 128 dimensions.
- **GL2Vec** [73]: 1000 epochs, 0.02 learning rate, and 128 dimensions.
- **FGSD** [303]: 128 number of histogram bins with the histogram range of 20.

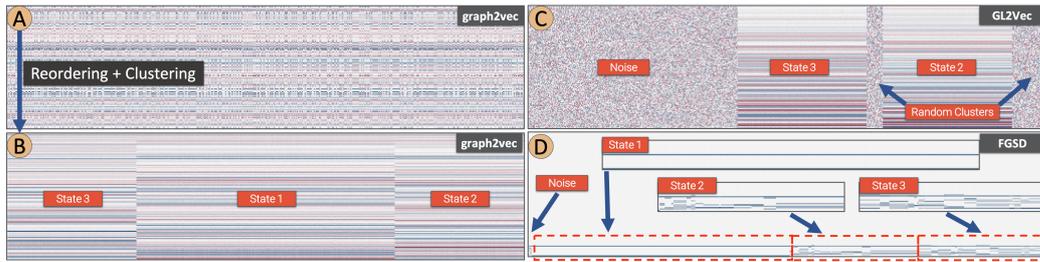


Fig. 4.5 The synthetic dynamic graph described in the use cases (see Section 4.5.1) displays three different graph embeddings with a ground truth of three temporal states. (A) presents the dynamic graph using graph2vec [230], and (B) shows the same data with the three temporal states. In (B-D), the same reordering strategies were applied to highlight the temporal states. The two other graph embeddings, (C) GL2Vec [73] and (D) FGSD [303], are partially able to learn and highlight the three temporal states in the synthetic dynamic graph.

The Figure 4.5 (A-D) shows the resulting *dg2pix* of the synthetic dynamic graph. In Figure 4.5 (A), the randomly shuffled data is displayed using the graph2vec [230] embeddings, and in (B), the same pixel-bars are presented after the application of reordering strategies. We reordered the embeddings (x-axis) based on the clustering of the embeddings (HDBSCAN [68]), and the rows were globally sorted based on the standard deviation of each row (ascending). The reordering strategies help to identify temporal and reoccurring states (e.g., clusters) by grouping similar and dissimilar pixel-bars and their respective rows together. For example, sorting the rows by the standard deviation of each row allows users to compare and identify the embedding dimensions that primarily distinguish temporal states. In Figure 4.5 (B), the three temporal states are visible, which can be verified by displaying the underlying graph structures in the graph view. Accordingly, graph2vec has learned the temporal states encoded in the underlying ground truth.

In contrast, GL2Vec [73] was not able to distinguish the three temporal states (see Figure 4.5 (C)). The same reordering strategies result in only two visible temporal states. The GL2Vec learned to distinguish the states with the three and four clusters, however, the model was not able to distinguish the larger group of two clusters (500-time steps) in the latent space. The clustering grouped the first temporal state as the visible block of noise and identified two similar states in the ground truth as two different clusters. The GL2Vec potentially requires a different learning rate or more epochs to distinguish the third state in the latent space.

In Figure 4.5 (D), the FGSD [303] is displayed which approximately learns the three temporal states. Compared to the first two methods, the FGSD model embeds the dimensions only to a positive range (blue color), and only seven dimensions of the embeddings contain values. The method is almost able to distinguish all three

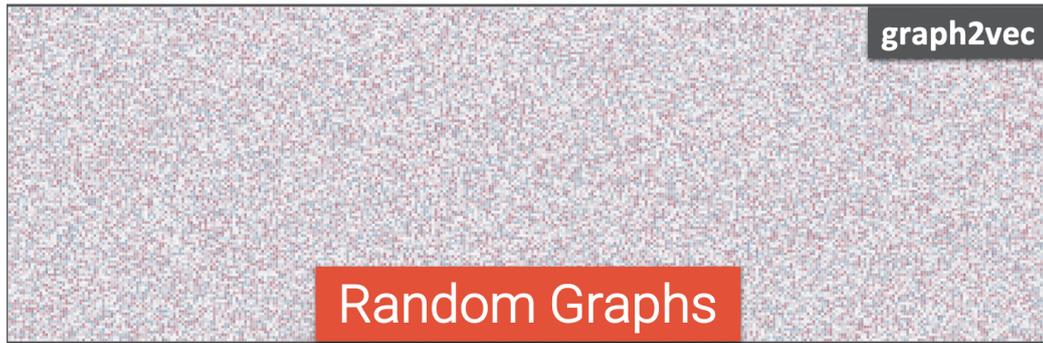


Fig. 4.6 The synthetic random dynamic graph contains no known ground truth, and the proposed approach produces no visible patterns.

clusters except for a little bit of noise, which can be verified by visualizing the graphs in the graph view. In contrast to the other graph embeddings, the FGSD model results in considerable white space that can be removed by deleting rows that do not contain any values.

In addition to the different synthetic graphs with known ground truth, we also created random dynamic graphs with different graph generators to confirm that the visible patterns are not arbitrarily learned in the latent space during the training process. For instance, Figure 4.6 shows a dynamic graph with 1000 randomly generated connected Watts–Strogatz small-world graph [317] with 2000 nodes (between 5-50 nearest neighbors), and < 0.1 edge probability edges for each time step. The same reordering strategies, as in Figure 4.5, were applied, and the resulting *dg2pix* shows the *graph2vec* (1000 epochs) embedding, which does not contain any visible patterns as the model was not able to learn the similarities between the random graphs in latent space.

4.5.2 Evolving Social Network

Next, *dg2pix* is applied to a real-world, large-scale social network. We describe the temporal visual analysis of the website Reddit [188] to discover structural and temporal changes as well as reoccurring states between social network communities (subreddits) during the 2016 US presidential elections. In the following, we describe the analyzed dataset, highlight the main task and challenges for the analysis of such data, and how *dg2pix* can be used to provide an overview of the temporal changes.

Reddit Data Reddit is a social news aggregation website with approximately 440 million users as of 2020. The website is made up of subreddits in which users post content (e.g., images or links to news sites) and upvote posts based on a voting based system to rank interesting content for each subreddit. The dataset [188] is a dynamic hyperlink graph and consists of subreddits (nodes), and time-stamped hyperlinks (edges). The analyzed data contains hyperlink graphs grouped by hours from the 1st January 2016 to 30th November 2016 in which the election campaign for the 2016 presidential election took place. The dynamic graph consists of 7974 graphs, 18546 nodes (subreddits), and 88328 edges (hyperlinks) between the subreddits with either positive or negative sentiment. We computed the following three graph embeddings graph2vec [230], GL2Vec [73], and FGSD [303] with the same input parameters as described in Section 4.5.1. We verified the resulting insight by comparing the identified changes and states of the underlying evolving hyperlink graphs to the real historic news coverage of the presidential elections.

Tasks and Challenges The visual analysis of social network data aims to provide an overview of structural changes over time, temporal states (e.g., reoccurring graph structures), and outlier graphs in the evolving data (e.g., political scandals). However, gaining an overview of large-scale social media data is challenging as it requires visualizing structural and temporal changes simultaneously and identifying suitable temporal analysis scales for changes and states of varying temporal length. Furthermore, the size and complexity of social networks pose another challenge in visualizing the evolving data since there is a trade-off between the visualization of the detailed graph structure for each time step and presenting the overall evolving graph properties. For instance, animations display each graph of the data in detail, however, animations are considered to be unsuited to provide an overview of long periods due to cognitive efforts to keep track of changes [296]. In contrast to previous approaches, we model and embed dynamic graphs at multiple temporal scales to enable the multiscale temporal analysis of long as well as large-scale dynamic graphs.

2016 US Presidential Election We begin by investigating the week before and the week during the 2016 US presidential elections (8th November 2016) to identify graphs with political subreddits in the temporal data. Per default, the prototype displays 400 pixel-bars of the middle level of temporal granularity using the graph2vec [230] embeddings. First, we use the multiscale temporal modeling to concentrate on the election weeks in November 2016. We aggregate the pixel-bars before October into aggregated supergraphs (roll-up), and further split (drill-down) the election weeks into the lowest temporal granularity of one hour. We display different graph embeddings to examine the resulting pixel-bars during the election

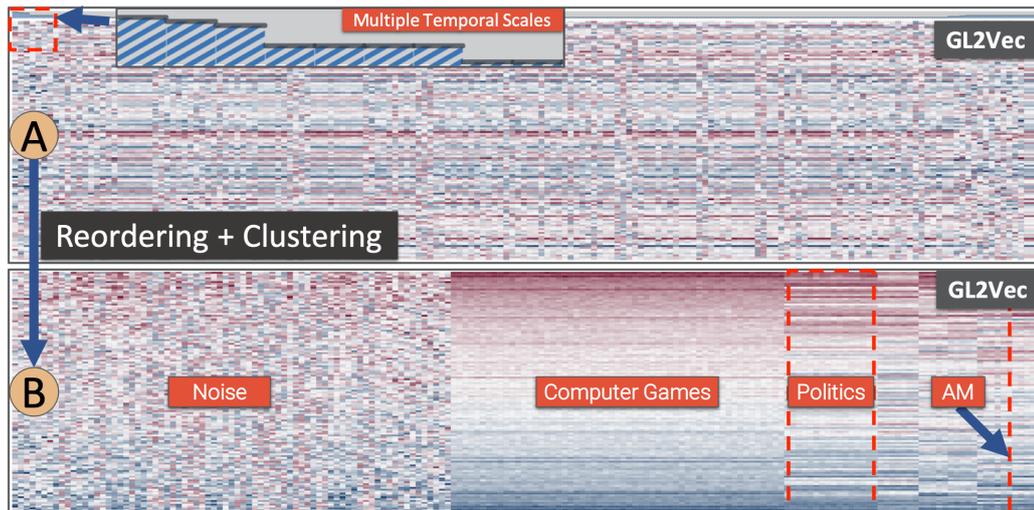


Fig. 4.7 The Reddit data described in the use cases (see Section 4.5.2) presents the election week of the 2016 US presidential election using GL2Vec [73] embeddings. The multiscale temporal modeling was used to display drill-into the election week and aggregate other intervals into supergraphs. (A) displays the evolving social networks sorted by time, and (B) shows the same data after applying reordering strategies to emphasize temporal states. We linked the clusters of embeddings to hyperlinks between different communities of subreddits, for example, computer games related topics, political topologies, or morning graphs structures (AM).

week period visually. We decide to use the GL2Vec [73] embeddings, as there are some noticeable similar pixel-bars in the $dg2pix$ (see Figure 4.7 (A)) in which the x-axis is sorted by time. Next, we apply the implemented reordering strategies to group and highlight similar pixels-bars. The median of each row reorders the y-axis, and we cluster and reorder the embeddings of the x-axis (see Figure 4.7 (B)). The first visibly large group of graph embeddings is classified as noise as the embeddings seem to have distinct values in the latent space. The next groups are clustered together and also have visually similar looking embeddings. We investigated the graphs in groups and between groups by displaying and comparing them in the graph view. Thereby, we interpreted and tried to link the embedding characteristics to evolving graph structures. For example, we noticed that the first group consists of many computer games subreddits (e.g., *pokemongo*) and that the following group contains various political subreddits (e.g., *the_donald*, *AskTrumpSupporters*, or *politics*). We were also able to identify graph structures related to specific time aspects. For example, the last group (AM) consists of hyperlinks posted only in the morning (between 8-11 am). These graphs posted in the morning have specific characteristics (e.g., fewer subreddits) that have been learned by the graph embedding.

Searching for Political Events Next, we search for political events during the 2016 presidential election to identify graph structures with hyperlinks between political subreddits. First, we change the temporal granularity of all embeddings to the duration of 8 hours, which results in approximately 1000 pixel-bars. We select the election night of 8th of November (6 pm - 12 am). We assume that political subreddits, which posted hyperlinks to other subreddits during the election night, were also active during the election campaign. Afterward, we use the ranking functionality to search in all three graph embeddings for similar embeddings, and we examine the top results. The top five-nearest neighbors in the three graph embeddings reveal different political events. Graph2vec and GL2Vec return the 1st February can be directly linked to Iowa’s democratic and republican caucus. Other graphs resulting from the similarity search can be related to the democratic nomination of Hillary Clinton (28th July) and Mike Pence being announced as the running mate of Donald Trump (15th July). Moreover, FGSD [303] ranks the 23rd July high, which can be associated with the Wikileaks email release that revealed a bias of the Democratic Party against Bernie Sanders. The publications of Wikileaks are particularly visible in the graph view, as some political subreddits are linked (e.g., *SandersForPresident*, *politics*, *political_revolution*). Overall, the use cases describe how *dg2pix* identifies temporal changes and states (e.g., political events) and relates the latent space to structural changes in the underlying graph.

4.6 Discussion

The cornerstone of *dg2pix* are the three steps: (1) the multiscale temporal modeling, (2) graph embeddings, and (3) the visual analysis of the pixel-based visualization. Next, we discuss the limitations of *dg2pix* and potential future research directions.

Parameters The (1-2) step has multiple input parameters that profoundly influence the perceived patterns in the pixel-based visualization, such as the latent space size, number of epochs, or the random initialization of the neural network. Currently, the parameter choices are set by the user as they depend on many factors, for example, the temporal aggregation depends on the discretization scale of the application domain. We consider the usage of various parameters as an advantage of our approach and a possibility for future work to investigate which parameter combinations (e.g., different graph embeddings) can capture distinct temporal changes, such as reoccurring motifs or outlier graphs.

Interpretability The interpretation of the resulting perceivable changes remains challenging due to multiple reasons (see Section 4.4), which affects the usability of

the approach as the visual encoding is challenging to read. We consider the interpretation limitation minor as our approach focuses mainly on highlighting temporal changes. However, we aim to support the latent space’s visual analysis by presenting the underlying embedded graph structures, enabling us to generate new insight into the evolving data and lead to new interpretations. We also offer reordering strategies to examine and interpret neighborhoods and clusters of embeddings in the latent space. Nevertheless, the extension with further contextual features (e.g., evolving graph metrics) is essential to allow a detailed interpretation and guide users towards interesting patterns.

Graph Embeddings We apply unsupervised graph embeddings to reduce the dimensionality of long sequences of dynamic graphs and automatically learn similarities between large-scale graphs. In contrast to topological graph metrics (e.g., density), such unsupervised graph embeddings scale to large graphs, do not require any feature engineering, and are domain as well as task agnostic. The main limitation of such embeddings is that it remains unclear how many embedding dimensions are required to capture specific structural changes [261]. We plan to investigate the required number of dimensions for synthetic temporal patterns and how different input parameters and noise influence the resulting embeddings.

Scalability For the computational scalability, we consider the graph size ($|V|$ nodes and $|E|$ edges) and the number of time steps T . The (1) step computes supergraphs at multiple levels and requires $O(\log(T) \cdot (|V| + |E|))$ memory and time complexity. The computation of the supergraphs can be parallelized to increase the approach’s scalability to long sequences of graphs. For further reading of runtime complexities of graph embeddings, we refer to the survey of Goyal and Ferrara [123], which emphasizes that recent graph embeddings run in $O(|E|)$. Therefore, the overall runtime complexity of the approach is $O(\log(T) \cdot (|V| + |E|))$. Due to the time and memory complexities, we suggest precomputing the embeddings for large-scale dynamic graphs on GPU servers. Once the embeddings have been calculated, they are small enough to fit into the main memory. Second, the computational efforts affect the interactive visual analysis of the *dg2pix*. For example, the reordering strategy by clustering scales linearly to the displayed time steps and embedding dimensions. Also, the visualization of large-scale graphs for the comparison and interpretation in the graph view does not scale to large-scale graphs as the size impairs the node-link diagram’s readability. A possible solution for this issue is to cluster the underlying large-scale graphs and display the identified clusters. However, such a clustering makes it challenging to compare graphs as nodes and edges are abstracted into meta-nodes. Therefore, we plan to examine how different graph embeddings, combined with evolving graph metrics, can be used to compare large-scale graphs.

4.7 Conclusion

We presented *dg2pix*, a visualization technique to provide an overview of temporal changes in long and large-scale dynamic graphs. The novel representation consists of multiscale temporal modeling, unsupervised graph embeddings, and a dense pixel-based visualization to explore the embeddings at different temporal scales. The main idea is visually analyze the latent space to identify temporal changes in the dynamic graph. The implemented prototype and the use cases show how *dg2pix* can be used to provide insight into evolving graphs and highlight the approach's applicability to synthetic and real-world dynamic graph data. Overall, the *dg2pix* is a promising new research direction for dynamic graphs and can be generalized for the visual analysis of unsupervised embedding methods and latent spaces.

Motif-Based Dynamic Network Visualization

Summary

Providing a scalable overview of dynamic networks remains challenging due to the underlying large-scale elusive topological changes. Therefore, previous dynamic network visualizations frequently utilize abstraction methods to provide a high-level overview of topological changes. This chapter presents two complementary pixel visualizations based on motif and graphlet analysis to provide a multiscale time-scalable overview of dynamic networks. The pixel-based visualizations allow identifying, comparing, tracing, and interpreting structural similarities between evolving network structures to reveal similar temporal states, trends, and outliers in dynamic networks. Moreover, we discuss the identification of visual patterns in both pixel-based visualizations, also considering different reordering strategies to emphasize such visual patterns. We showcase the approach's usefulness through use cases analyzing synthetic and real-world large-scale dynamic networks, such as the evolving social networks of Reddit or Facebook. Overall, this chapter presents a visualization approach to provide a scalable overview of significant sub-structural changes in dynamic networks.

The chapter is based on the following publication. Please refer to Section 1.5 for contribution clarifications.

- [59] **Eren Cakmak**, Johannes Fuchs, Dominik Jäckle, Tobias Schreck, Ulrik Brandes, Daniel Keim. “*Motif-Based Visual Analysis of Dynamic Networks*“. Submitted to Visualization in Data Science (VDS), 2022.

5.1 Introduction

Many data exploration and analysis problems rely on network representations in one form or another. Besides the visual analysis of static networks, i.e., where nodes and relationships are fixed, in many domains *dynamic* network data arises. These, in turn, pose challenging questions about the change of the network structure, features, and patterns over time. For example, social networks, computer networks, or transportation networks change over time. While first examples of visual analysis of dynamic networks have recently explored real-world applications [31], such as in biology [130] or communication analyses [131], it remains a challenging problem.

A typical user task in such applications is to obtain an overview by identifying similar and dissimilar network structures over time to gain an understanding of topological changes [103]. However, providing a scalable overview of changing network structures remains challenging due to large-scale network data that usually evolve over long periods. For instance, the growing linkage behavior of the social network Reddit [188] consists of five years of data with roughly 55K nodes and 850K edges. Previous dynamic network visualizations, therefore, regularly utilize abstraction methods to reduce the complexity and provide a high-level overview of temporal changes [103]. However, such abstraction methods depend on the graph size, the frequency of changes, and the extracted global or local metrics (e.g., diameter or node degrees). A promising approach is a local analysis of sub-networks (e.g., motifs or graphlets) that define and provide insight into complex network topologies [299]. However, in visualization research, sub-networks are mainly used to abstract and display static networks. For instance, Dunne and Shneiderman [94] display motifs as simplified glyph representations. To this day, dynamic network visualizations refrained from using motifs or graphlets, although they can provide useful, scalable overviews of evolving sub-networks.

In this chapter, we propose two complementary scalable pixel visualizations [170] to provide an overview of changing motif structures in large-scale dynamic networks. The first pixel-based representation of *network-level census* displays significantly occurring motifs to reveal structural changes, trends, states, and outliers. The visualization allows users to compare topological structures within and across several dynamic networks. Moreover, we propose a second linked pixel-based representation of *node-level sub-network metrics* that presents detailed node neighborhood information and allows us to compare individual networks within a dynamic network in more detail. We introduce potential visual patterns and discuss different reordering strategies to emphasize visual patterns, for instance, rearranging the pixel

representations based on network metrics to highlight similar network superfamilies. We likewise display a node-link diagram juxtaposed to relate the visual patterns with the network topology. To improve the scalability of our approach, we apply clustering to find superfamilies of similar networks and node neighborhoods. Both pixel-based visualizations allow identifying, comparing, tracing, and interpreting similar evolving network topologies to understand evolving sub-network structures in dynamic networks. We demonstrate the usefulness of our approach through use cases analyzing synthetic and real-world datasets.

The main contributions are: (1) we discuss and exploit the possibilities of a motif analysis to provide an overview of significant topological changes in dynamic networks, (2) we visualize the results with two linked pixel visualizations and discuss the applicability of reordering strategies to emphasize visual patterns, and (3) we implement a prototype to evaluate the usefulness of our approach in a use case analysis with real-world data.

5.2 Related Work

In the following section, we first provide background information on dynamic networks, motifs, and graphlets. Then, we discuss related static and dynamic network motif-based visualization approaches. The discussed research is selected based on back and forward search using the surveys of Kerracher et al. [174], Borgo et al. [300], Beck et al. [31], Nobre et al. [233], and Ribeiro et al. [251]. Finally, we compare and delineate our work from related approaches.

5.2.1 Background

Dynamic networks model evolving relationships between real-world entities in various application domains, such as social network analysis. A dynamic network DN can be defined as a series of T static graphs $DN = (N_1, N_2, \dots, N_T)$. Where each network $N_i = (V_i, E_i)$ at the time step i consists of a set of vertices or nodes V and a set of directed edges $E \subseteq V \times V$. In this chapter, we follow the common visualization terminology and use the term network to describe graphs in which nodes and edges have attributes [309]. Next, since our approach analyzes motifs, network census, and graphlets, we briefly introduce these terms.

Motifs are significantly over-represented directed sub-networks that are often regarded as the basic building blocks of a network [225]. Network motifs are crucial

in various application domains to analyze topological structures, such as in co-authorship networks [76] or brain networks [284]. Generally, a network motif is a distinct sub-network that occurs more often than expected in a random reference network model [225]. Likewise, motifs that are significantly under-represented are considered to be anti-motifs. We consider motifs as induced sub-networks, meaning that all existing edges between the sub-network nodes are included.

Network censuses (motif significance profiles) help to compare different-sized networks [224]. The census is computed by counting the specific number of motifs m_i in a network N^{real} and normalizing occurrences of the motifs to a set of randomized networks N^{rand} with the same degree sequence. The statistical significance is defined as $Z_i = (N_i^{real} - N_i^{rand}) / std(N_i^{rand})$ with N_i^{real} being the real number of motif occurrences, and N_i^{rand} the occurrence of the motif in a randomized network. By default, we use the configuration model [232, Chapter 4] as a null model to create random networks for the computation of the network census. The normalized significance profile is defined as follows: $SP_i = Z_i / \sqrt{\sum_j Z_j^2}$. SP_i indicates the relative significance of the motif m_i compared to the frequency of the same motif m_i in a randomly generated network with the same degree sequence. The values of SP_i are between $[-1, 1]$, with $SP_i = 1$ indicating that the motif is significantly over-represented, and inversely $SP_i = -1$ defines an anti-motif, meaning that the motif is significantly under-represented. Furthermore, thirteen triad motifs without self-loops are often used to compute the network census [224]. We also facilitate these triads since the motifs capture the lowest level of social structures, considering relations between three nodes [148]. The triads are crucial to study social networks, such as triadic closures or transitivity [210]. Such triads are also mainly used in statistical models for dynamic networks, such as the stochastic actor-oriented models [283] and temporal exponential random graph models [136].

Graphlets are non-isomorphic induced undirected sub-networks without the concept of significance and over-representation [247]. Graphlets can be used to calculate the topological similarity between nodes from different networks [222]. Graphlets are connected sub-networks and capture the instances of induced motifs occurring in the neighborhood of a node. The graphlets for two to five nodes around a particular node are known as the 73-dimensional graphlet degree vector (GDV) [139]. GDVs enable us to compare the topological similarity between nodes and are essentially the neighborhood signature of four hops around a given node. There are various graphlet counting algorithms with reasonable runtimes, such as the orbit counting (ORCA) algorithm [146]. Please refer to the recent survey of Ribeiro et al. [251] for a more detailed introduction to motif and graphlet detection algorithms.

5.2.2 Network Motif Visualizations

In visualization research, motifs are typically used to display static networks. However, in such scenarios, the concept of over-representation is not taken into account. For instance, Dunne and Shneiderman [94] simplify and depict fan or clique motifs as glyph representations. Other related static network visualizations are utilizing motif-based features to explore motif frequencies in biological networks [268], to cluster networks [193], to display large signaling networks [212], to visually search networks [192], to explore biological mutation graphs [199], or to structurally explore large networks [74]. For example, EgoNav [137] enables users to explore summarized ego-networks utilizing motif analysis and dimensionality reduction methods. Moreover, Kwon et al. [190] used graphlet frequencies to compute the similarity between networks. As for network matrix visualizations, motifs are visual patterns in a matrix, such as a line pattern. For instance, *HiPiler* [198] allows users to explore matrix snippets (motifs) in genome interaction matrices. All of the aforementioned motif-based visualizations are for static networks. However, the listed static network visualizations are not suitable for providing a scalable overview of changes in dynamic networks. The work of von Landesberger et al. [194] is a unique system in this category. The proposed system allows users to aggregate user-specified motifs and highlight local motif changes by utilizing a what-if-analysis. However, the system is not suited for analyzing changes in dynamic networks since the system only allows investigating the impact of individual differences on local motif structures. Although the utility of motif-based visualizations is well-known for static networks, they have not been utilized to present dynamic networks.

5.2.3 Dynamic Network Visualizations

Previous dynamic network visualizations often display the evolving data as timeline visualizations to reduce the complexity and provide a high-level overview of temporal changes in dynamic networks [31]. For example, van Elzen et al. [103] apply dimensionality reduction methods to embed dynamic network snapshots to connected points in a 2D scatterplot. However, such dimensionality-based abstraction methods depend on the graph size, the frequency of changes, the used distance metric, the extracted global or local metrics (e.g., node degrees), and the non-linear dimensionality reduction methods. Moreover, van den Elzen et al. [102] extend *Massive Sequence Views* to visually explore dynamic networks, including simple sub-network structures and communities, such as star patterns. However, the approach does not support visual detection of distinct motifs, such as feed-forward loops, which are of

primary interest in gene regulator networks [273]. Likewise, the approach does not scale well to an increasing number of nodes and networks, causing more overlap and clutter, making the visual detection of sub-networks challenging. Hadlak et al. [131] cluster domain attributes to detect groups of nodes with similar trends and behavior. However, the approach only clusters time-varying node and edge attributes, creating clusters without considering or utilizing the network topology. Bach et al. [24] propose *GraphDiaries* utilizing animated transitions to navigate and highlight changes in a dynamic network. Yet, animations are not suitable for displaying large-scale dynamic networks due to the high cognitive effort to compare and trace changes over time [296]. Other dynamic visualization approaches utilize persistent homology [132] or display dynamic networks on large physical displays [195]. However, again both approaches do not allow exploring motif-based changes in dynamic networks. Recently, Xie et al. [322] proposed *MeasureFlow* to explore time-series of network metrics (e.g., network density) to provide an overview of changes in dynamic networks. The approach also enables tracking and comparing trends of user-defined sub-networks using metrics (e.g., number of connected nodes) as superimposed line and bar charts. Yet, the approach's usefulness depends on the user-selected sub-networks, the network size, and the frequency of changes, including the selected temporal granularity. *MeasureFlow* also does not scale to a large number of motifs since every motif requires a single line chart. For further readings, please refer to the surveys of Kerracher et al. [174] and Beck et al. [31].

Recently, visualization researchers proposed initial pixel-based visualizations for dynamic networks. Pixel visualizations can present large amounts of data without overlap and clutter [170], being dense and ultimately able to scale to large datasets. They are generally useful, among others, for visual explorations of groups, trends, correlations, and outliers in large datasets [35]. Only a few pixel-based dynamic network visualizations have been proposed, which we present next in chronological order. First, Stein et al. [286] proposed pixel-based glyphs to present temporal patterns in an adjacency matrix. The proposed method works only for small social networks and does not allow motif exploration. Second, Burch et al. [55] proposed the parallel edge splatting approach to display a series of static as bipartite layouts, including the interleaving concept [52] to increase the approaches scalability. However, the proposed approaches are only helpful for visually exploring edges and their attributes. Next, Cui et al. [84] proposed *GraphFlow* to display structural changes of metrics in dynamic networks using a pixel and energy-based visualization. However, the *GraphFlow* method depends on the node metric (e.g., node degree) and can only display smaller networks. Archambault and Hurley [17] present a design study to highlight trends in telecommunication networks as pixel-oriented visualizations,

focusing on displaying clustered privacy-preserving histogram data. Again the approach clusters social network data based on temporal domain attributes (summary histograms), neglecting the temporal analysis of network topologies. Recently, Cakmak et al. [62] proposed *dg2pix*, a multiscale pixel-based visualization to highlight temporal states and changes in dynamic networks. Yet, the approach’s usefulness depends heavily on non-transparent graph embeddings, posing the challenge of mapping latent space changes to explicit structural changes. Contrary to *dg2pix* [62], our approach is interpretable and provides an overview of significantly occurring sub-network structures. We thereby ensure that the visible patterns are not merely random in some latent space since we do not use non-linear dimension reduction techniques and only extract interpretable features. Our approach allows comparing multiple dynamic networks and single networks against each other.

5.2.4 Delineation to our Work

We compare a selection of related work to delineate our work and highlight the research gap we intend to close in Table 5.1. The compared dimensions comprise the following aspects: the visualization type, the scalability regarding the number of networks, the visually analyzed sub-network structures, and the sub-group analysis tasks based on the network evolution task taxonomy by Ahn et al. [4].

Table 5.1 reveals common features and outlines the following research gaps: First, the publications colored in blue (see Table 5.1) utilize sub-networks to abstract and visualize static networks. The static network visualizations are used to abstract and increase the readability of node-link diagrams and highlight common motifs. However, all the listed static motif-based network visualizations do not allow visualizing changes within sub-networks in dynamic networks. Second, animations of dynamic networks enable exploring structural properties and individual features. However, animations are not used to display sub-networks in dynamic networks since they tend to increase cognitive load for users, making it difficult to detect and trace structural changes over time [296]. Moreover, multiple dynamic network approaches (orange) provide an overview of evolving dynamic network changes without enabling the exploration of temporal sub-group tasks. Finally, four timeline visualizations by van den Elzen et al. [102], Hadlak et al. [131], Archambault et al. [17], and Xie et al. [322] allow users to explore basic group structures in dynamic networks. However, the four papers either focus on clustering domain-specific attributes or only enable to explore simple motifs or clustered sub-network, such as star motifs in *Massive Sequence Views* [102]. For a more detailed delineation of the last four papers, please refer to the previous Section 5.2.3.

Publication, Year	Visualization			Scalability			Sub-Network			Temporal Sub-Group Tasks				
	Static	Animation	Timeline	Small	Medium	Large	Sub-Networks	Motifs	Graphlets	Structural Properties	Domain Attributes	Individual Features	Shape of Changes	Rate of Changes
Our Approach	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Schreiber et al. [268], 2005	•							•						
von Landesberger et al. [193], 2009	•							•		•		•		
Ma'ayan et al. [212], 2009	•							•	•					
von Landesberger et al. [192], 2010	•							•	•					
Harrigan et al. [137], 2012	•							•	•					
Dunne et al. [94], 2013	•							•	•					
Lenz et al. [199], 2013	•							•	•					
Lekschas et al. [198], 2017	•							•	•					
Kwon et al. [190], 2017	•							•		•				
Chen et al. [74], 2018	•							•	•	•				
Bach et al. [24], 2014		•						•			•		•	
Stein et al. [286], 2010			•	•										
Burch et al. [55], 2011			•	•	•									
Cui et al. [84], 2014			•	•	•									
van den Elzen et al. [103], 2016			•	•	•									
Lee et al. [195], 2019			•	•	•									
Hajji et al. [132], 2018			•	•	•	•								
Cakmak et al. [62], 2020			•	•	•	•								
van den Elzen et al. [102], 2013			•	•				•			•			
Hadlak et al. [131], 2013			•	•	•			•			•		•	•
Archambault et al. [17], 2014			•	•	•	•		•			•		•	•
Xie et al. [322], 2020	•		•	•	•	•	•	•			•		•	•

Tab. 5.1 The comparison delineates our approach from related work. The table is colored by static (blue) and dynamic network (orange) visualizations, with an additional sub-ordering by visualization and sub-network type. The first category **visualization** shows if the work utilizes static network visualizations, animation, or timeline representations based on the taxonomy of Beck et al. [31]. The **scalability** category indicates the temporal scalability with small (< 100), medium (< 1000), and large (> 1000) number of networks in a dynamic network. The **sub-network** category reveals the visually analyzed structures. The last category describes the supported **temporal sub-group tasks** based on the task taxonomy by Ahn et al. [4].

Overall, to this day, dynamic network visualizations did not use motifs and graphlets, although such sub-network structures can provide useful, scalable overviews of evolving sub-network topologies. To the best of our knowledge, our proposed approach is the first dynamic network visualization that provides an overview of evolving sub-network structures in dynamic networks.

5.3 Structure-Based Visual Abstraction

We propose two pixel-based visualizations: a network-level census view presenting an entire dynamic network and a detailed node-level sub-network metric view to investigate the local node neighborhoods of single networks. The views combine

motif-based network analysis with pixel-based visualizations to reveal evolving topological structures in dynamic networks and examine network topologies in detail. Our central idea is to identify significantly occurring motifs and then analyze the motifs using scalable and clutter-free pixel visualizations. Next, we describe the employed dynamic network model and both pixel-based visualizations.

5.3.1 Dynamic Network Model

The input to our approach is a discrete series of directed networks, such as daily snapshots of an evolving social network (see Section 5.6.2). From a practical viewpoint, often dynamic networks are modeled as a sequence of events, such as varying connectivity between nodes. For such cases, temporal discretization can be applied by computing supergraphs to generate static networks [131]. However, identifying a proper temporal discretization for dynamic networks remains challenging since it depends on the application domain, the user task at hand, and the underlying evolving data. For instance, a low temporal discretization results in a large set of static networks with no differences in motif structures. On the other hand, a too coarse temporal discretization leads to large static networks that may hide motif changes. Thus, selecting a potential temporal discretization scale needs to be predefined by the user, considering that identifying a proper temporal analysis scale in dynamic networks is a non-trivial task [88].

5.3.2 Network-Level Census Visualization

Analyzing dynamic networks requires obtaining an overview of the diversity of topological changes in the evolving data, such as identifying changes, trends, states, and outliers [103]. Providing such an overview usually goes beyond solely counting nodes and edges for each time step. There is a great interest in understanding how the underlying topological structures changed. For instance, in social networks studying the formation of triadic closures is of great interest [255].

Our network-level census visualization provides an overview of the evolving structural properties and reveals structural changes, trends, states, and outliers in dynamic networks. Our visual representation enables identifying similar network structures, such as networks that consist over-proportionally of triad motifs (see Section 5.6). Figure 5.1 outlines the three main steps: (1) the computation of a network census (significance profile) for a set of motifs, (2) the visual mapping, and (3) the exploration of the resulting pixel-based visualization. Our idea is to capture significantly

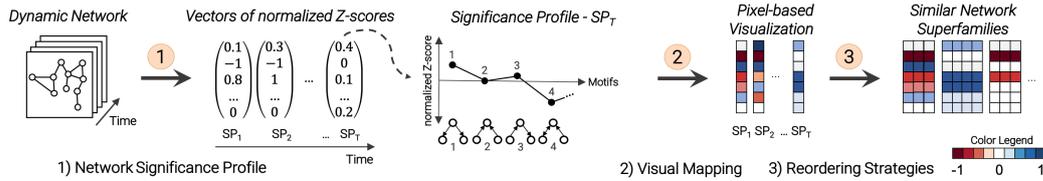


Fig. 5.1 The Figure displays the steps for generating the network-level census visualization. (1) the network motif significance profile (census) is calculated for each time step, (2) the vectors are presented as a pixel-based visualization, and (3) reordering strategies are used to reveal similar network superfamilies. SP_T illustrates the relation of the vector values to their respective motifs. The reordering strategies (see Section 5.4.1) are crucial for grouping similar network topologies to emphasize structural changes, trends, states, and outliers.

occurring network motifs over time and explore the resulting network census as dense overlap and clutter-free representations. The basic idea of the computation of a network census is to reduce and abstract the number of occurring motifs in a single network into a network census vector. The census helps identify similar networks and network superfamilies, enabling the comparison of different sized networks [224]. Network superfamilies are groups (clusters) of similar censuses and thus similar underlying network topology [224]. We also propose to utilize by default the thirteen triad motifs without self-loops, which are basic building blocks of networks [224]. However, the motif selection depends on the application domain and thus needs to be manually adapted based on the user task. For example, users might calculate induced or non-induced quads motifs census [235].

In the second step, the vectors are displayed as a pixel-based visualization to provide an overview of the significantly evolving network structures. The vectors are visualized as pixel bars that encode each value SP_i as a colored rectangle. We utilize a divergent colorblind-safe color scheme from ColorBrewer [138] to emphasize anti-motifs (red) and motifs (blue). The colorblind-safe color scheme utilizes perceptually linear color coding for the ranges between red (under-represented), white (as expected), and blue (over-represented). The used colors are easily distinguishable and have an intense contrast. The network census x-axis displays, by default, the temporal dimension, and the y-axis the motifs of interest. In the third step, we utilize different reordering and aggregation strategies to highlight visual patterns along both dimensions (see Section 5.4).

5.3.3 Node-Level Sub-Network Metric Visualization

A further challenge in dynamic networks is the in-depth comparison of networks and their topological structures at given temporal states. Therefore, we propose

a node-level pixel visualization to compare multiple networks and nodes in more detail, using graphlets instead of motif significance profiles. The graphlet degree vectors (GDV) are node-level sub-network metrics that describe the local network structure around a given node and independently of a given null model [139]. We utilize and display the GDVs of one network as pixel-based visualizations to investigate and compare the structural properties of individual networks in more detail. The main difference from the previous network-level census visualization is that we obtain one visualization for each network as the GDVs are computed for each node. The node-level pixel visualizations are useful for comparing the structural neighborhood of nodes in a network and several networks against each other. For instance, visualizing such graphlets can be used to compare and align topologies of biological networks [187]. The visualization displays on the x-axis the nodes of the selected network and the y-axis displays the individual GDVs. We also want to emphasize that two graphlet-based pixel visualizations of two different-sized networks will also have varying lengths. We use a linear grayscale color scale from ColorBrewer [138] for the graphlet-based visualization to highlight occurrences of local neighborhood graphlets. The color scale highlights frequently occurring graphlets, enabling a simple comparison of GDVs.

5.4 Motif-Based Visual Analysis

In the following, we describe the implemented prototype, which is available at the following online repository <https://github.com/eren-ck/motif-pixel-vis>. The prototype consists of four central components (see Figure 5.2): A *toolbar*, the *network-level census view*, the *node-level metric views*, and the juxtaposed *network view* which displays the underlying network structure as a node-link diagram. As for the *network view*, we compute a supergraph and derive a ForceAtlas2 [156] layout for the whole dynamic network to preserve the user's mental map. Moreover, to increase the network view's scalability, we cluster networks with more than 100 nodes using the Clauset-Newman-Moore algorithm [78]. Thereby, we break down the exploration of large networks into smaller components, focusing on the existing motifs in each cluster. Users can explore all views through zooming and panning using linking and brushing to study the exact pixel values, nodes, and edges attributes via mouseover tooltips.

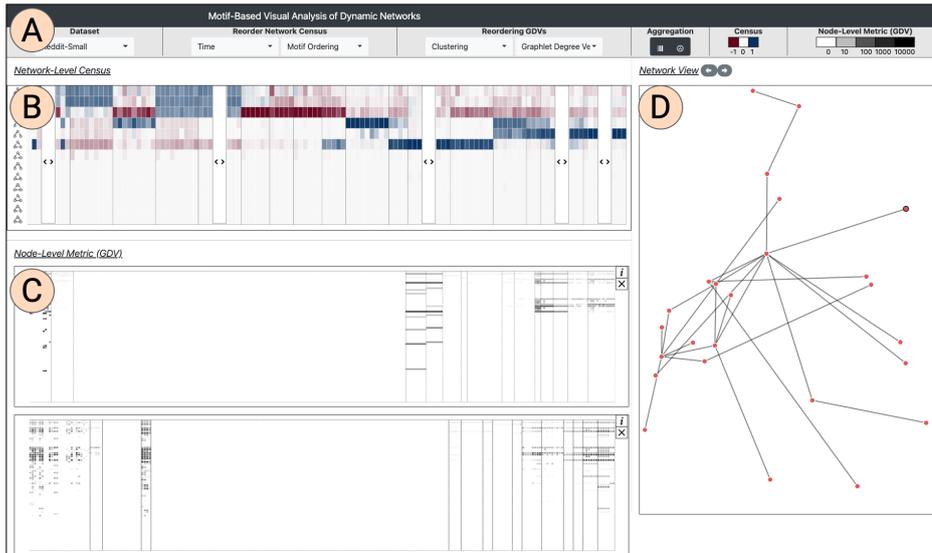


Fig. 5.2 The prototype consists of four central components: (A) the toolbar, (B) the network-level census view, (C) the node-level sub-metric views, and (D) the network view.

5.4.1 Reordering Strategies

The pixel-based visualizations enable users to obtain an overview of dynamic networks through the visual analysis of similar and different pixel bars (see Section 5.5). However, such visual patterns may remain hidden and difficult to detect in pixel visualizations due to the vast amount of visualized data. Therefore, we propose several reordering strategies to reveal similar pixel bars, such as clustering network censuses to uncover similar network superfamilies. We want to emphasize that we cannot suggest an optimal reordering strategy, considering that the reordering depends on the user task at hand, such as identifying the *shape and rate of changes* in group structures as described by Ahn et al. [4]. Each pixel-based visualization can be seen as a $m \times n$ matrix A in which each element $a_{i,j} \in \mathbb{R}$ with $0 < i < m$ and $0 < j < n$. The matrix rows $a_{i,:}$ represent the motif significance profile values over time for the network-level census visualization and the number of graphlets for the node-level sub-network view. The columns of the matrix $a_{:,j}$ encode the motif census or the graphlet degree vectors. Moreover, a matrix reordering is a bijective function $\varphi \rightarrow \mathbb{N}$ that maps the rows or columns with a unique new index position.

We enable users to arrange the columns $a_{:,j}$, hence, the x-axis of the network-level and node-level views using clustering and sorting. The clustering of the network-level census view allows for examining superfamilies of similar network topologies. The clustering utilizes the cosine similarity between the $a_{:,j}$ vectors and HDBSCAN [68] to identify superfamilies of similar dynamic network structures. We

use by default the widely applied cosine similarity, however, other distance functions like Euclidean or Earth Mover distance can interchangeably be used. Moreover, we utilize HDBSCAN [68], as this approach implements a heuristic over different parameters to discover clusters with differing densities. The vectors $a_{:,j}$ in each identified cluster are ordered by the temporal dimension. We also allow reordering of the network-level census columns $a_{:,j}$ by sorting the networks using evolving graph metrics, such as the number of edges or the average clustering coefficient of each network N_i . The reordering using such metrics enables us to relate global network metrics with the evolving structural properties over time. The columns $a_{:,j}$ of node-level metric views can be analogously reordered by clustering and sorted by node metrics, such as the page rank or centrality of a node to highlight important nodes. In addition, we enable users to reorder the rows $a_{i,:}$ by computing statistical measures, such as the mean, minimum, maximum, variance, and standard deviation of the SP_i over time and GDV_i values. The reordering of the rows lets us rank dimensions according to statistical measurements to highlight patterns, such as block and band patterns as described by Behrisch et al. [34].

The reordering strategies help investigate changes in the underlying evolving sub-networks and provide an overview of a dynamic network. However, pixel-based visualizations are hard to understand due to the cognitive effort to derive patterns from several thousand or more pixels [170]. Thus, we also propose aggregation methods to abstract visual patterns and reduce cognitive efforts for users.

5.4.2 Aggregation Strategies

We cluster the vectors and allow users to expand and collapse specific clusters to analyze them in detail if the x-axis does not scale with the number of time points or nodes. We utilize the HDBSCAN [68] to cluster the vectors, including also the temporal aspect for the network-level census view. We do not cluster the y-axis since we expect both visualizations to scale up to 1000 motifs. We include the temporal aspect into the clustering process for the network-level census view by adjusting the similarity metric using a temporal filtering threshold ε_{time} . We compute the distance matrix between all network censuses using the cosine similarity. Afterward, we filter the distance matrix using an epsilon ε_{time} for the temporal dimension to cluster only temporally close time steps. For example, for dynamic networks with a daily temporal granularity, $\varepsilon_{time} = 7$ filters and detects clusters of network censuses that lie within a week interval. The ε_{time} value can be set in the interface and is per default set to ten to consider only temporary close networks.

For the second node-level sub-network metric visualizations, we are utilizing the standard HDBSCAN [68] algorithm using the cosine similarity between the graphlet degree vectors. In both pixel-based visualizations, an exploration of the found clusters is possible by expanding or collapsing them. We display the collapsed cluster visualization as abstracted versions of each cluster. The abstracted version depicts the first three and the last three vectors, plus an unfold button which indicates hidden vectors (see Figure 5.2-(B)). Overall, the aggregation increases the visual scalability of both pixel visualizations. Naturally, users can combine reordering and aggregation strategies to emphasize and reveal visual patterns in each cluster.

5.5 Visual Patterns

We want to describe the potential visual patterns for one motif in the network-level census visualization, meaning row-based changes in a series of pixels (see Figure 5.3-(A)). There are five fundamental low-level patterns in a series of colored pixels: the value *changes*, remains *constant*, *increases* or *decreases* slowly, and *alternating* colored pixels. The pattern interpretation depends on the represented motifs and the underlying dynamic network. However, we can interpret color changes as overall shifts in the underlying networks. For instance, a white to blue pixel color change reveals that the network topology changed, meaning that the motif now appears significantly more often than expected in a random network. Moreover, we expect domain-specific motifs to occur more often, such as constant anti-motifs in some social networks. For instance, Figure 5.5-Facebook consists of chain response anti-motifs (3-triad). In addition, we expect some visual patterns to be rare in real dynamic networks since real-world data usually does not radically change within a single time step. For example, changes from anti-motifs (red) to motifs (blue) between two consecutive pixels. If such rare patterns occur, we support examining them in more detail to understand why they appear, utilizing our network view.

We want to present high-level visual patterns based on the described changes between pixel bars (see Figure 5.3-(B)). Changes in such pixel bars can occur for single or multiple pixel values between two pixel bars. In the network-level view, blocks of similar pixel bars are *temporal states*, and the underlying networks are groups of network superfamilies. *Changes* between such temporal states are visible distinct block patterns. Constant changes of pixel bars indicate a temporary *trend* and slowly evolving network topologies. Finally, *outlier* networks in the dynamic network are visible distinct pixel bars enclosed by similar pixel bars. In the node-level sub-network metric visualization, each pixel bar encodes the actual occurrences of

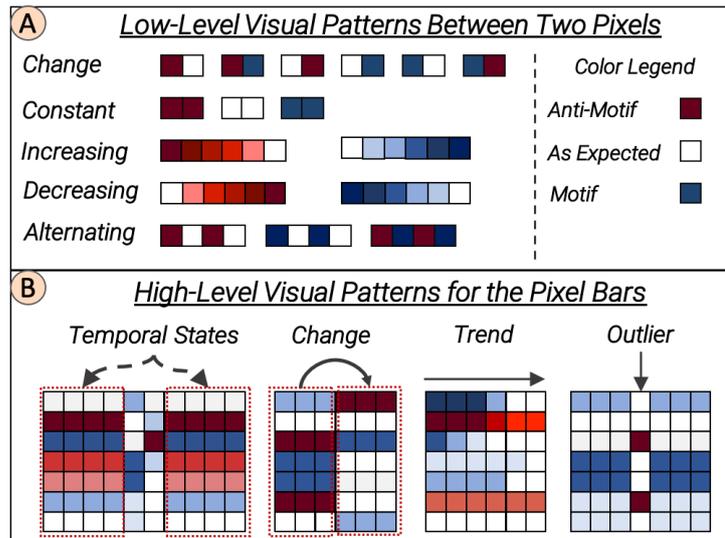


Fig. 5.3 The visual patterns and their interpretations are described in Section 5.5. (A) highlights the changes between pixels in a row, and (B) displays the potential visual patterns for pixel bars.

motifs in the node’s topological neighborhood. The view allows to break down large network structures and compare multiple networks by displaying the graphlet degree vector (GDV) of a node as a pixel bar (column). The GDV vector interpretation is relatively straightforward. Similar local sub-network structures have similar pixel bars and vice versa. Hence, discovering similar motif structures in large networks requires only the pairwise comparison of similar or dissimilar GDVs.

In both pixel visualizations, users have to identify similar and dissimilar pixel bars to detect relevant visual patterns. Overall, discovering similar pixel bars is relatively simple due to the Gestalt principles of continuity, similarity, proximity, and closure [313, Chapter 3]. The similarity and proximity principles in combination let us perceive a sequence of similar pixel bars as a block. Such similar blocks are essentially temporal states which are a sequence of similar network superfamilies in the evolving network. Moreover, the closure principle lets us perceive reoccurring blocks of similar pixel bars as repeating temporal states. The visual analysis of the pairwise similarity between neighboring pixel bars enables users to detect temporal changes and trends in the dynamic network. If the pixel bars change abruptly, this indicates that the underlying structure in the dynamic network has changed drastically. Likewise, based on the continuity principle, constant changes in the pixel bars indicate a trend of shifting network structures. Discovering an outlier pixel bar in a block of similar pixel bars or the whole pixel-based visualization can be seen as a local or global outlier network structure.

The described visual patterns can be matched to existing dynamic network task taxonomies. Next, we want to briefly highlight the supported tasks based on the taxonomy for network evolution analysis by Ahn et al. [4]. The low-level visual patterns in the network-level census view support the tasks: the shape and rate of changes of *growth & contraction*, *convergence & divergence*, *stability*, *repetition*, plus the *fast & slow*, and *accelerate & decelerate* for individual structural groups. For instance, increasing and decreasing series of pixels can be used to analyze *growth & contraction* of motifs. The low-level visual patterns enable identifying the listed shape and rate of changes. Moreover, the high-level patterns enable analyzing the shape and rate of changes of multiple motifs simultaneously. Moreover, the node-level metric view supports some *individual temporal feature* tasks: *examining* and *comparing* structural metrics using graphlet degree vectors between two time points. For example, we can use node-level metric views to compare the number of triads or star motifs in a network and also between networks. We also want to highlight the *individual temporal feature* tasks that are not supported. The proposed visualizations do not allow examining or tracking entities over time, such as the *birth and death* of single motifs. The two pixel-visualizations are unsuitable for identifying a single motif's appearance or disappearance. However, the network-level census view allows the identification of such motifs if they occur significantly more often than expected in a random network. Likewise, our pixel-based visualizations do not support the temporal analysis of *domain attributes* of nodes, links, or motifs.

5.6 Evaluation

Next, we showcase the applicability of our approach in two extensive use cases, analyzing synthetic and real-world dynamic networks.

5.6.1 Synthetic Dynamic Network

The following use case highlights the scalability of our approach using a synthetic dynamic network with a generated ground truth. The synthetic dataset is used to showcase how known network superfamilies can be identified and how individual networks within the dynamic network can be compared.

Dataset Generation We generated 600 synthetic directed networks using five commonly used graph generators to derive a dynamic network consisting of 150 nodes and ≈ 220.000 edges over time. Each synthetic network is made up of 20 to 150

nodes. In terms of time, we randomly arranged the generated networks to obstruct the visual identification of similar networks. Moreover, we computed the network censuses using the default 13-triads (see Figure 5.4). As a null model, we used the configuration model [232, Chapter 4]. We utilized the following graph generators and parameters of the networkX [223] Python package to generate the 600 networks. We explored different parameters for all the used graph generators to generate networks with distinct motif structures. First, we created 100 Erdős-Rényi [107] networks with an edge probability of 0.1. We expect these networks to have slightly over-represented motifs, in particular, often-times all 1 – 7-triad motifs. Second, we created 100 networks using a graph generator with 10 – 200 edges using the selection sampling technique by Knuth [180]. Hereby, we obtain networks with randomly occurring 2,4,5,6-triad motifs. Third, we created 100 rings of cliques and 100 connected caveman networks [316], consisting of up to ten cliques composed of complete graphs ranging from three to six nodes. We expect the 8 and 13-triad motifs to be over-represented in the resulting 200 networks. Finally, we generated 100 networks each, using a growing network with redirection (GNR) [184] using two different redirection probabilities of 0.7 and 0.8. For a redirection probability of 0.7 and 0.8, the 2 and 3-triad motifs occur. The 0.7 probability leads to slightly different networks, with the 4-triad often appearing as an anti-motif.

Problem Background The described artificial dataset contains various synthetic networks with similar topologies and sub-networks. The first task in our use case is to explore changes in motif structures over time in the network-level census visualization, including *stability*, *growth & contraction*, *convergence & divergence* as described by Ahn et al. [4]. However, providing an overview of such evolving sub-networks in dynamic networks is a non-trivial task. A fully computational approach to calculate the similarities between such networks is not feasible. For instance, the graph editing distance [51] does not scale to networks with more than 16 nodes [41]. Therefore, heuristics are often used, such as the computation of network censuses which help to identify similar networks [224]. However, such heuristics are imprecise and require additional visualizations to analyze and validate potential network superfamilies. To the best of our knowledge, no visualization approach allows users to visually analyze similar sub-network structures in dynamic networks. For example, to identify network superfamilies and compare sub-networks of individual networks. The second task is to examine and compare individual temporal features and structural metrics using the node-level sub-network view. The comparison enables us to analyze and compare network topologies of single networks, which is rather challenging and was never implemented using pixel visualizations to the best of our knowledge.

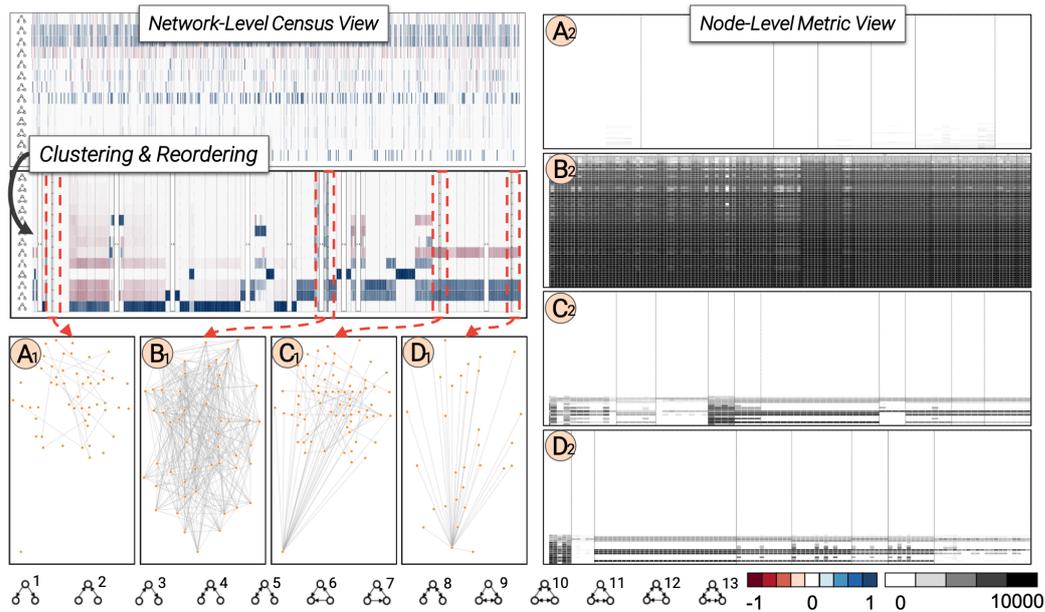


Fig. 5.4 The use case outlines the analysis of a synthetic dynamic network with ground truth (see Section 5.6.1). First, the top-left pixel visualization displays the entire 600 synthetic networks as a network-level census visualization. The second row shows the same data after applying clustering and reordering strategies to reveal visual patterns. (A-D) displays some distinct pixel bars and their underlying networks as node-link diagrams (left) and node-level metric views (right). The synthetic dataset reveals how both pixel visualizations can be used to expose temporal states and compare networks using the complementary node-level metric view.

Network-Level Census View Figure 5.4 (top-left) shows the resulting pixel visualization of the generated synthetic dynamic network. Some visual patterns are already visible in the network-level census visualization, such as temporal states in the form of reoccurring pixel bars over time. We apply aggregation and reordering strategies to emphasize and reveal the encoded visual patterns. First, we cluster and abstract similar network censuses (columns) on the x-axis to group and abstract the temporal states of similar network superfamilies. Moreover, we reorder the y-axis (rows) using the variance of each motif, arranging all motifs with a high variance at the bottom of the y-axis in the network-level view. The reordering of the y-axis arranges significance values with a high variance together, allowing one to compare motifs quickly. More specifically, all motifs commonly occurring within the dynamic network are placed together, and vice versa, discriminatory motifs between the networks are placed close to each other. The aggregation and reordering step is depicted in Figure 5.4 labeled as clustering and reordering. The resulting visible blocks of similar pixel bars reveal similar networks encoded in our synthetic ground truth. Moreover, some clusters contain similar pixel bars and are still not assigned to the same cluster. This can be explained by the fact that we include the temporal di-

mension in the clustering process. The clustering further increases the visualization's scalability since the number of pixel bars has been significantly reduced from 600 to approximately 100 pixel bars. The cluster abstractions also enable us to analyze and compare pixel bars in more detail by unfolding and displaying all pixel bars of each respective cluster. For example, to analyze and identify local pixel bar outliers in each temporal state. In the following, we analyze some clusters by displaying individual networks of each cluster in the network view. Figure 5.4-($A_1 - D_1$) displays some distinct pixel bars and their respective underlying networks as node-link diagrams. (A_1) presents a white pixel bar that reveals a randomly sparse network which was generated with a random graph generator using the sampling technique as described by Knuth [180]. The label (B_1) depicts a more densely connected network which is recognizable as an almost continuous blue pixel bar displaying the network census for a network that was created with Erdős-Rényi [107] graph generator. The following labels ($C_1 - D_1$) highlight similar pixels bars clustered differently due to the temporal distance between the networks. The underlying networks also look more similar to each other than the previous labels ($A_1 - B_1$). Therefore, we add complementary node-level metric views to enable the comparison of motifs in the selected networks of ($A_2 - D_2$).

Node-Level Metric View Figure 5.4 also displays on the right side the four networks of ($A_2 - D_2$) as node-level sub-network metric views. We use clustering and reordering strategies to cluster the graphlet degree vectors (pixel bars) on the x-axis and reorder the y-axis according to the variance of each graphlet feature. We can easily distinguish the sparse and densely connected in (A_2) and (B_2) node-level metric views. In addition, one can also see differences between (C_2) and (D_2), which were both generated with GNRs [184] and have similar pixel bars in the network level census visualization. You can see that the networks have a similar topology, but they are slightly different, which is also visible to some extent in the network view. For instance, in (C_2), more nodes are connected than in (D_2), with fewer nodes resembling a star network. The clustering and ordering of the node-level metric view are computed for each network separately; hence, we cannot directly compare the individual nodes on the x-axis with another node-level metric view. Therefore, we have implemented a linking and brushing that highlights the same nodes in each node-level metric view using a mouseover. In summary, Figure 5.4-($A_1 - D_1$) depicts four distinct networks with a varying number of nodes, including two similar networks ($C_1 - D_1$), which are generated with the same graph generator and contain visible differences in the node-level metric view.

5.6.2 Real-World Data

Next, we analyze real-world dynamic networks to reveal and interpret structural changes, trends, states, and outliers. Moreover, we provide an overview of the structural changes and compare the evolving structural properties of three real-world dynamic networks.

Datasets Figure 5.5 displays the three directed dynamic networks as network-level census views. The presented real-world dynamic networks are publicly available in the Stanford Network Analysis Project [200]. The datasets were pre-aggregated to a daily temporary granularity. Therefore, in the following, every pixel bar corresponds to one day in one of the following datasets. The *Facebook* [308] displays wall posts between users in the City of New Orleans with 1560 days, 45.8K nodes, and 856K edges. The *Bitcoin OTC* [189] presents a who-trust-who network on the Bitcoin OTC platform with 1763 days, 5K nodes, and 35K edges. The *Reddit* [188] encodes hyperlinks (edges) in a social network between subreddits (nodes) with 1217 days, 55K nodes, and 858K edges. We want to also briefly describe the network evolution tasks [4] for the real-world datasets. Similarly, the tasks are the exploration of *shape and rate of changes* using the network-level census view and the comparison of *individual temporal features* using the node-level sub-network metric view.

Dynamic Network Exploration The initial striking observations are visible differences and changes in over and under-represented motifs within and across the displayed evolving networks. The 3-triad motif representing a chain response is under-represented in all three views, visible as a low-level constant anti-motif pattern. In particular, the 3-triad is prominently visible as an anti-motif in the Facebook dataset, meaning that chain responses on multiple Facebook walls are underrepresented. In this context, the first label Figure 5.5-(A) highlights a significant structural change where 4 and 5-triads started to be more frequently over-represented in May 2006, being visible as an increasing low-level visual pattern. The 4 and 5-triads reflect the mutual posting and replying behavior between friends in the Facebook network. The appearance of these motif triads in 2006 correlates to the growing number of Facebook users, which doubled worldwide in 2006, leading to an over-representation of the 4 and 5-triads. The visible change in Figure 5.5-(A) is a direct result of the fact that more users joined and started to use the wall feature, creating more communication in the social network and thus motifs.

The label Figure 5.5-(B) highlights two trends in which the 8-triad is strongly over-represented on the Bitcoin OTC platform. The trend is visible as an increasing and afterward decreasing low-level visual pattern for the 8-triad. The 8-triad motif

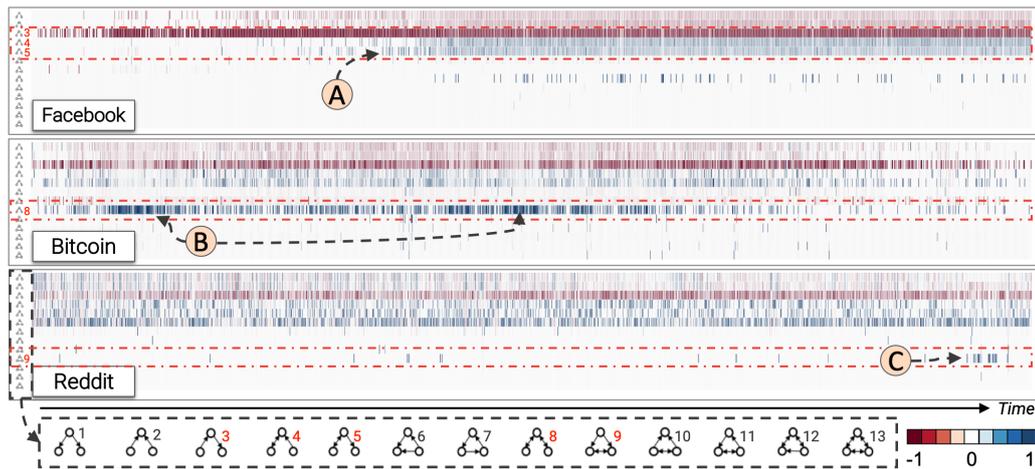


Fig. 5.5 The three network-level census views show distinct real-world dynamic networks (see Section 5.6.2). (A) highlights a significant structural change between two temporal states in a social network. The 4 and 5-triads reflect the mutual posting and replying behavior between friends in the evolving social network. (B) emphasizes two temporal trends with 8-triads representing the behavior that two users give each other trust ratings after transactions. (C) indicates an outlier period in which the 9-triad is over-represented due to NFL Superbowl.

represents the behavior in which two users on the platform frequently give each other trust ratings after transactions. Furthermore, the two highlighted periods can be linked to real-world events. The first period is between May to June 2011, during which the Bitcoin price rose from \$1 to \$30, and the second period is from March to May 2013, in which the Bitcoin price briefly increased to \$250. During these two periods, there was probably increased trading activity, and as a result, users issued more trust ratings.

Finally, Figure 5.5-(C) outlines an outlier period in the Reddit network between February to March 2017 in which the 9-triad is over-represented. The visual pattern is a change, including some alternations between motif (blue) and the occurrences as expected (white). We can investigate the visible outlier period by selecting the various networks with the 9-triad motif and displaying the underlying network structure as node-level sub-network metric views. The detailed views reveal numerous hyperlinks between subreddits dealing with the National Football League (NFL) in the USA. The triads appear quite prominent following the NFL Superbowl in February 2017. These structural changes can be linked to real-world events. Between February and March 2017, there were general discussions about potential NFL player trades and draft picks, various debates about NFL players protesting during the national anthem, and a discussion about a proposed “bathroom bill”.

Social Network Analysis Next, we show how the reordering strategies and the node-level sub-network metric view help to investigate and compare the structural properties of networks in more detail. We continue to visually explore the Reddit hyperlink network [188] in more detail. First, we reorder both axes of the network census view using clustering for the x-axis and then sort the rows of the y-axis using the median of each SP_i value. As a result, we obtain a network census view presenting clusters of network superfamilies, including a noise group. The reordering of the y-axis also highlights block and band patterns in the pixel visualization as described by Behrisch et al. [34]. Next, we analyze the view in more detail by zooming into specific parts of the network census view. The first view in Figure 5.6 reveals distinct pixel bars in the “noise” group and superfamilies of similar networks in the identified “clusters”. The clusters are grouped temporal states previously described as high-level visual patterns. In each cluster, the grouped pixel bars are ordered according to time. Furthermore, we can easily detect minor differences within the network superfamilies (clusters), which can be investigated and compared in more detail in the node-level sub-network metric views. For instance, we can investigate subtle trends or outlier pixel bars within each cluster, visible as a high-level visual pattern.

Node-Level Metric View Next, we provide an overview of the structural properties of single networks. We select outstanding networks from the “noise” group (see Figure 5.6-(A)) and one prominent census from one of the clusters (see Figure 5.6-(B)). The prototype then displays the two selected networks as node-level sub-network views, which we then sort the x-axis according to node page rank in ascending order. Afterward, we investigate the two node-level metric views and interpret the apparent groups to expose structural differences between the two networks. The Figure 5.6-(A) consists of individual nodes that are strongly connected and many individual nodes that have only one link to another node, forming the white space in the middle of the pixel visualization. A quick exploration of the nodes via mouse-over reveals that the left group consists of NFL teams subreddits (e.g., *49ers* and *ravens*). The right group consists of more general NFL subreddits (e.g., *nfl* or *nfl_draft*) discussing and linking potential trades and draft announcements. The visible similar pixel bar blocks appear after using the reordering strategy. We want to highlight that the NFL teams and draft groups have distinct pixel bars, meaning the nodes have different topological neighborhoods. The NFL teams group consists of NFL teams that are all linked by the *nfl* subreddit, which can be a typical bot activity to advocate trending topics in the subreddits. Moreover, after zooming in and analyzing the pixel bars, we can see subtle differences between the NFL team nodes, which we can then investigate in the network view. The subtle differences are

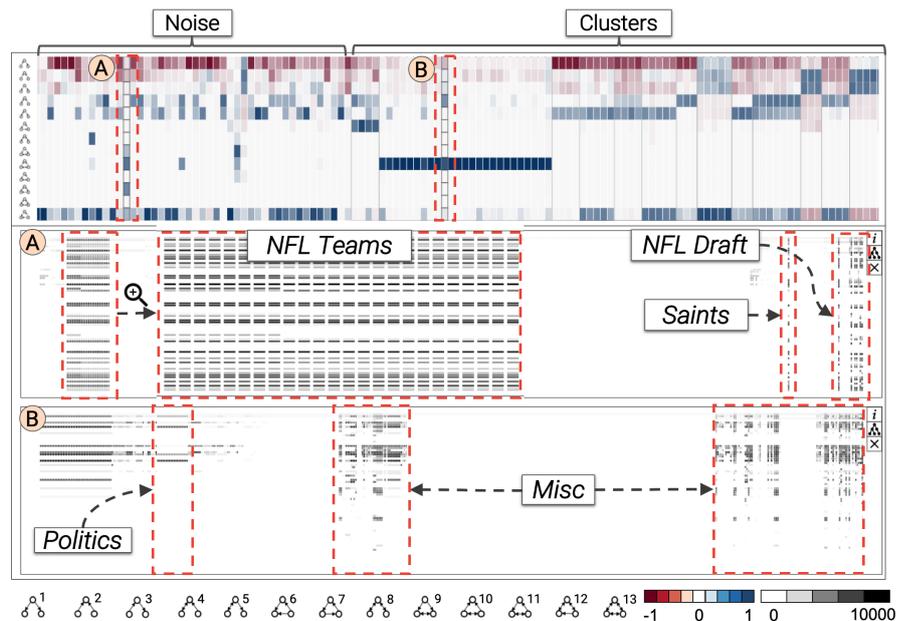


Fig. 5.6 The first row displays a portion of the Reddit data [188] as a network census view, and below that are two node-level metric views that depict two selected single networks (A-B). (A) highlight a distinct pixel bar in the “noise” cluster containing varying network censuses. (B) shows an outstanding pixel bar in a cluster of similar network superfamilies. The node-level metric view of (A) shows NFL subreddits being linked to trades and draft announcements. (B) reveals political and misc subreddits linking each other. Section 5.6.2 details the social network analysis.

visible as a high-level pattern of similar pixel bars and originate from the fact that some NFL team subreddits are also linking each other, which seems to be the typical linkage behavior of Reddit users. In contrast, the NFL draft group consists mainly of central nodes linking to all the NFL teams. Moreover, there is one outlier pixel bar, the NFL team subreddit, the *saints* linking to more than ten teams, indicating some bot activity again. The second network Figure 5.6-(B), consists of more linked sub-network structures, including more nodes with various hyperlinks between them. For example, a group with political topics that link one subreddit with various nation subreddits (e.g., *Sweden* and *Greece*) is visible. To the left of the political topics group, there are again numerous nodes that are linked to only one node, indicating some bot activity. Further, there are larger sub-networks that consist of miscellaneous topics (misc), such as computer games or education subreddits. Again, there are sub-groups with similar pixel bars in each of the described groups, which are either disconnected or connected small sub-networks.

Finally, we want to compare the two node-level sub-network views, revealing that both networks have different network topologies. In Figure 5.6-(A), there are many more isolated linked nodes (white space) and fewer mixed subreddits groups. The

network topology seems to be an exception in the evolving Reddit dataset. In comparison, the network in Figure 5.6-B occurs more often in the dynamic network since we have a cluster of similar networks in the network census view. The use case illustrates how we can use pixel visualizations and reordering strategies to discover similar networks and compare specific sub-network topologies.

5.7 Discussion

Our use cases demonstrate the applicability of our approach for identifying and comparing changes, states, trends, and outliers, including superfamilies of similar sub-networks structures in large-scale dynamic networks. Still, our approach has some limitations.

Motifs We want to discuss the input parameters of our approach. The first step in Figure 5.1 depends on the selected motifs and the used null model to compute the network census (significance profile) for each time step. Hence, these two parameters have to be set by a user as they heavily depend on the analyzed data properties, the application domain, and the user's task. For instance, the null model depends on the application domain as it has to generate networks with similar topological properties to the real-world networks (e.g., similar network density). We plan to investigate useful motifs and null model combinations in future work, including filtering motifs with particular node and edge attributes. Moreover, we propose using triad motifs as default since triads are considered the lowest level of social structures [148], and they are used to reveal network superfamilies [224]. Alternatively, users can select smaller or larger motifs, such as dyads or quads. However, discovering large motifs is computationally expensive since the runtime of motif discovery algorithms depends on the motif and network size. Computing motif censuses for large dynamic networks are only feasible up to eight node motifs since the runtime increases dramatically starting from eight node motifs, as shown in the runtime comparison of Masoudi-Nejad et al. [217]. The choice of motifs directly influences the visual patterns and, hence, the analysis and perception of changes in the dynamic network. We consider the usage of different motifs, including motif sizes, as an advantage of our approach and a chance for further future work to examine how motif sizes affect the perceived visual patterns.

Usability Pixel visualizations remain challenging to read due to the sheer amount of displayed data. We tried to address this limitation by providing various reordering strategies to highlight similar rows and columns. In addition, the interpretation of

network census depends on the application field. For instance, motifs have different meanings in biology or social network analysis. We also identified the risk that the white color pixels ($SP_i \approx 0$) might be misunderstood. The white pixels do not necessarily imply that the motifs are not occurring in the network but rather that they are not occurring significantly more or less often than expected in a null model. Moreover, the visible patterns depend on the used color scheme. For instance, the decreasing and increasing low-level visual patterns are challenging to see if the color map nuances resemble each other too much. We plan to resolve this limitation by interactively adapting the color scheme to user input to highlight particular visual patterns. We want to point out that two equivalent motif censuses do not imply that the networks are identical. For a similar motif census, one can only conclude that the networks have a similar underlying network structure. Still, we cannot infer whether the network nodes or edges are identical. For this purpose, we propose to utilize the node-level sub-network metric view to compare multiple networks to explore similar sub-networks and nodes. Furthermore, the approach remains challenging for untrained users unfamiliar with network science due to the challenging interpretation of the network census, graphlet degree vectors, and the variety of proposed reordering strategies. We also want to examine reordering strategies for different user tasks, for example, to identify helpful reordering strategies to emphasize the shape and rate of changes in network censuses. In future work, we plan to evaluate the approach's usability with users and develop suitable user guidance methods to analyze the pixel visualizations semi-automatically. Moreover, the node-level sub-network metric views are independently reordered, so the direct comparison of columns is currently only possible using linking and brushing. In future versions, we want to improve the comparison of multiple pixel visualizations through discrepancy visualizations that highlight minor differences, including exploring new reordering strategies that better align multiple node-level sub-network metric views.

Scalability The scalability of the approach poses another challenge since the computational time grows exponentially with the size of the motifs as the subgraph matching problem is known to be NP-complete [80]. Overall, computing network census or applying the orbit counting algorithm is only feasible for sub-network structures between three to eight nodes [146]. For more details regarding the execution times of various subgraph counting algorithms, please refer to the recent survey of Ribeiro et al. [251]. In addition, the generation of null model networks can also be computationally expensive. We are currently generating by default 100 null models for each network census. Therefore, we suggest precomputing the network census for each step and loading the significance profiles into the main memory. Apart from that, the last computationally expensive aspect is the clustering of potentially

large static networks for the visualizations in the network view. For larger networks with more than 1000 nodes, the aggregation and visualization method has to be adapted. The survey of von Landesberger et al. [309] lists and discusses useful methods to simplify and display large networks. In addition, the visual scalability for the network census view and the node-level metric view depends heavily on the available display space. We increase the scalability along the x-axis by aggregating and simplifying the clusters of similar pixel bars. However, if the clustering outputs too many clusters, this can lead to the visualization no longer scaling with respect to the x-axis. In future work, we aim to overcome this limitation by providing an interactive multiscale clustering for the temporal dimension and also by utilizing frequent pattern analysis. We assume that this will enable users to change the temporal and network granularity to reduce the complexity of pixel visualizations. The y-axis scales up adequately for each pixel visualization up to 1000 motifs. Although, we expect displaying 1000 motifs is not feasible in most applications since the computation is quite expensive for an entire dynamic network.

5.8 Conclusion

In this chapter, we presented a visualization approach to provide a scalable overview of structural changes in long and large-scale dynamic networks. The approach utilizes motif significance profiles and graphlet degree vectors to capture and display the structural similarities between evolving network structures as pixel-based visualizations. The pixel-based visualizations reveal similar temporal states, trends, and outliers in dynamic networks using motifs and node-level statistics. Moreover, the approach allows exploring abstracted dynamic network summaries searching for temporal patterns (e.g., network superfamilies) without previous knowledge about the evolving data. Overall, the proposed visualizations enable us to display static and dynamic networks to provide an overview of the underlying evolving structural properties. The main idea of our approach is not limited only to motifs or graphlets and can be generalized to display other structure-based properties (e.g., evolving roles) of dynamic networks. In future work, we plan to integrate new methods for tracing and comparing temporal motif structures and motif sub-networks, including the same set of nodes and edges over time. Moreover, we plan to study network metrics and their effects on the overall changes in the evolving network to semi-automatically identify essential motifs in the network census which potentially reveal the before-mentioned network metric changes.

Multiscale Visual Analysis of Temporal Summaries in Dynamic Networks

Summary

Providing an overview of changes in a dynamic network remains challenging due to the underlying large-scale evolving data. Visual analytics approaches, therefore, often utilize abstraction methods to reduce the complexity, for instance, by applying temporal aggregation. However, previous approaches usually abstract the dynamic processes at only one temporal abstraction scale. We present in this chapter Multiscale Snapshots, a visual analytics approach to analyze temporal summaries of dynamic graphs at multiple temporal scales. Multiscale Snapshots combines a hierarchical temporal model with unsupervised graph learning methods to semi-automatically analyze temporal states, trends, and outliers in a dynamic network. Multiscale Snapshots enable users to discover similar temporal summaries and explore structural and temporal properties of a dynamic network. We demonstrate the approach's usefulness through a quantitative evaluation and the application to a real-world dataset. Multiscale Snapshots enables retracing and comparing dynamic patterns and changing network properties in large-scale dynamic networks at multiple levels of temporal resolution.

The chapter is based on the following publication. Please refer to Section 1.5 for contribution clarifications.

- [64] **Eren Cakmak**, Udo Schlegel, Dominik Jäckle, Daniel Keim, Tobias Schreck. “Multiscale Snapshots: Visual Analysis of Temporal Summaries in Dynamic Graphs“. IEEE Transactions on Visualization and Computer Graphics 27.2 517-527, 2020.

6.1 Introduction

A dynamic graph models changing relationships between entities over time. Many real-world data analysis problems rely on dynamic graphs, including, among others, social, computer, and communication networks, and in practice, contain large amounts of dynamic data, hence presenting challenges for effective exploration. An important task in such dynamic graphs is to obtain an overview of the evolving topology by identifying meaningful temporal intervals and their underlying changing structural properties [54]. For instance, analysts are often interested in identifying stable, reoccurring, transition, and outlier states [103]. However, as dynamic graphs are often large-scale and evolve over long periods, it is a major challenge to identify suitable analysis methods and present the data in a readable, scalable, and expressive manner [31]. Previous approaches for visual analysis of dynamic graph data, therefore, often incorporate temporal abstraction methods (e.g., temporal aggregation and dimensionality reduction) to provide an overview of higher-level structures over time [103]. In real-world applications, the usefulness of such temporal abstraction methods depends on many factors, including the selection of an appropriate temporal scale, the user task at hand, graph size, and frequency of topological changes. Currently, visual analytics systems for dynamic graphs lack methods for the visual analysis of dynamic processes at different temporal abstraction scales (multiscale analysis), often leaving the analyst with the challenging task of distinguishing overlapping temporal changes manually.

We propose *Multiscale Snapshots*, a visual analytics approach to semi-automatically provide a multiscale overview of structural and temporal changes in dynamic graphs. We combine temporal hierarchical abstractions with unsupervised graph learning methods to enable the identification of similar evolving graphs. First, the temporal hierarchical snapshots summarize the dynamic graph recursively at multiple temporal scales to reduce the size of the large-scale data. Second, we apply unsupervised graph learning (e.g., graph2vec [230]) to the snapshots of the hierarchy to learn low-dimensional representations of graph sequences, which enables users to use the embeddings for analytical tasks (e.g., similarity search) and later on to adapt the temporal scale semi-automatically. Third, the visualization of the hierarchy of snapshots provides an overview of trends, allows users to compare periods, and to explore structural as well as temporal properties of dynamic graphs. The approach enables exploring various abstraction methods at multiple temporal scales to provide an overview of large dynamic graphs temporal and structural properties.

With Multiscale Snapshots, we can retrace how dynamic patterns and changing graph properties affect the overall evolving data and compare temporal structures at different levels of temporal resolution. The contributions of this chapter are the following: (1) The Multiscale Snapshots approach to visually analyze temporal and structural similarities at multiple temporal scales; (2) A temporal hierarchical abstraction using unsupervised graph learning methods to reduce the size of dynamic graphs and speed up analytical tasks (e.g., similarity search).

6.2 Related Work & Application Background

Multiscale Snapshots combines temporal summaries with graph embeddings to present an overview of the underlying dynamic phenomena. In the following, we discuss related work from automated analysis, visualization, visual analytics, and multiscale visualization approaches for dynamic graphs.

6.2.1 Dynamic Graph Analysis and Visualization

The visual analysis of long graph sequences has lately gained research attention [31]. The automatic analysis, such as the temporal analysis of static as well as dynamic graph metrics (e.g., centrality, diameter [45], or change centrality [113]), enables us to examine the structural properties of the data. Furthermore, recent approaches in unsupervised learning focus on embedding graph structures into low-dimensional space [330]. However, only analyzing such automatically extracted structural properties might hide specific local dynamic changes (e.g., changes in density) and may fail to capture the overall dynamic phenomena [330]. Interactive visualizations can overcome these challenges by allowing analysts to explore the dynamic relationships in their evolving structural context, and several visualization techniques for dynamic graphs have been proposed. Popular approaches display dynamic graphs as animations [90, 248, 19], timeline [131, 84, 26, 85, 323] and hybrid visualizations [259, 23, 56]. For further reading, we refer to the surveys of Kerracher et al. [174], Beck et al. [31], and Nobre et al. [233].

However, many of the proposed visualization techniques do not scale to a large number of nodes, edges, and time steps at the same time [130]. Consequently, to adapt existing techniques to large-scale dynamic graphs, visual analytics approaches were proposed that combine automatic analysis methods with interactive visualizations to reduce the presented data and highlight structural changes.

6.2.2 Visual Analytics of Dynamic Graphs

The visual analytics of dynamic graphs aims to seamlessly integrate graph analysis methods [231] with visualization techniques [31] to interactively analyze the evolving structural properties. Such visual analytics approaches facilitate abstraction methods for large-scale dynamic graphs to reduce the amount of data and provide an overview of high-level changes. The usefulness of such abstraction methods, however, depends on the applied method (e.g., temporal clustering) and input parameters (e.g., number of clusters) [131]. Therefore, according to Aigner et al. [6], it is essential to interactively adapt abstraction methods and tune their underlying parameters to identify changes that otherwise would remain hidden.

In general, there are two categories of abstraction methods; *data space abstraction* (e.g., sampling, clustering) and *visual space abstraction* (e.g., zooming, focus-and-context) [83]. The *data space abstraction* in dynamic graphs reduces the number of graph elements or time steps [270]. Often, data space abstraction methods lower the resolution of the data (e.g., temporal aggregation [228]). For instance, Van den Elzen et al. [103] proposed a visual analytics approach that segments and aggregates sequences of graphs to a vector and applies dimensionality reduction to obtain an overview of the states in dynamic graphs. However, the resulting overview depends strongly on the selected segmentation scale and the abstraction method (extracted features) into vectors. Further, the dimensionality reduction technique is hard to interpret as the projection does not visualize the evolving graph structure. For an overview of *data space abstraction* methods, we recommend the recent survey of Liu et al. [207]. The *visual space abstraction* methods in dynamic graphs reduce the amount of presented data (e.g., by applying zooming [33]). Many of the *visual space abstraction* methods allow the user to interactively change the depicted level of detail [1, 142, 18, 97, 320, 38, 335]. For example, temporal navigation methods help to interactively adapt the horizontal (e.g., TempoVis [5]) and vertical (e.g., BenderdeMoll and McFarland [36]) time dimension. Multiple visual analytics approaches, including *visual abstraction methods*, were recently proposed. For instance, Small MultiPiles [23] enables users to interactively stack and present a sequence of graphs as piles of adjacency matrices to reduce the number of displayed views. Furthermore, Cubix [25] allows users to visually explore adjacency matrices of dynamic graphs in a cube metaphor. However, in both approaches, identifying temporal patterns in long sequences of adjacency matrix visualizations remains challenging due to limited display space and overlapping issues in 3D visualizations.

Many of the previously proposed approaches focus mainly on aggregation and display the abstracted temporal or structural dimension at one scale, making it challenging

to investigate the influences of abstraction methods on the resulting visualization, as patterns may be found at different scales and intervals. Multiscale visualizations aim to overcome these challenges by simultaneously displaying different levels of abstraction, hence providing an encompassing overview of possible structural and temporal aggregation levels.

6.2.3 Multiscale Dynamic Graph Visualizations

Multiscale (multiresolution) visualizations present the data at multiple user-defined levels of abstraction and are useful for setting detailed abstraction levels into the overall temporal context [99]. For example, Javed and Elmqvist [159] stack different levels of zoomed time-series data in a tree structure to serve as a graphical history and preserve the context during zooming. Nearly all of the previously mentioned approaches visualize the dynamic graphs on a single time granularity (scale) using mostly one adjustable abstraction method. One notable exception is the recent work of Burch and Reinhardt [54] that proposed a timeline visualization technique that allows exploring dynamic graphs at different temporal granularities. However, the approach focuses on bipartite graphs, and due to the overplotting produced by the interleaving method, identifying temporal patterns remains challenging. Most of the listed approaches for dynamic graphs focus on analyzing dynamic graphs at a particular temporal scale and require the manual definition of parameters. For instance, the work of Van den Elzen et al. [103] requires users to define a discretization scale, feature selection, and the choice of a suitable dimensionality reduction technique. In contrast to these approaches, we propose using temporal hierarchical abstractions with unsupervised learning methods to explore input parameters (e.g., discretization scale) and simultaneously visualize graph sequences at different levels of temporal abstraction.

Application Background

The analytical goal of our approach is to provide an overview of evolving graph properties at multiple abstraction scales. The following section describes the addressed problems, the research gap we close, and our derived design goals.

6.2.4 Problem Description

The starting point of data analysis is often an overview visualization to examine the overall data structure and to identify useful analytical and visualization techniques [277]. However, providing an overview of large-scale dynamic graphs can be challenging for multiple reasons [31]. First, the complexity and size of data pose various challenges as many dynamic graph visualization techniques do not scale [30]. Second, it is challenging to visualize dynamic graphs as there is a trade-off between displaying the detailed graph structure for each time step and presenting the evolving properties over time. For instance, animations support the exploration of each static graph over time. However, animations are considered unsuited to provide an overview of long periods due to the problems caused by cognitive effort [296, 140, 19] and difficulties maintaining a mental map in dynamic graphs [248]. Third, creating compact temporal abstractions (summaries) of dynamic graphs is user task-dependent and relies upon the application domain as well as data properties. For example, a fine-grained temporal aggregation in large-scale dynamic graphs results in various intervals with little information and is unable to provide an overview [36]. In contrast, coarse-scale aggregation produces only a few intervals, which may contain a high variance, where meaningful intervals could go unnoticed. Finding appropriate levels of abstraction is a non-trivial task [175].

6.2.5 Gaps in Related Approaches

We compare a selection (see Table 6.1) of recent work on dynamic graph visualization to point out the gap, we intend to close. The selected publications are based on the recursive search of references from the recent surveys of Beck et al. [31] and Nobre et al. [233]. The categories of our comparison comprise visualization techniques from the dynamic graph taxonomy [31], the temporal scalability, including multiscale approaches, and the temporal explorability of different graph structures [233]. The comparison reveals several insights. First, the number of temporal multiscale approaches for dynamic graphs is limited. Multiscale approaches present either time series in a multiresolution design (e.g., graph metrics [131]) or include visualizations having multiscale characteristics (e.g., time curves [26]). Second, timeline visualizations (time-to-space mappings) reduce the size of the data and are suitable for providing an overview of a sequence of graphs. However, such approaches abstract and discretize structural information at one temporal scale (uniform time slicing), often requiring users to manually identify overlapping temporal patterns [311]. For example, timeline-based visualizations often require defining

Publication, Year	Visualization			Scalability			Temporal Explorability					
	Animation	Timeline	Hybrid	Small	Large	Multiscale	Nodes	Edges	Neighbors	Paths	Clusters	Subgraphs
Multiscale Snapshots			•	•	•	•	•	•	•		•	•
Rufiange & McGuffin [259], 2013			•	•			•	•	•	•		
Hadlak et al. [131], 2013		•		•		•					•	•
Cui et al. [84], 2014		•		•	•				•		•	
Bach et al. [23], 2015			•	•							•	•
Burch & Weiskopf [56], 2015			•	•			•	•	•		•	
Bach et al. [26], 2015		•		•	•	•						
Van den Elzen et al. [103], 2016		•		•	•		•	•	•			
Burch et al. [54], 2017		•		•		•		•			•	
Dal et al. [85], 2017		•		•			•		•			
Xu et al. [323], 2018		•		•			•	•			•	•
Wang et al. [311], 2019		•		•			•	•	•	•		

Tab. 6.1 The comparison highlights the essential temporal properties of related visualization techniques, ordered by publication date, and assessed by us to the best of our knowledge from analysis of the works. The **visualization** category classifies the techniques using the taxonomy of Beck et al. [31]. The **scalability** category elaborates on multiscale temporal approaches and the temporal scalability with large scalability meaning a dynamic graph with more than 1000 graphs. The **temporal explorability** is adapted from the work of Nobre et al. [233] and illustrates whether the graph structures (e.g., neighbors, clusters) are explorable and comparable within the temporal dimension.

input parameters, e.g., discretization parameters and derived features [103]. Third, current work either introduces complex visualizations or the combination of various simple views, which poses the question of how to arrange the different visualizations in the limited screen space and link them effectively.

In summary, many visualization techniques view a dynamic graph as a series of static graphs that neglects to capture the evolving structural properties of dynamic graphs simultaneously. In contrast to previous approaches, we interactively apply an unsupervised graph learning method (graph embeddings) on a multiscale temporal hierarchy to directly learn structural properties. We use graph embeddings as a familiar representation for an analytical user task (e.g., similarity search) and utilize the visual exploration of different visual metaphors.

6.2.6 Design Goals

We derived three design goals for our visual design from the previously described research challenges outlined in related work [30, 16, 31].

G1: Time-oriented visual analysis The visual analysis of dynamic graphs lacks new paradigms to examine structural (static) and temporal properties simultaneously. First, identifying structural properties, e.g., clusters in a static graph, allows searching for similar structural properties over time. Such an exploration enables users to identify temporal states and continue to search for similar trends, reoccurring structures, and outlier structures.

G2: Temporal multiscale overview Our core idea is to provide an overview of multiple levels of temporal granularity, which facilitates users to relate higher-level overviews with low-level details. Such a multiscale overview allows detecting useful temporal analysis scales and gives additional context while navigating the temporal dimension (e.g., temporal filtering). For instance, a multiscale overview allows comparing states and transitions across multiple temporal granularities.

G3: Multiple visual metaphors Combining different visual metaphors in a consistent interface allows adjusting the visualization to the data characteristics of particular intervals. It also increases the task coverage by enabling an analyst to adapt the visual representations to the task at hand. For example, matrix-based visualization techniques are better suited for dense dynamic graphs.

6.3 Multiscale Snapshots

Multiscale Snapshots provides an overview of higher-level and fine-grained temporal intervals of large-scale dynamic graphs. The approach reduces the complexity of the data by integrating temporal summarization and graph embeddings in an interactive multiscale visualization.

Our proposed visual analytics approach consists of three adjustable steps (see Figure 6.1) to promote the exploration and summarization tasks for dynamic graphs [46]. The first step transforms the temporal dimension into a hierarchy of snapshots summarizing subsets of the dynamic graph into overlapping multiscale intervals. The second step reduces the complexity of the snapshots by embedding the summaries of the evolving topology into vector representations (e.g., using graph2vec [230]). The mapping of intervals into vector representation allows us

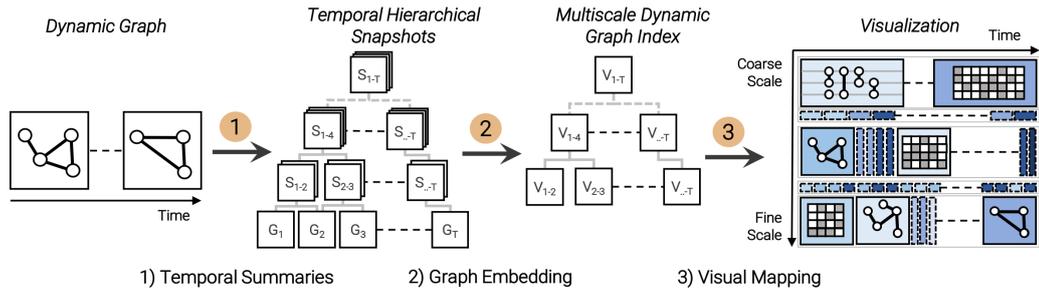


Fig. 6.1 Overview of our Multiscale Snapshots: We (1) recursively create temporal summaries (snapshots) of graphs at different temporal scales (time granularities); (2) We then apply an unsupervised graph learning method (graph embedding) to learn low-dimensional representations of snapshots; Then, we (3) then give an exploratory visualization that organizes the snapshots of different temporal granularities in a hierarchy to provide an overview of the evolving structural properties, which utilizes the graph embeddings for analytical tasks (e.g., similarity search).

to automatically adjust the visualization to highlight temporal states, trends, and outliers. The third step transforms the abstracted temporal data into a flexible and interactive hierarchical visualization and supports essential interaction as well as navigation methods to analyze the evolving graph structure visually. Furthermore, the visualization intends to increase the task coverage by combining different visualizations of dynamic graphs in a consistent interface. The following subsections describe each transformation step in more detail.

6.3.1 Temporal Hierarchical Snapshots

Dynamic graphs model relationships over time (e.g., social networks) and can be described as a number of T static graphs $DG = (G_1, G_2, \dots, G_T)$. The temporal abstraction of dynamic graphs (e.g., aggregation) helps to reduce the data size, speed up temporal queries, support interactive analysis, and eliminate noise [207]. However, the temporal abstraction of sequence graphs into summaries remains challenging due to the selection of *time granularity*, which depends on many factors (e.g., data size) and the choice of *abstraction method* (e.g., summarization).

In various dynamic graph visualizations, a simple selection of one *time granularity* (uniform time-slicing) is used due to the simplicity of the approach [311]. In contrast, we propose a recursive temporal abstraction into a hierarchy with temporal overlaps to model multiple *time granularities* (see Figure 6.2). We generate and stack multiple partitionings using uniform intervals (time slices) of different temporal granularities. We organize the stacked partitionings in a hierarchy that orders the different levels of abstraction (discretizations) from coarse to fine-grained temporal representations.

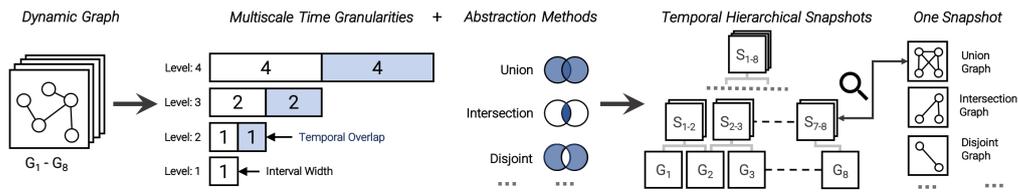


Fig. 6.2 The figure displays the generation of temporal hierarchical snapshots for a dynamic graph with eight timesteps. First, the dynamic graph is partitioned into overlapping intervals at four levels of temporal granularity. The fourth level contains all data, and the first level consists of intervals of size one (static graphs). Second, abstraction methods are applied to the intervals to generate different compact summaries of the subsets of the dynamic graph. The result is a hierarchy of temporal snapshots that contains multiple summary graphs (e.g., union or intersection graph).

Our bottom-up approach groups per default the temporal dimension into intervals of length 2^l with the level $l \in 1, \dots, \lceil \log(T) \rceil$. Figure 6.2 displays an example partitioning for a dynamic graph with eight time steps. Level one of the hierarchy consists of intervals of length one, containing only one graph of the evolving data. The intervals are generated using a rolling window method, which facilitates time discretization without hard boundaries. The rolling window approach for level l is computed by shifting the interval of width 2^l by the temporal overlap of width 2^{l-1} . This results in each level having $\lceil T/2^{l-1} \rceil$ intervals and the whole hierarchy having $(3 \cdot T) - 1$ intervals. Essentially, as seen in Figure 6.2, each generated interval overlaps partly (e.g., per default half) with the next interval except for level one (single graph) and the root node (all graphs). The default recursive partitioning into multiscale intervals results in the height of $\lceil \log(T) \rceil$. In practice, for most datasets, the height of the hierarchy is below 20 (< 1 million graphs). The width of time slicing can be modified to the application domain, for example, intervals with a width of a day, week, month, and year.

The uniform time-slicing produces intervals of the same width for each level. The generation of non-uniform intervals for each level can be computed by applying temporal clustering techniques with varying input parameters. For example, the temporal clustering approach of Hadlack et al. [131] can be used to identify similar substructures based on graph properties to provide an overview of temporal trends. A hierarchy of temporal intervals can also be automatically generated by facilitating unsupervised learning with boundary detectors to obtain hierarchical temporal dependencies at different time scales [77]. The generation of such hierarchical temporal dependencies only works on time-series data of a dynamic graph, for example, on evolving graph metrics such as average clustering coefficient or density. Therefore, applying such methods remains challenging as there is no single graph metric that can capture all of the evolving graph structures.

The *temporal abstraction* summarizes and captures the evolving structural properties of sequences of graphs. We suggest utilizing multiple abstraction methods to generate diverse representations of the generated multiscale intervals as there is not a single abstraction method able to encode all evolving properties of a dynamic graph. We transform the intervals into graph summaries per default using set operations (union, intersection, disjoint graph). For example, the union operation abstracts the interval into a supergraph, which helps to provide an overview of all nodes and edges [130]. The three default graph summarization techniques (see Figure 6.2) are the *union graph* that consists of a union of the set of all node and edge sets. The *intersection graph* which consists of all nodes appearing more than i -times in the interval. The *disjoint graph* consists of all nodes appearing less than i -times in the interval. We set the default value for the parameter i to the interval overlap of an interval. If the values of i are below the interval overlap, this will most probably result in successive intervals with a similar intersection and disjoint graphs.

We call the three computed graph summaries of an interval a snapshot $S_{l,k}$ (see Figure 6.2). A snapshot aims to capture the structural and temporal properties of a sequence of graphs on level l and the k generated interval. The resulting intervals of the snapshots can be indexed in an interval tree to support the efficient support window queries, for example, identifying the best fitting interval to a user-defined period. We suggest, furthermore, utilizing more graph summarization methods based on the analytical task, data characteristics, and application domain. For example, we implemented the Clauset-Newman-Moore community detection algorithm [78] to reduce the overall number of nodes in each static graph and to extract higher-level properties (e.g., meta-nodes and edges). For more graph summarization methods that can be added to our approach, we refer to the survey by Liu et al. [207].

Overall, the first step results in a hierarchy of abstracted snapshots at different temporal granularities (see Figure 6.2). Every interval in the hierarchy contains multiple graph summaries, which can be used for different types of queries later on. For example, we can search for similar changes between intervals by using the disjoint graphs to identify reoccurring changes in the dynamic graph. The resulting temporal hierarchy of the dynamic graphs is used in the next step of our Multiscale Snapshots approach by mapping each summary to its vector representation.

6.3.2 Multiscale Dynamic Graph Index

As for the next step, the resulting hierarchical snapshots are learned and embedded into low-dimensional space to reduce the complexity of the graphs and speed up

analytical tasks (e.g., similarity search). The main goal is to use unsupervised learning methods to model the similarities between the different multiscale temporal summaries and reduce the complex data characteristics to low-dimensional vectors preserving information. We apply a graph embedding (e.g., graph2vec [230]) to map all snapshot graphs (e.g., union and disjoint graphs) to vector representations. In contrast to earlier approaches (e.g., Van den Elzen [103]), unsupervised graph learning methods learn the topological structures of graphs and do not require any hand-engineered features. The embeddings can be precomputed and are also typically small enough to fit into main memory. To the best of our knowledge, Multiscale Snapshots is the first visual analytics approach to propose using unsupervised graph learning methods with different temporal granularities to visually analyze intervals sharing similar properties over time.

Recently, new unsupervised graph learning methods have been proposed to learn node and graph embeddings [330]. However, many of these methods mainly focus on learning static graph embeddings and cannot model the evolving properties of dynamic graphs [330]. In contrast to earlier approaches, we propose to model dynamic graphs by embedding summaries of subsets of the evolving data to capture the temporal dependencies between graphs. The analyst can apply graph embeddings such as graph2vec [230], GL2Vec [73], and FGSD [303] to the snapshots. The approach embeds all snapshots of the temporal hierarchy except for level one (single graphs), which results in the embedding of $2T - 1$ snapshots. The resulting $2T - 1$ embeddings are also indexed to support efficient K-nearest neighbor search queries. We employ the following two index structures: an interval tree to support efficient temporal queries for the intervals, and an individual index structure for each level. We utilize for the indexing of the graph embeddings the proposed method of Malkov et al. [215] to perform a fast K-nearest neighbor search in each level.

In our evaluation (see Section 6.4), we compare different unsupervised graph embeddings, discuss the scalability of the approaches, and show that the embeddings of the snapshots can capture structural as well as temporal changes.

6.3.3 Multiscale Snapshots Visualization

The final step applies a visual mapping to organize the temporal snapshots in a multiscale visualization to enable the visual analysis of the generated snapshots and uses the graph embeddings for analytical tasks. In the following, we describe the components of our visual and interaction design (see Figure 6.3).

The visualization presents the hierarchy of snapshots and orders them from coarse to granular scale (top-down) and facilitates the horizontal (time) as well as vertical (time granularity) temporal navigation to search for similar properties over time (**G1**). The visualization stacks and displays the multiscale temporal abstractions (**G2**), allowing to analyze and compare the abstracted data at different temporal granularities. Presenting multiple abstraction levels enables us to gain more knowledge about the underlying abstracted dynamic graph (e.g., data distribution) [99]. The highest level (root) displays an aggregated version of the whole dynamic graph (e.g., union graph), and the bottom level enables us to depict a limited number of each time step. The levels in-between allow visualizing a subset of the generated snapshots in snapshots views (juxtaposed small multiples).

A snapshot view combines different visual metaphors in a consistent interface to increase the task coverage (**G3**) and displays one of the summary graphs (e.g., union graph). Every view enables users to depict the data using four kinds of visual metaphors (node-link, adjacency matrix, animation, and time series of graph metrics). We use these visual metaphors since the individual benefits and drawbacks of the representations are well studied (e.g., graph layout and matrix reordering) [31]. We utilize multiple visual metaphors for certain intervals as the usefulness of dynamic graph visualization depends on the underlying changing data (e.g., sparse versus dense graphs) [54]. We consider our snapshot views as hybrid visualizations, as the view combines different visual metaphors in small multiple representations. Furthermore, the Clauset-Newman-Moore community detection algorithm [78] is applied to minimize visual clutter and to reduce the number of nodes in a snapshot view, if the size of the displayed summary graph exceeds a specific threshold (more than 100 nodes). This threshold is based on the size classification of Nobre et al. [233]. The resulting communities are then shown as meta-nodes and allow to filter the respective nodes and edges of the community for the entire Multiscale Snapshot visualization. For instance, the filtering of a structural cluster allows us to explore the evolving properties of the cluster in the displayed snapshot views. The snapshot view also visualizes derived structural properties using the background color of each snapshot view to highlight differences between adjacent visual metaphors. The summary graph's derived properties (graph metrics) are used to identify and emphasize temporal or structural graph properties. For instance, we compute graph metrics such as the sum of the number of edges in a snapshot, which indicates the density of the underlying graph sequence. A linear color scale from light blue (low values) to darker blue (high values) is used to highlight changes in the derived structural properties [138].

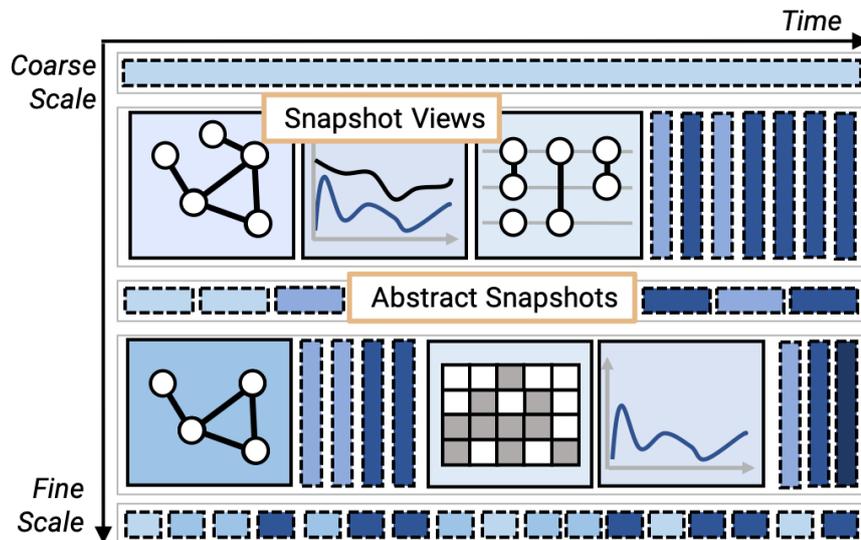


Fig. 6.3 The hierarchy organizes and displays the summaries from the snapshots from coarse to fine-grained representations. The visual metaphors in each snapshot view can be manually or semi-automatically adapted. The snapshot views can be abstracted to reduce the number of displayed views and duplicate information. The background color of each snapshot is mapped to graph metrics (e.g., number of edges).

Using multiple levels of juxtaposed small multiples remains challenging due to limited display space and the preservation of the viewer's mental map. The simultaneous presentation of multiple levels and their snapshot views does not visually scale due to the restricted display space with an increasing number of snapshot views, as the readability of each view decreases. We, therefore, incorporate visual space abstraction methods to limit the number of displayed levels and snapshot views. The number of displayed levels is limited (default four), and during the vertical navigation, the respective lowest or highest level of temporal granularity is removed. Furthermore, we abstract snapshot views to reduce the number of shown visualizations and on particular snapshots while keeping the context of the abstracted views (focus-and-context principle). An abstracted snapshot is displayed as a compact colored rectangle without any visual representation. The background color can be mapped to extracted graph metrics of the selected summary graph, for example, the number of nodes as well as edges, average clustering coefficient, density, and transitivity. The coloring of such abstracted snapshot views enables the identification of intervals with specific properties, such as subsequences of dense graphs. In general, the usage of such color indicating graph properties allows users to identify and compare temporal intervals [291]. The abstraction can be done manually by reducing individual snapshot views or whole levels of the hierarchy, using a user-driven threshold, and an automated abstraction algorithm.

The automated algorithm limits the number of intervals by traversing the hierarchy and abstracting redundant information. The algorithm abstracts snapshots if the number of views exceeds a specific threshold or if the algorithm detects duplicate displayed periods. The algorithm traverses each level of the hierarchy (top-down) and compares the displayed snapshots at each level against each other. If coarse snapshots (high level) are displayed in the fine-grained snapshots (low levels), they are abstracted. The automatic abstraction is done based on overlapping windows in the interval tree, which means that the snapshot view with the highest overlap with low-level snapshots is abstracted. The algorithm compares, for example, the time interval of the root view against all other not abstracted snapshots, and if the periods of these more granular levels display the majority of temporal information of the root view, then the root snapshot view is abstracted. The thresholds for the automatic abstraction algorithm, such as the overall number of levels and snapshot views, are adjustable by the analyst.

Furthermore, we aim to preserve the viewer's mental map, which increases the readability and interpretability of the evolving data [248]. To maintain the viewer's mental map, we fix and use one global layout for each visual metaphor. For instance, we compute one layout for the overall supergraph of the dynamic graph. Furthermore, the usage of linking and brushing aims to preserve the mental map between adjacent snapshots using different visual metaphors and the different levels of abstraction in the hierarchy.

Multiscale Snapshots utilizes the graph embeddings for automated analysis to identify trends, reoccurring, and outlier states. For example, an analyst can select a snapshot view and can apply a k-nearest neighbor search query to detect similar summary graphs (see the query interface Figure 6.4). The detected k-nearest neighbor snapshots can also be disaggregated to more granular views using the interval tree (drill-down). The similarity search can also be applied to a particular type of summary graph, for instance, search for similar intersection graphs. Such similarity queries also enable us to semi-automatically abstract and adapt the displayed snapshot views. The k-nearest neighbor queries can also be applied to particular intervals (subqueries) and to specific levels, which allows examining the summaries of the dynamic graphs in a top-down manner. The embeddings can also be used to cluster levels of the hierarchy and to identify outlier states by applying outlier detection algorithms [2].

In summary, the visual design provides an overview of snapshots of a dynamic graph by combining automatic analysis methods with visual space abstraction methods (focus-and-context).

6.3.4 Multiscale Snapshots Prototype

We showcase the approach’s usefulness by applying it to real-world data using our prototype ¹. The prototype has two components (see Figure 6.4 A-B): the Multiscale Snapshots visualization and the query interface. The components allow users to semi-automatically search for similar temporal states in the dynamic graph.

The **Multiscale Snapshots visualization** consists of a toolbar, the stacked snapshot views, and two context bars. The toolbar facilitates the application of automated analysis methods (e.g., open the query interface) and visualizes the summary graphs of the snapshots (e.g., display union or intersection graph). Moreover, the toolbar enables changing the data space abstraction methods (e.g., filter and cluster nodes) and adapting visual transformations (e.g., reordering algorithms for matrix visualization). The prototype displays the root of the hierarchy as a supergraph (union graph) using a node-link diagram visualization. The layout of the node-link diagram is computed once for the root supergraph using, per default, the Fruchterman-Reingold [114] layout algorithm and later used for all snapshot views. The hierarchy enables an analyst to navigate horizontally (time) or vertically (overview to detail) on the temporal dimension. The two context bars display additional information during the horizontal and vertical navigation of the temporal dimension. The time context bar on the top shows the visualized intervals, and the level context bar on the right allows to add and remove levels. Each snapshot view can be visually analyzed via zooming, panning, brushing, and changing the layout in all views (e.g., matrix reordering) to make visual patterns more apparent [34]. The visual transformations for individual or all snapshot views can be adjusted by the analyst to enable the adaption of visual metaphors to the underlying sequence of graphs, such as switching for periods of dense sequences of graphs to matrix visualization. The prototype also enables filtering by specific graph properties (e.g., node degree) and clustering [78] to reduce overall displayed elements to extract higher-level features (e.g., meta-nodes and edges). The background color of each snapshot view can be mapped to extracted graph metrics and computed node characteristics (e.g., clustering coefficient) to node size. To apply a k-nearest neighbor query, an analyst has to select a specific summary graph in a snapshot view.

The **query interface** allows applying specific k-nearest-neighbor queries to search for similar summary graphs on all or particular levels of temporal granularity. The query interface displays each time dimension of a level and encodes the currently visualized and already investigated snapshots via color. This additional information helps to keep an overview of the already explored snapshots of all levels. The timelines can

¹<https://github.com/eren-ck/MultiscaleSnapshots>

be ordered by different features, such as by the percentage of explored snapshots. An analyst can select the levels, time interval, and the summary graph type (e.g., only union graphs) to apply the k -nearest neighbor search. The number of k -nearest neighbors is also configurable. The query results are displayed as dots on the timeline, and the euclidean distance between the underlying graph embeddings is mapped to the opacity of the dot. The analyst has to select a subset of the nearest-neighbors, which are then displayed in the Multiscale Snapshots visualization. The selected results are shown as snapshot views and allow users to analyze and compare similar temporal states in lower or higher temporal granularities against each other.

6.4 Evaluation

The following section evaluates the two main components of the Multiscale Snapshots approach. We provide a usage scenario to demonstrate how the visual analytics approach can be utilized to gain an overview of temporal summaries in a dynamic graph. We furthermore quantitatively evaluate the similarity (k -nearest neighbors) search of the graph embeddings with synthetic and real-world datasets.

6.4.1 Usage Scenario

We demonstrate the applicability of our approach using a real-world, large-scale dynamic graph of the website Reddit [188]. Reddit is a social news aggregation website with 440 million active users who can publish and upvote posts of interest (e.g., link to news sites) in particular communities (subreddits). The analyzed dataset is a dynamic hyperlink graph in which nodes are subreddits, and edges are hyperlinks posted between subreddits.

Tasks In the following, we outline the actions that a fictitious analyst takes to discover structural and temporal changes during the 2016 US presidential elections (see Figure 6.4). A task in the visual analysis of such hyperlinks is to gain an overview of temporal events (e.g., political scandals), identify reoccurring links between communities, and examine structural changes within groups of subreddits. The visual analysis of such data with state-of-the-art visual analytics approaches remains challenging due to the varying duration of such events. For example, the length of political scandals varies significantly due to media exposure and their temporal context (e.g., during elections). In contrast to previous approaches, Multiscale Snapshots enables us to detect events/states of different temporal lengths due to the temporal multiscale modeling.

Dynamic Graph The Reddit dataset [188] comprises hyperlinks between subreddits from 1st January 2016 to 30th November 2016. The resulting dynamic graph contains 7974 graphs (grouped by hours), 18546 subreddits (nodes), and 88328 hyperlinks between subreddits (edges). The timestamped hyperlink posts have a sentiment label indicating if the post is positive or negative towards the other subreddit. The dynamic graph index was computed using the Graph2Vec [230] embedding approach for 80 epochs, and three summary graphs for each snapshot were generated (union, intersection, and disjoint graphs). The validation of the detected findings is done by comparison to the real historical news coverage.

Initial Setup Per default, our prototype displays the entire graph as an aggregated node-link diagram (supergraph). Then, based on the Kamada-Kawai algorithm [165], a global layout is computed for all snapshot views once. This way, the mental map is preserved during the visual analysis. Furthermore, snapshot views that display more than 100 nodes are automatically clustered using the greedy Clauset-Newman-Moore community detection algorithm [78] to reduce the number of nodes and to extract higher-level properties (e.g., meta-nodes and meta-edges). The clustering of the approximately 8000 nodes of the analyzed data reveals several clusters of subreddits (e.g., computer games subreddits).

Starting Point: Election Week First, the analyst wants to analyze the election week of the 2016 US presidential race (8th November 2016) to identify important groups of political subreddits. The analyst enters the dates of the election week, and the prototype automatically searches for the best fitting snapshot period using the interval tree. The prototype depicts a union graph of election week, and the analyst maps the size of the cluster to the node size to discover large groups of subreddits (see Figure 6.4-A). He selects the largest visible meta-node and all underlying political subreddits of the cluster. He filters these political subreddits as he assumes that the political subreddits of the election week have also been active in the political discourse of the whole election.

Similarity Search To identify political events similar to the election week in the dynamic graph, the analyst searches for similar embeddings using the election weeks supergraph. Using the query interface (see Figure 6.4-B), he queries the five nearest neighbors for each level and sorts the levels by the similarity of the embeddings. The executed nearest neighbor query is calculated on the unfiltered summary graphs for each snapshot, which means that the similarity search results will include false positives that do not necessarily include any political subreddits. The analyst discovers that the query results are similar embeddings on the second (2-hour periods) and third level (4-hour periods), which means that these rather

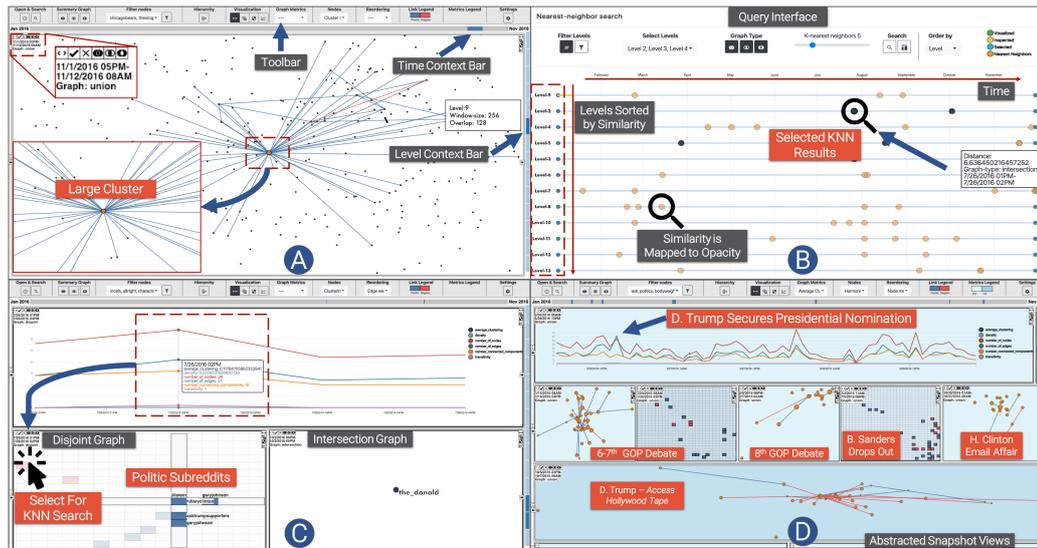


Fig. 6.4 The prototype implementation consists of two primary components the Multiscale Snapshots visualization (A) and the query interface (B). The figures present the visual analysis of the Reddit hyperlink dataset (see Section 6.4.1). The displayed nodes are subreddits, and the edges are timestamped hyperlinks between subreddits with either positive (blue) or negative (red) sentiment. The example illustrates by the case of the 2016 US election how the approach allows searching for similar temporal states in the dynamic graph. The intermediate steps of the visual analysis and the resulting interfaces are presented in the sub-figures C-D. In D, the results of the visual analysis by similarity search are displayed, which are significant events in the timeline of the presidential election.

short sequences of graphs consist of a subset of hyperlinks similar to the ones during the election week. The analyst selects the three closest neighbors for both levels and therefore navigates from a high temporal aggregation (a week) to a lower granularity (2-4 hours). Three queried snapshots are empty, meaning the views do not contain any of the previously filtered political subreddits. The empty views are presumably false positives that capture other graph sub-structures of the election week. The analyst removes the three empty snapshots and examines the remaining three snapshots by changing the visual mapping from a node-link diagram to the time series of graph metrics.

Fine-Grained Temporal Analysis The three remaining snapshots contain a different amount of nodes. The intersection graph on the second level contains only one subreddit (*the_donald*), which means the subreddit was referenced in both graphs of the snapshot (2 hours). The analyst discovers that a high-level summary graph (disjoint) includes a displayed snapshot of the second level. The unexpected overlap steers the analysts towards the low-level disjoint graph, which seems to also be

the peak in the time series of graph metrics. The time series presents the number of nodes, edges, as well as connected graph components, the graph density, the average clustering coefficient, and the transitivity over time. It seems that this second level snapshot is essential for the search results as the snapshot has similar graph structures compared to the supergraph of the election week. The peak is presented as a matrix visualization (see Figure 6.4-C) and can be attributed to the events of the national democratic convention where H. Clinton was nominated for the presidential election. The disjoint graph represented as a matrix visualization (see Figure 6.4-C) can be associated to the political event of the democratic nomination H. Clinton which resulted in a cluster of hyperlinks between political subreddits (e.g., *hillaryclinton*, *asktrumpsupporters*, and *garyjohnson*) and other hyperlinks between political subreddits (e.g., *communism101*, *altright*, and *crazyideas*) The analyst uses the snapshot (disjoint graph) for another similarity search. He expects the similarity search to return more political events because the low-level graph embedding of the two-hour snapshot contains mainly linked political subreddits.

Searching for Political Events The similarity search finds many similar snapshots at different temporal granularities indicating that these political events also seem to be discussed for different periods. The query returns several similar snapshots of the sixth level with an interval length of 32 hours, which can refer to potential political events and their daily news coverage scheme (see Figure 6.4-D). The analyst investigates the different snapshot views, mostly union and disjoint graphs, and abstracts all snapshot views with only a few subreddits. The remaining presented snapshots are on levels 5-7 and contain intervals of 16, 32, and 64 hours. The analyst maps the average clustering coefficient to the background color of each snapshot view to identify periods with structural clusters. He changes the visual metaphors of the dense snapshots to matrix visualizations and the higher-level periods to the time-series metaphor. The different metaphors allow the analyst to put the events on lower levels into the overall temporal context, for example, the analyst can relate how the linkage behavior between subreddits declines after political scandals.

Political Events and Scandals The analyst then visually analyzes the periods and sees that political subreddits link each other during the selected periods, mainly in a positive (blue edge) or negative (red edge) way. Various subreddits such as *the_donald* and *asktrumpsupporters* usually have positive hyperlinks between each other. He examines external resources of the timeline of major events for the 2016 US elections and can refer the presented snapshot views to events in the presidential race. The sixth level of the hierarchy displays several GOP (republican party) political debates, B. Sanders dropping out of the primary election, and the H. Clinton Email affair. The analyst is also able to identify structural changes between the

views, for instance, after B. Sanders drops, the linking activity of some subreddits (e.g., *SandersForPresident* or *Democratic Socialism*) declines. The snapshot view of 6-7th October on the fifth level stands out as it mostly contains negative links between the subreddits. The analyst can relate the period to the leaked tapes of the 2005 Access Hollywood show in which D. Trump brags about sexual exploits and also on the same day WikiLeaks published the email of H. Clintons campaign manager revealing her paid Wall Street speeches. The analyst wants to analyze this snapshot further and displays the supergraph as an animated node-link diagram to examine the news spread between the subreddits on an hourly basis. During the further analysis of snapshot views, the analyst can also detect other events, for instance, the final nomination of D. Trump by the GOP, which results in visible changes in the time series plot of graph metrics. He also detects some events that he cannot directly relate to major political events. Those events are probably general political discussions initiated by Reddit users or targeted news distribution from public-relations groups or political bots. To further investigate such events, the analyst can select these non-assignable events and search for similar periods, for example, to identify the reoccurring post of political bots.

6.4.2 Experimental Evaluation

The generated graph embeddings for the multiscale snapshots are independent of any analytical task and can be used for clustering, graph prediction, and outlier detection. In the following, we show that the multiscale graph embeddings allow us to search for similar sequences of graphs. Across all experiments, we use the same parameter settings to generate the multiscale index.

Problem Background A similarity search for a set of graphs can be interpreted as a query to return k -nearest neighbors to a specific graph. An exhaustive simple brute-force algorithm would compute the distance between all graphs, for example, the graph editing distance (GED) [51] and return the list of k nearest graphs. However, the extensive brute-force approach does not scale as the GED computation is not feasible for graphs with more than 16 nodes [41]. Therefore, heuristics are usually applied to decrease the computation effort of k -nearest neighbor queries, which frequently reduces the accuracy of the results. In the following, we apply window queries for sequences of graphs to show that summarization methods (e.g., union graph) can capture some temporal characteristics.

Datasets We evaluate the performance of similarity searches on synthetic and real-world data. We generated five synthetic dynamic graphs using the dynamic stochastic

block model with diminishing communities [122]. The synthetic datasets consist of 150 nodes with three communities and 100-time steps, containing varying amounts of diminishing communities (up to 20-time steps) in which two nodes are exchanged for each time step. We evaluate the approach with real-world datasets. The Reddit data [188] is a dynamic hyperlink graph with subreddits (nodes) and hyperlinks or crossposts between subreddits (edges). The Wikipedia dataset [237] consists of a dynamic graph that captures the editing behavior (edge) between Wikipedia Talk pages (nodes). For each real-world dataset, we preprocess the data by computing a supergraph for each hour, which generates descriptive dynamic graphs with more than two nodes per time step. We evaluated our approach on randomly picked subsets (100 graphs) of the real-world data. We select a subset of the data as the computation of the following ground truth is quite expensive.

Ground Truth We calculate a ground-truth similarity score for the k -nearest neighbor search by computing the distance between the input and all other graphs. We employ the following similarity measure between two graphs. Our similarity measurement first models the graphs as two adjacency matrices A and B and then compute for each matrix the singular values via the singular value decomposition. Afterward, we calculate the $fnorm$ using

$$fnorm = \sqrt{\sum_{i=0}^S \sigma_i^2}$$

We define the distance between two graphs as

$$madist(A, B) = |fnorm(A) - fnorm(B)|$$

Using the given similarity measurement, we compute the distances between all graphs to obtain a ground-truth of k -nearest neighbors.

Baseline Methods We used three unsupervised graph learning methods on the described datasets. The graph embeddings are applied once with and once without the multiscale temporal modeling. We used the following graph embedding methods [258] with the described input parameters:

- **graph2vec** [230]: 250 epochs, 0.025 learning rate, 2 Weisfeiler-Lehman iterations, and 128 dimensions.
- **GL2Vec** [73]: 250 epochs, 0.025 learning rate, and 128 dimensions.
- **FGSD** [303]: 200 number of histogram bins with a histogram range of 20.

For window queries for the single graph embeddings without any summarization methods, we utilize the median value of the embeddings as the representative value of the interval. We use the median as the average of the embeddings as these can lead to potential distortions in the embedded space. For the multiscale temporal embedding, we applied only one temporal summarization method to generate a union graph for each snapshot. The searched intervals for the k nearest neighbor search are extracted before training of embedding techniques. We randomly extracted five intervals with different lengths (1 – 8) from the dynamic graph and randomly removed one node from each graph.

Evaluation Metrics The following metrics are used to evaluate the approach. We compute the accuracy of the 5-nearest neighbor queries based on the ground-truth. For the accuracy computation, we do not incorporate the ordering of the nearest neighbors and expect only the presence in the result set.

Experimental Setup All experiments were computed on a computer with two CPU cores (Intel i7-6567U 3.30GHz) and 16 GB RAM. The experiment was repeated five times, and the average accuracy was computed for each randomly picked interval with different lengths.

Results The results are described in Table 6.2. The results indicate that FGSD [303] works best to identify nearest neighbors on an embedding basis using the median. The results show that the single graph embeddings have a higher accuracy on the synthetic data. In contrast, the real-world datasets indicate different results by demonstrating equal or improved results by using the multiscale index for longer intervals (< 4). An explanation for this can be the fact that there is a drastic difference between the topology of the synthetic and real-world datasets. For example, the real-world nodes and edges are added and removed more frequently between time steps. The synthetically generated dataset has a quite high density, while in contrast to this, the real-world datasets are much more sparse. For example, in the synthetic data, the nodes are just moved between the clusters, so only edges change over time. These synthetic properties prevent supergraphs from encoding the topological changes over time. Therefore, the multiscale graph index requires a different temporal summarization method to capture the changes of the synthetic dataset (e.g., disjoint graph).

	Synthetic Data					Reddit Data [188]					Wikipedia Data [237]				
Interval length	1	2	3	4	8	1	2	3	4	8	1	2	3	4	8
Graph2Vec	0.096	0.146	0.096	0.15	0.096	0.332	0.066	0	0	0.264	0	0	0	0	0.066
GL2Vec	0.096	0.122	0.024	0.024	0	0.198	0.132	0.266	0.066	0.066	0.066	0	0.066	0	0.066
FGSD	0.146	0.224	0.198	0.198	0.048	0.464	0.332	0	0.132	0.066	0.866	0.132	0.066	0.132	0.134
Multiscale Graph2Vec	0.148	0.072	0.096	0.072	0.122	0.266	0.264	0	0.066	0.264	0.234	0.066	0.066	0.066	0.198
Multiscale GL2Vec	0.148	0.072	0.096	0.072	0.122	0.198	0.132	0.066	0.132	0.4	0.234	0.066	0.198	0.132	0.198
Multiscale FGSD	0.146	0.096	0.096	0.072	0.122	0.264	0.264	0	0	0.332	0.466	0.066	0.2	0.198	0.266

Tab. 6.2 The Table presents the quantitative evaluation results of the k-nearest neighbor search with and without the multiscale graph index for different graph embedding methods. The average accuracy values for window queries of different lengths (1-8) are depicted for each dataset. The experiment was repeated five times on synthetically generated data and with randomly selected subsets of real-world data. The evaluation results indicate an improved accuracy on window queries on the listed real-world dataset.

6.5 Discussion

The Multiscale Snapshots approach consists of three steps: (1) applying temporal summarization methods, (2) utilizing graph embedding methods to reduce the size of the generated graph summaries, and (3) the visual analysis of the generated snapshots. Our quantitative evaluation indicates the usefulness of the multiscale graph embeddings, and the usage scenario shows the application of the approach to real-world data. Overall the utility of the approach yet depends on multiple aspects (e.g., summarization method and graph embedding), the data characteristics (e.g., data distribution), and the task at hand (e.g., outlier analysis).

Steps (1-2) involve multiple methods with parameters. For instance, the graph embeddings methods require defining the number of layers and epochs. For an analyst, such parameter choices pose a challenge as he has to determine suitable methods and their input parameters to generate useful embeddings. We consider the flexibility of using different temporal summarization methods and graph embeddings as an advantage of our approach and a possibility for future work.

Another challenge for steps (1-2) is the computational scalability for the precomputation of the embeddings. For example, the computation of a dynamic graph of length T with $|V|$ nodes and $|E|$ edges require for only union graphs $O(\log(T) \cdot (|V| + |E|))$ memory and time complexity. We speed up the computation of temporal summaries by parallelizing each level's snapshot generation and using an interval tree. Goyal and Ferrara [123] surveyed the time complexity of graph embeddings, and scalable embeddings run in the time complexity of $O(|E|)$. Due to the time and memory complexities, we suggest computing the graph embeddings for large-scale dynamic graphs on a server.

Step (3) aims to display the temporal dimension at multiple scales, which poses new user-related aesthetic challenges [30]. To preserve the mental map, we compute and use only one layout for each applied visual metaphor (e.g., global node-link diagram layout). An analyst can change the global layout for all snapshot views, for instance, by reordering the cells and rows of the adjacency matrix visualization. The snapshot views can also result in adjacent snapshots displaying different dynamic graph visualizations (e.g., node-link and matrix visualization). Consequently, the mental map between such views cannot be preserved as it is not possible to track and identify changes efficiently. We provide brushing and linking methods to minimize the cognitive load of identifying nodes in different visual metaphors. Another limitation of our approach is the fact that specific snapshots can be mistaken for other periods (temporal aliases [30]). We aim to overcome such temporal aliases by displaying the period in each snapshot view and the time context bar highlighting the underlying period in the overall temporal context. We consider these aesthetic challenges [30] as open possibilities for developing new methods for the interactive comparison of two or more snapshots at different granularities. For example, investigating how such mixed visual metaphors impact the overall user experiences poses an opportunity for future work.

The applied methods during the visual analysis influence our approach's computational and visual scalability. For instance, the live computation of displayed graph summaries scales linearly to the number of time steps and the size of the evolving graphs. Furthermore, the real-time analysis of snapshots can suffer based on the algorithmic time complexities of applied methods, for example, the Clauset-Newman-Moore community detection algorithm [78]. A possible solution to these challenges is to investigate how graph embeddings can be utilized to guide an analyst towards temporal changes to speed up the analysis process. Furthermore, the display space limits the visual scalability and readability of structural properties in a snapshot view since they depend on the number of presented snapshots. To address this, we limit and automatically abstract the number of depicted snapshot views to provide visually readable representations. The limit for the number of snapshots is adjustable and is, as a heuristic, limited to six snapshot views for each level. The visual scalability can also be increased by adapting the visual metaphors based on graph properties, such as automatically presenting matrix-based visualization for dense graphs.

We showed the approach's applicability through the visual analysis of similar periods in a dynamic hyperlink graph, which required an initial starting point for the similarity search (e.g., the election week). An analyst has to be aware of such states in advance or apply automated analysis methods to identify them, for instance, by using change-point detection [7] algorithms on the graph embeddings. Furthermore, the

variety of functionality also affects the usability of the approach since the application prototype can be challenging to use for untrained users. In general, the usage of user guidance in combination with the potential application of more automatic analysis methods (e.g., outlier detection algorithms), can help to set high-level snapshots in the context of low-level snapshots, drill down the temporal hierarchy, and steer the user towards a useful combination of data and visual transformations to highlight specific trends. For example, the utilization of sub-queries in the temporal hierarchy can be used to steer an analyst towards fine-grained states with particular graph properties (e.g., motifs).

A limitation of our work is the lack of a formal comparative study to compare Multiscale Snapshots with other visual analytics approaches. In general, such a comparison remains challenging as our approach allows us to integrate visualization techniques (e.g., van Elzen et al. [103]), which is a simple way to increase the overall task coverage. Despite the shortage of a comparative study, our quantitative evaluation and the usage scenario highlight key benefits of our approach, such as the multiscale embedding of sequences of graphs to speed up analytical tasks (e.g., similarity search). Graph embeddings come with the sacrifice of information loss compared to methods such as the computation of graph editing distance (GED) [51]. In future work, we aim to overcome shortcomings by integrating new visual metaphors to allow analysts to examine snapshots and their graph embeddings to understand and interpret the quality of the underlying graph embeddings.

6.6 Conclusion

In this chapter, we presented Multiscale Snapshots, a visual analytics approach, to provide an overview of a dynamic graph. The approach consists of three steps: creating multiscale temporal summaries, applying graph embeddings, and semi-automatic visual analysis. The combination of the steps enables us to visually explore how temporal and structural properties affect the overall dynamic graph. We implemented a prototype and showed in a quantitative evaluation that the approach helps to identify similar temporal states in artificial and real-world dynamic graphs. We also show the applicability by a usage scenario analyzing a real-world dataset, demonstrating that patterns in dynamic graphs can be visually analyzed over time. The application of Multiscale Snapshots and the underlying multiscale temporal analysis paradigm is not limited to dynamic graphs and can be extended to work with any temporal data. For instance, the Multiscale Snapshots approach can be adjusted to support the user-driven analysis of multivariate time-series data.

Conclusions and Future Work

7.1 Conclusion

This thesis presented studies for enhancing the multiscale visual analysis of dynamic networks. The proposed visualizations combine automated analysis methods with interactive visualizations to provide an overview of large-scale dynamic networks at different abstraction scales. The presented multiscale dynamic network visualizations scale to larger datasets, produce less clutter, and reveal the emergence of patterns at different abstraction scales. Moreover, the interactive visualizations help understand the implications of abstraction methods, identify useful abstraction scales, and present the data in a readable and scalable manner.

Overall, the thesis makes several contributions to information visualization and visual analytics research. This thesis presents studies to improve and advance the multiscale visual analysis of dynamic networks. The following paragraph provides an overall picture, summarizing each chapter's contributions. First, Chapter 2 presents a comprehensive overview and taxonomy of multiscale visualizations. Researchers and practitioners can use the literature analysis to understand common design practices, trends, and research gaps to create new multiscale navigation and visualization techniques. Next, Chapter 3 proposes a design study for the multiscale visual abstraction of spatio-temporal networks in the field of collective animal behavior. The proposed glyph designs enable domain experts to seamlessly encode relationships between individuals and groups of movers to reveal emergent group properties over time. In addition, the chapter presents a spatio-temporal clustering benchmark for the field of collective animal behavior. Chapter 4 presents *dg2pix*, a pixel-based visualization to provide a scalable overview of temporal and structural changes in dynamic networks. The multiscale visualization technique reveals changes and similar temporal states in a dynamic network. *dg2pix* also enables analysts to interactively adapt the temporal analysis scale to compare high-level and fine-grained structural changes. Chapter 5 proposes two complementary pixel-based visualizations based on motif and graphlet network analysis to provide a time-scalable overview of dynamic networks. The proposed visualizations allow exploring significant topological motif changes to reveal similar temporal states, trends, and outliers in different-sized networks. Finally,

Chapter 6 presents Multiscale Snapshots, a visual analytics approach to visually analyze temporal and structural summaries of dynamic networks at multiple temporal scales. The approach enables analysts to retrace dynamic patterns, changing graph structures, and similar temporal summaries by exploring both structural and temporal aggregates in a dynamic network.

"How can we enhance the multiscale visual analysis of dynamic networks?" was the driving research question of this thesis. Thus, this thesis proposed designs and visualization approaches to visually analyze large-scale dynamic networks in a readable and scalable way. We learned valuable lessons from analyzing common design practices for multiscale visualization and developing the presented multiscale dynamic network visualizations. First, multiscale visualizations reduce visual clutter and enhance visual scalability. However, such multiscale visualizations are relatively challenging to use for untrained analysts and hence require the integration of semi-automatic analysis methods to facilitate the exploration of large-scale datasets. For instance, the Multiscale Snapshots approach utilizes a hierarchical temporal abstraction combined with unsupervised graph learning methods to semi-automatically explore similar temporal summaries in a dynamic network. Second, visualization practitioners must design multiscale dynamic network visualizations based on task and requirement analyses for particular application domains. In such applications, the utilized abstraction methods and visual metaphors have to be adjustable based on the analysis scale, ideally leveraging already existing domain-specific visual metaphors. For example, the MotionGlyphs design study combines data and visual abstraction methods to simplify and abstract dense spatio-temporal networks based on a domain-specific requirements analysis. Third, multiscale dynamic network visualizations have to abstract both the relational and temporal data aspects and display data abstraction measurements to help analysts assess and understand the effects of abstraction methods across different scales. For instance, in dg2pix, the multiscale pixel visualization allows analysts to visually analyze temporal changes and similar temporal states across different temporal scales while maintaining an overview of the visualized temporal granularities. Finally, multiscale visualizations must present high-level overviews with low-level details at the same time to reveal and retrace structural changes in a dynamic network. Therefore, this thesis proposed several dense pixel-based visualizations to display large amounts of abstracted dynamic network data without overlap and clutter while using the whole display space. For example, we proposed two complementary pixel-based visualizations based on motif and graphlet network analysis methods to provide an overview and detailed view of changing network structures over time.

In summary, this thesis presented multiscale visualizations to analyze long sequences of large-scale dynamic networks. The presented studies allow analysts to visualize, navigate, and relate the relational and temporal data across multiple abstraction scales. The main contributions are the multiscale visualization taxonomy and the presented multiscale dynamic network visualizations, including MotionGlyphs, *dg2pix*, the motif-based pixel visualizations, and the Multiscale Snapshots approach. The presented multiscale visualizations provide an overview of large-scale dynamic networks and allow identifying, comparing, tracing, and interpreting similar network structures over time. We showed the usefulness and applicability of each approach through use cases, benchmarks, or domain expert evaluations. The presented studies and visualization approaches are also generalizable to other application domains with similar network analysis tasks. The proposed multiscale visualizations are released as open-source projects and available online (see Section 1.3).

7.2 Future Perspectives

The following paragraphs highlight general promising research challenges and future perspectives for the multiscale visual analysis of dynamic networks. Moreover, the thesis outlines future work for each respective multiscale dynamic network visualization at the end of each chapter.

Multiscale Comparison of Dynamic Networks A relatively unexplored task is the comparison of multiple networks over time. The main goal of such a comparison has to be the alignment and comparison of relational and temporal data within and across multiple dynamic networks. For example, comparing information diffusion over different temporal granularities is crucial for understanding how information spreads across networks, such as fake news in social networks or information about predators in animal swarms. Such a multiscale network comparison poses interesting visualization and interaction challenges. Ideally, such a visualization enables analysts to compare multiple different-sized networks while still preserving the analyst’s mental model by providing an overview of the temporal ordering and the relationship between the temporal granularities. A potential solution can be the usage of hierarchical temporal aggregation and dimensionality reduction methods comparable to the Multiscale Snapshots approach. Yet, visualization practitioners need to develop novel multiscale comparison and interaction methods to align and compare multiple networks over different temporal scales.

Motifs in Dynamic Networks Various network visualizations utilize motifs to abstract and present underlying similar sub-network structures. However, visualization approaches rarely use network motifs to explore structural changes in dynamic networks. Such dynamic network motif visualizations can help gain insight into specific communication patterns or interaction mechanisms in dynamic networks. For instance, this thesis presented approaches for visually analyzing structural motif changes over time. Therefore, future dynamic network visualizations must support the visual analysis of temporal motifs [237], i.e., a sequence of motifs in the dynamic network. However, visualizing temporal motifs remains challenging since they can also occur at different temporal granularities, thus, requiring multiscale visualizations. A potential method for visualizing temporal motifs is displaying the patterns in a multiscale pixel visualization, displaying temporal motifs in a scalable and readable way.

Multivariate Dynamic Networks Nodes and edges in dynamic networks often have time-varying multivariate attributes. For example, in social hyperlink networks [188], edge attributes can be text messages of varying lengths with more than additional 80 attributes. The primary challenge for multivariate dynamic network visualization is encoding the node and edge attributes within a possible multiscale relational and temporal analysis scale. A potential method for encoding multiple additional node attributes is the usage of multiscale temporal glyphs or displaying the average attributes over time as line charts in a hierarchy of small multiples, similar to the Multiscale Snapshots approach. In such cases, the simultaneous visualization of evolving network structure and changing attributes is essential for understanding how structural shifts potentially influence node or edge attribute changes.

User Guidance for Multiscale Visualizations Visually analyzing large-scale datasets with multiscale visualizations is often challenging for untrained analysts due to the number of abstraction methods and scales, having numerous input parameters. For example, analysts often need to specify the temporal discretization scales in advance based on the application domain and the underlying user tasks. The number of potential abstraction methods and scales, including their input parameters, affects the usability of multiscale visualizations. Thus, developing multiscale user guidance methods is necessary to guide analysts semi-automatically toward useful abstraction methods, scales, and input parameters. For instance, the Multiscale Snapshots approach helps analysts to semi-automatically search in a hierarchy of temporal summaries to reveal similar network structures over time. Visualization practitioners need to develop new user guidance methods to help guide analysts in the multiscale visual exploration of large-scale datasets.

Graph Representation Learning Recently, researchers proposed novel deep learning methods to help understand large-scale networks, such as graph representation learning [134]. The proposed methods automatically learn relevant network features without any feature engineering. The proposed graph representation learning methods perform better than most state-of-the-art methods for particular tasks (e.g., node classification [126]). However, the proposed methods are mainly black-box models, which remain difficult to explain and understand. Thus, developing novel multiscale visual analytics systems that integrate graph representation learning methods can help understand, explain, and debug such black-box models. A potential solution is the visual exploration of the latent spaces, helping generate initial hypotheses about such black box models. For instance, the visual exploration of the latent space using *dg2pix* and displaying the underlying network structures can help to understand which potential network properties have been learned in each latent space dimension.

List of Figures

1.1 Thesis overview.	5
2.1 Commonly used expression in multiscale visualization (122 papers).	19
2.2 Multiscale visualization paper types for the years 1995-2020.	24
2.3 Multiscale visualization evaluation for the years 1995-2020.	26
2.4 Overview of the multiscale visualization taxonomy classes.	27
2.5 The visualization idioms for the multiscale visual representation sub-classes.	28
2.6 Multiscale visualization application areas.	29
2.7 Multiscale visualization evaluation methods.	32
3.1 Overview of MotionGlyphs displaying golden shiner fish	41
3.2 MotionGlyphs prototype displaying 151 golden shiner fish.	49
3.3 MotionGlyphs single and cluster glyph.	51
3.4 MotionGlyphs alternative designs.	54
3.5 MotionGlyphs use case of 151 golden shiner fish.	59
3.6 Spatio-temporal clustering benchmark results.	65
4.1 Synthetic data represented as <i>dg2pix</i>	71
4.2 <i>dg2pix</i> pipeline.	74
4.3 Overview of <i>dg2pix</i> prototype.	78
4.4 <i>dg2pix</i> zoom context bar.	79
4.5 <i>dg2pix</i> synthetic use case.	84
4.6 <i>dg2pix</i> random graph - synthetic use case.	85
4.7 <i>dg2pix</i> Reddit use case.	87
5.1 Network-level census visualization pipeline.	100
5.2 Motif-based dynamic network visualization prototype.	102
5.3 Visual patterns and their interpretation for the motif-based visualizations.	105
5.4 Motif-based dynamic network visualization synthetic use case.	108
5.5 Motif-based dynamic network visualization real-world use case.	111
5.6 Motif-based dynamic network visualization real-world Reddit data details.	113
6.1 Overview of the Multiscale Snapshots pipeline.	125
6.2 Multiscale Snapshots: temporal hierarchy generation example.	126

6.3 Multiscale Snapshots: hierarchical snapshot view. 130
6.4 Multiscale Snapshots Reddit use case. 135

List of Tables

2.1 Multiscale visualization coding scheme with tags.	22
3.1 MotionGlyphs comparison to related work.	48
5.1 Motif-based dynamic network visualization related work comparison. . .	98
6.1 Multiscale Snapshots comparison of related work.	123
6.2 Multiscale Snapshots quantitative evaluation.	140

Bibliography

- [1]James Abello and Frank Van Ham. “Matrix zoom: A visual interface to semi-external graphs”. In: *IEEE Symposium on Information Visualization*. IEEE. 2004, pp. 183–190. DOI: 10.1109/INFVIS.2004.46 (cit. on p. 120).
- [2]Charu C Aggarwal. “Outlier analysis”. In: *Data Mining*. Springer. 2015, pp. 237–263. DOI: 10.1007/978-3-319-47578-3 (cit. on p. 131).
- [3]KP Agrawal, Sanjay Garg, Shashikant Sharma, and Pinkal Patel. “Development and validation of OPTICS based spatio-temporal clustering technique”. In: *Information Sciences* 369 (2016), pp. 388–401. DOI: 10.1016/j.ins.2016.06.048 (cit. on pp. 61, 62, 67).
- [4]Jae-wook Ahn, Catherine Plaisant, and Ben Shneiderman. “A task taxonomy for network evolution analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.3 (2014), pp. 365–376. DOI: 10.1109/TVCG.2013.238 (cit. on pp. 97, 98, 102, 106, 107, 110).
- [5]Jae-wook Ahn, Meirav Taieb-Maimon, Awalin Sopan, Catherine Plaisant, and Ben Shneiderman. “Temporal visualization of social network dynamics: Prototypes for nation of neighbors”. In: *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer. 2011, pp. 309–316. DOI: 10.1007/978-3-642-19656-0_43 (cit. on p. 120).
- [6]Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. “Visual methods for analyzing time-oriented data”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.1 (2008), pp. 47–60. DOI: 10.1109/TVCG.2007.70415 (cit. on p. 120).
- [7]Leman Akoglu and Christos Faloutsos. “Event detection in time series of mobile communication graphs”. In: *Army Science Conference*. Vol. 1. 2010 (cit. on p. 141).
- [8]Mark Alber, Adrian Buganza Tepole, William R Cannon, Suvaranu De, Salvador Duran-Bernal, Krishna Garikipati, George Karniadakis, William W Lytton, Paris Perdikaris, Linda Petzold, et al. “Integrating machine learning and multiscale modeling perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences”. In: *NPJ Digital Medicine* 2.1 (2019), pp. 1–11. DOI: 10.1038/s41746-019-0193-y (cit. on p. 17).
- [9]Keith Andrews. “Evaluation Comes in Many Guises”. English. In: *CHI 2008 Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV '08)*. 2008 (cit. on pp. 22, 23).

- [10]Gennady Andrienko and Natalia Andrienko. “A general framework for using aggregation in visual exploration of movement data”. In: *The Cartographic Journal* 47.1 (2010), pp. 22–40. DOI: 10.1179/000870409X12525737905042 (cit. on p. 46).
- [11]Gennady Andrienko, Natalia Andrienko, Sebastian Bremm, Tobias Schreck, Tatiana Von Landesberger, Peter Bak, and Daniel Keim. “Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns”. In: *Computer Graphics Forum*. Vol. 29. Wiley Online Library. 2010, pp. 913–922. DOI: 10.1111/j.1467-8659.2009.01664.x (cit. on p. 52).
- [12]Gennady Andrienko, Natalia Andrienko, Georg Fuchs, and Jo Wood. “Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.9 (2016), pp. 2120–2136. DOI: 10.1109/TVCG.2016.2616404 (cit. on pp. 42–44, 46–48, 52, 54).
- [13]Natalia Andrienko and Gennady Andrienko. “Spatial generalization and aggregation of massive movement data”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.2 (2010), pp. 205–219. DOI: 10.1109/TVCG.2010.44 (cit. on pp. 47, 48).
- [14]Natalia Andrienko, Gennady Andrienko, Fabian Patterson, and Hendrik Stange. “Visual Analysis of Place Connectedness by Public Transport”. In: *IEEE Transactions on Intelligent Transportation Systems* (2019). DOI: 10.1109/TITS.2019.2924796 (cit. on p. 43).
- [15]Mohd Yousuf Ansari, Amir Ahmad, Shehroz S Khan, Gopal Bhushan, et al. “Spatiotemporal clustering: a review”. In: *Artificial Intelligence Review* 53.4 (2019), pp. 1–43. DOI: 10.1007/s10462-019-09736-1 (cit. on p. 61).
- [16]Daniel Archambault, James Abello, Jessie Kennedy, Stephen Kobourov, Kwan-Liu Ma, Silvia Miksch, Chris Muelder, and Alexandru C Telea. “Temporal multivariate networks”. In: *Multivariate Network Visualization*. Springer, 2014, pp. 151–174. DOI: 10.1007/978-3-319-06793-3_8 (cit. on pp. 7, 124).
- [17]Daniel Archambault and Neil Hurley. “Visualization of trends in subscriber attributes of communities on mobile telecommunications networks”. In: *Social Network Analysis and Mining* 4.1 (2014), pp. 1–17. DOI: 10.1007/s13278-014-0205-5 (cit. on pp. 96–98).
- [18]Daniel Archambault, Tamara Munzner, and David Auber. “GrouseFlocks: Steerable exploration of graph hierarchy space”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.4 (2008), pp. 900–913. DOI: 10.1109/TVCG.2008.34 (cit. on pp. 35, 120).
- [19]Daniel Archambault, Helen Purchase, and Bruno Pinaud. “Animation, small multiples, and the effect of mental map preservation in dynamic graphs”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.4 (2011), pp. 539–552. DOI: 10.1109/TVCG.2010.78 (cit. on pp. 43, 119, 122).

- [20]Richard Arias-Hernandez, Linda T Kaastra, Tera M Green, and Brian Fisher. “Pair analytics: Capturing reasoning processes in collaborative visual analytics”. In: *2011 44th Hawaii International Conference on System Sciences*. IEEE. 2011, pp. 1–10. DOI: 10.1109/HICSS.2011.339 (cit. on p. 41).
- [21]Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. “Spatio-temporal data mining: A survey of problems and methods”. In: *ACM Computing Surveys (CSUR)* 51.4 (2018), pp. 1–41. DOI: 10.1145/3161602 (cit. on p. 61).
- [22]David Auber, Yves Chiricota, Fabien Jourdan, and Guy Melançon. “Multiscale visualization of small world networks”. In: *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714)*. IEEE. 2003, pp. 75–81. DOI: 10.1109/INFVIS.2003.1249011 (cit. on p. 28).
- [23]Benjamin Bach, Nathalie Henry-Riche, Tim Dwyer, Tara Madhyastha, J-D Fekete, and Thomas Grabowski. “Small MultiPiles: Piling time to explore temporal patterns in dynamic networks”. In: *Computer Graphics Forum*. Vol. 34. Wiley Online Library. 2015, pp. 31–40. DOI: 10.1111/cgf.12615 (cit. on pp. 43, 119, 120, 123).
- [24]Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. “Graphdiaries: Animated transitions and temporal navigation for dynamic networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 20 (2014), pp. 740–754. DOI: 10.1109/TVCG.2013.254 (cit. on pp. 1, 96, 98).
- [25]Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. “Visualizing dynamic networks with matrix cubes”. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM. 2014, pp. 877–886. DOI: 10.1145/2556288.2557010 (cit. on pp. 43, 120).
- [26]Benjamin Bach, Conglei Shi, Nicolas Heulot, Tara Madhyastha, Tom Grabowski, and Pierre Dragicevic. “Time curves: Folding time to visualize patterns of temporal evolution in data”. In: *IEEE Transactions on Visualization and Computer Graphics* 22 (2015), pp. 559–568. DOI: 10.1109/TVCG.2015.2467851 (cit. on pp. 72, 119, 122, 123).
- [27]Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. “Gephi: an open source software for exploring and manipulating networks”. In: *Third International AAAI Conference on Weblogs and Social Media*. 2009 (cit. on p. 55).
- [28]Leilani Battle and Jeffrey Heer. “Characterizing exploratory visual analysis: A literature review and evaluation of analytic provenance in tableau”. In: *Computer Graphics Forum*. Vol. 38. 3. Wiley Online Library. 2019, pp. 145–159. DOI: 10.1111/cgf.13678 (cit. on pp. 22, 23).
- [29]Benedikt Bäuml, Ina Boesecke, Raphael Buchmüller, Yannick Metz, Juri Buchmüller, Eren Cakmak, Wolfgang Jentner, and Daniel A. Keim. “Interactive Webtool for Temporal Data and Visual Audio Analysis”. In: *IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC2)*. 2018 (cit. on p. 12).
- [30]Fabian Beck, Michael Burch, and Stephan Diehl. “Towards an aesthetic dimensions framework for dynamic graph visualisations”. In: *2009 13th International Conference Information Visualisation*. IEEE. 2009, pp. 592–597. DOI: 10.1109/IV.2009.42 (cit. on pp. 70, 79, 122, 124, 141).

- [31] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. “A taxonomy and survey of dynamic graph visualization”. In: *Computer Graphics Forum*. Vol. 36. Wiley Online Library. 2017, pp. 133–159. DOI: 10.1111/cgf.12791 (cit. on pp. 40, 42, 43, 47, 70–72, 92, 93, 95, 96, 98, 118–120, 122–124, 129).
- [32] Fabian Beck, Michael Burch, Corinna Vehlow, Stephan Diehl, and Daniel Weiskopf. “Rapid serial visual presentation in dynamic graph visualization”. In: *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE. 2012, pp. 185–192. DOI: 10.1109/VLHCC.2012.6344514 (cit. on p. 43).
- [33] Benjamin B Bederson, Larry Stead, and James D Hollan. “Pad++: Advances in multi-scale interfaces”. In: *Conference on Human Factors in Computing Systems: Conference companion on Human factors in computing systems*. Vol. 24. 1994, pp. 315–316. DOI: 10.1145/259963.260379 (cit. on p. 120).
- [34] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. “Matrix reordering methods for table and network visualization”. In: *Computer Graphics Forum*. Vol. 35. Wiley Online Library. 2016, pp. 693–716. DOI: 10.1111/cgf.12935 (cit. on pp. 77, 82, 103, 112, 132).
- [35] Michael Behrisch, Michael Blumenschein, Naam Wook Kim, Lin Shao, Mennatallah El-Assady, Johannes Fuchs, Daniel Seebacher, Alexandra Diehl, Ulrik Brandes, Hanspeter Pfister, et al. “Quality metrics for information visualization”. In: *Computer Graphics Forum*. Vol. 37. Wiley Online Library. 2018, pp. 625–662. DOI: 10.1111/cgf.13446 (cit. on p. 96).
- [36] Skye Bender-deMoll and Daniel A McFarland. “The art and science of dynamic network visualization”. In: *Journal of Social Structure* 7.2 (2006), pp. 1–38 (cit. on pp. 2, 74, 75, 120, 122).
- [37] Jacques Bertin. *Semiology of graphics; diagrams networks maps*. Tech. rep. 1983 (cit. on p. 52).
- [38] Anastasia Bezerianos, Fanny Chevalier, Pierre Dragicevic, Niklas Elmquist, and Jean-Daniel Fekete. “Graphdice: A system for exploring multivariate social networks”. In: *Computer Graphics Forum*. Vol. 29. Wiley Online Library. 2010, pp. 863–872. DOI: 10.1111/j.1467-8659.2009.01687.x (cit. on pp. 42, 120).
- [39] Derya Birant and Alp Kut. “ST-DBSCAN: An algorithm for clustering spatial-temporal data”. In: *Data & Knowledge Engineering* 60.1 (2007), pp. 208–221. DOI: 10.1016/j.datak.2006.01.013 (cit. on pp. 61, 63).
- [40] Ayan Biswas, Guang Lin, Xiaotong Liu, and Han-Wei Shen. “Visualization of time-varying weather ensembles across multiple resolutions”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2016), pp. 841–850. DOI: 10.1109/TVCG.2016.2598869 (cit. on p. 30).
- [41] David B Blumenthal and Johann Gamper. “On the exact computation of the graph edit distance”. In: *Pattern Recognition Letters* (2018). DOI: 10.1145/3018661.3018731 (cit. on pp. 107, 137).

- [42]Eric Bonabeau, Guy Theraulaz, Jean-Louis Deneubourg, Serge Aron, and Scott Camazine. “Self-organization in social insects”. In: *Trends in Ecology & Evolution* 12.5 (1997), pp. 188–193. DOI: 10.1016/S0169-5347(97)01048-3 (cit. on p. 60).
- [43]Stephen Bonner, Ibad Kureshi, John Brennan, Georgios Theodoropoulos, Andrew Stephen McGough, and Boguslaw Obara. “Exploring the semantic content of unsupervised graph embeddings: An empirical study”. In: *Data Science and Engineering* 4.3 (2019), pp. 269–289. DOI: 10.1007/s41019-019-0097-5 (cit. on pp. 76, 80, 81).
- [44]Rita Borgo, Johannes Kehrer, David HS Chung, Eamonn Maguire, Robert S Laramée, Helwig Hauser, Matthew Ward, and Min Chen. “Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications.” In: *Eurographics (STARs)*. 2013, pp. 39–63. DOI: 10.2312/conf/EG2013/stars/039-063 (cit. on p. 52).
- [45]Ulrik Brandes and Dorothea Wagner. “Analysis and visualization of social networks”. In: *Graph Drawing Software*. Springer, 2004, pp. 321–340. DOI: 10.1007/978-3-642-18638-7_15 (cit. on pp. 42, 73, 119).
- [46]Matthew Brehmer and Tamara Munzner. “A multi-level typology of abstract visualization tasks”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2376–2385. DOI: 10.1109/TVCG.2013.124 (cit. on pp. 22, 74, 124).
- [47]Juri Buchmüller, Eren Cakmak, Natalia Andrienko, Gennady Andrienko, Jolle W. Jolles, and Daniel A. Keim. “Moving Together: Towards a Formalization of Collective Movement”. In: *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2019. DOI: 10.2312/eurova.20191122 (cit. on p. 12).
- [48]Juri Buchmüller, Dominik Jäckle, Eren Cakmak, Ulrik Brandes, and Daniel A. Keim. “MotionRugs: Visualizing Collective Trends in Space and Time”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 76–86. DOI: 10.1109/TVCG.2018.2865049 (cit. on pp. 12, 73).
- [49]Juri F. Buchmüller, Udo Schlegel, Eren Cakmak, Daniel A. Keim, and Evanthia Dimara. “SpatialRugs: A compact visualization of space and time for analyzing collective movement data”. In: *Computers & Graphics* 101 (2021), pp. 23–34. DOI: 10.1016/j.cag.2021.08.003 (cit. on p. 11).
- [50]Juri F. Buchmüller, Udo Schlegel, Eren Cakmak, Daniel A. Keim, and Evanthia Dimara. “SpatialRugs: Enhancing Spatial Awareness of Movement in Dense Pixel Visualizations”. In: *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2020. DOI: 10.2312/eurova.20201078 (cit. on p. 12).
- [51]Horst Bunke. “What is the distance between graphs”. In: *Bulletin of the EATCS* 20 (1983), pp. 35–39 (cit. on pp. 107, 137, 142).
- [52]Michael Burch, Marcel Hlawatsch, and Daniel Weiskopf. “Visualizing a sequence of a thousand graphs (or even more)”. In: *Computer Graphics Forum*. Vol. 36. Wiley Online Library, 2017, pp. 261–271. DOI: 10.1111/cgf.13185 (cit. on pp. 72, 96).
- [53]Michael Burch, Michael Raschke, Adrian Zeyfang, and Daniel Weiskopf. “A scalable visualization for dynamic data in software system hierarchies”. In: *2017 IEEE Working Conference on Software Visualization (VISSOFT)*. IEEE, 2017, pp. 85–93. DOI: 10.1109/VISSOFT.2017.16 (cit. on p. 73).

- [54]Michael Burch and Thomas Reinhardt. “Dynamic Graph Visualization on Different Temporal Granularities”. In: *2017 21st International Conference Information Visualisation (IV)*. IEEE. 2017, pp. 230–235. DOI: 10.1109/iv.2017.44 (cit. on pp. 118, 121, 123, 129).
- [55]Michael Burch, Corinna Vehlow, Fabian Beck, Stephan Diehl, and Daniel Weiskopf. “Parallel edge splatting for scalable dynamic graph visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2344–2353. DOI: 10.1109/TVCG.2011.226 (cit. on pp. 72, 96, 98).
- [56]Michael Burch and Daniel Weiskopf. “Flip-Book Visualization of Dynamic Graphs.” In: *Int. J. Software and Informatics* 9.1 (2015), pp. 3–21 (cit. on pp. 119, 123).
- [57]E. Cakmak, H. Schäfer, J. Buchmüller, J. Fuchs, T. Schreck, A. Jordan, and D. Keim. “MotionGlyphs: Visual Abstraction of Spatio-Temporal Networks in Collective Animal Behavior”. In: *Computer Graphics Forum* 39.3 (2020), pp. 63–75. DOI: 10.1111/cgf.13963 (cit. on pp. 9, 39).
- [58]Eren Cakmak, Giuliano Castiglia, Wolfgang Jentner, Juri Buchmüller, and Daniel A. Keim. “Visualization For Train Management: Improving Overviews in Safety-critical Control Room Environments”. In: *4th International Symposium on Big Data Visual and Immersive Analytics*. 2018 (cit. on p. 13).
- [59]Eren Cakmak, Johannes Fuchs, Dominik Jäckle, Tobias Schreck, Ulrik Brandes, and Daniel Keim. “Motif-Based Visual Analysis of Dynamic Networks”. In: *Submitted to Visualization in Data Science (VDS)*. 2022 (cit. on pp. 10, 91).
- [60]Eren Cakmak, Alexander Gärtner, Thomas Hepp, Juri Buchmüller, Fabian Fischer, and Daniel A. Keim. “Applying visual analytics to explore and analyze movement data”. In: *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2015, pp. 127–128. DOI: 10.1109/VAST.2015.7347643 (cit. on p. 13).
- [61]Eren Cakmak, Dominik Jackle, Tobias Schreck, Daniel Keim, and Johannes Fuchs. “Multiscale Visualization: A Structured Literature Analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* (2021), pp. 1–1. DOI: 10.1109/TVCG.2021.3109387 (cit. on pp. 8, 15).
- [62]Eren Cakmak, Dominik Jäckle, Tobias Schreck, and Daniel Keim. “dg2pix: Pixel-Based Visual Analysis of Dynamic Graphs”. In: *2020 Visualization in Data Science (VDS)*. 2020, pp. 32–41. DOI: 10.1109/VDS51726.2020.00008 (cit. on pp. 10, 28, 69, 97, 98).
- [63]Eren Cakmak, Manuel Plank, Daniel S. Calovi, Alex Jordan, and Daniel Keim. “Spatio-Temporal Clustering Benchmark for Collective Animal Behavior”. In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Animal Movement Ecology and Human Mobility*. HANIMOB ’21. Beijing, China: Association for Computing Machinery, 2021, pp. 5–8. DOI: 10.1145/3486637.3489487 (cit. on pp. 9, 39).
- [64]Eren Cakmak, Udo Schlegel, Dominik Jäckle, Daniel Keim, and Tobias Schreck. “Multiscale Snapshots: Visual Analysis of Temporal Summaries in Dynamic Graphs”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 517–527. DOI: 10.1109/TVCG.2020.3030398 (cit. on pp. 11, 31, 117).

- [65]Eren Cakmak, Udo Schlegel, Matthias Miller, Juri Buchmüller, Wolfgang Jentner, and Daniel A. Keim. “Interactive Classification Using Spectrograms and Audio Glyphs”. In: *IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC1)*. 2018 (cit. on p. 12).
- [66]Eren Cakmak, Daniel Seebacher, Juri Buchmüller, and Daniel A. Keim. “Time Series Projection to Highlight Trends and Outliers”. In: *IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC2)*. 2018 (cit. on p. 12).
- [67]Daniel S Calovi, Ugo Lopez, Sandrine Ngo, Clément Sire, Hugues Chaté, and Guy Theraulaz. “Swarming, schooling, milling: phase diagram of a data-driven fish school model”. In: *New Journal of Physics* 16.1 (Jan. 2014), p. 015026. DOI: 10.1088/1367-2630/16/1/015026 (cit. on p. 62).
- [68]Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. “Density-based clustering based on hierarchical density estimates”. In: *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172. DOI: 10.1007/978-3-642-37456-2_14 (cit. on pp. 63, 78, 82, 84, 102–104).
- [69]Kelei Cao, Mengchen Liu, Hang Su, Jing Wu, Jun Zhu, and Shixia Liu. “Analyzing the noise robustness of deep neural networks”. In: *IEEE Transactions on Visualization and Computer Graphics* (2020). DOI: 10.1109/TVCG.2020.2969185 (cit. on p. 30).
- [70]Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, eds. *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999 (cit. on p. 7).
- [71]Sheelagh Carpendale. “Evaluating information visualizations”. In: *Information Visualization*. Springer, 2008, pp. 19–45. DOI: 10.1007/978-3-540-70956-5_2 (cit. on pp. 22, 23).
- [72]Serina Chang, Mandy L Wilson, Bryan Lewis, Zakaria Mehrab, Komal K Dudakiya, Emma Pierson, Pang Wei Koh, Jaline Gerardin, Beth Redbird, David Grusky, et al. “Supporting COVID-19 policy response with large-scale mobility-based modeling”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 2632–2642. DOI: 10.1145/3447548.3467182 (cit. on p. 1).
- [73]Hong Chen and Hisashi Koga. “GL2vec: Graph Embedding Enriched by Line Graphs with Edge Features”. In: *International Conference on Neural Information Processing*. Springer. 2019, pp. 3–14. DOI: 10.1007/978-3-030-36718-3_1 (cit. on pp. 76, 78, 81, 83, 84, 86, 87, 128, 138).
- [74]Wei Chen, Fangzhou Guo, Dongming Han, Jacheng Pan, Xiaotao Nie, Jiazhi Xia, and Xiaolong Zhang. “Structure-based suggestive exploration: a new approach for effective exploration of large networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2018), pp. 555–565. DOI: 10.1109/TVCG.2018.2865139 (cit. on pp. 95, 98).
- [75]Myoungsu Cho, Bohyoung Kim, Hee-Joon Bae, and Jinwook Seo. “Stroscope: Multi-Scale Visualization of Irregularly Measured Time-Series Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.5 (2014), pp. 808–821. DOI: 10.1109/TVCG.2013.2297933 (cit. on p. 35).

- [76] Sarvenaz Choobdar, Pedro Ribeiro, Sylwia Bugla, and Fernando Silva. “Comparison of co-authorship networks across scientific fields using motifs”. In: *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE. 2012, pp. 147–152. DOI: 10.1109/ASONAM.2012.34 (cit. on p. 94).
- [77] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. “Hierarchical multiscale recurrent neural networks”. In: *5th International Conference on Learning Representations, ICLR 2017*. 2019. DOI: 10.48550/ARXIV.1609.01704 (cit. on p. 126).
- [78] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. “Finding community structure in very large networks”. In: *Physical Review E* 70.6 (2004), p. 066111. DOI: 10.1103/PhysRevE.70.066111 (cit. on pp. 101, 127, 129, 132, 134, 141).
- [79] Andy Cockburn, Amy Karlson, and Benjamin B Bederson. “A review of overview+ detail, zooming, and focus+ context interfaces”. In: *ACM Computing Surveys (CSUR)* 41.1 (2009), pp. 1–31. DOI: 10.1145/1456650.1456652 (cit. on p. 33).
- [80] Stephen A Cook. “The complexity of theorem-proving procedures”. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. 1971, pp. 151–158. DOI: 10.1145/800157.805047 (cit. on p. 115).
- [81] Iain D Couzin. “Collective cognition in animal groups”. In: *Trends in Cognitive Sciences* 13.1 (2009), pp. 36–43. DOI: 10.1016/j.tics.2008.10.002 (cit. on pp. 40, 44).
- [82] Iain D Couzin, Jens Krause, Richard James, Graeme D Ruxton, and Nigel R Franks. “Collective memory and spatial sorting in animal groups”. In: *Journal of Theoretical Biology* 218.1 (2002), pp. 1–11. DOI: 10.1006/jtbi.2002.3065 (cit. on pp. 44, 62, 67).
- [83] Qingguang Cui, Matthew Ward, Elke Rundensteiner, and Jing Yang. “Measuring Data Abstraction Quality in Multiresolution Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006), pp. 709–716. DOI: 10.1109/TVCG.2006.161 (cit. on pp. 33, 35, 120).
- [84] Weiwei Cui, Xiting Wang, Shixia Liu, Nathalie H Riche, Tara M Madhyastha, Kwan Liu Ma, and Baining Guo. “Let it flow: a static method for exploring dynamic graphs”. In: *2014 IEEE Pacific Visualization Symposium*. IEEE. 2014, pp. 121–128. DOI: 10.1109/PacificVis.2014.48 (cit. on pp. 73, 96, 98, 119, 123).
- [85] Alcebiades Dal Col, Paola Valdivia, Fabiano Petronetto, Fabio Dias, Claudio T Silva, and L Gustavo Nonato. “Wavelet-based visual analysis of dynamic networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.8 (2017), pp. 2456–2469. DOI: 10.1109/TVCG.2017.2746080 (cit. on pp. 119, 123).
- [86] Urška Demšar, Kevin Buchin, Francesca Cagnacci, Kamran Safi, Bettina Speckmann, Nico Van de Weghe, Daniel Weiskopf, and Robert Weibel. “Analysis and visualisation of movement: an interdisciplinary review”. In: *Movement Ecology* 3.1 (2015), p. 5. DOI: 10.1186/s40462-015-0032-y (cit. on pp. 44, 45).

- [87]Frederik L. Dennig, Eren Cakmak, Henrik Plate, and Daniel A. Keim. “VulnEx: Exploring Open-Source Software Vulnerabilities in Large Development Organizations to Understand Risk Exposure”. In: *2021 IEEE Symposium on Visualization for Cyber Security (VizSec)*. 2021, pp. 79–83. DOI: 10.1109/VizSec53666.2021.00014 (cit. on p. 11).
- [88]Pravallika Devineni, Evangelos E Papalexakis, Danai Koutra, A Seza Doğruöz, and Michalis Faloutsos. “One size does not fit all: Profiling personalized time-evolving user behaviors”. In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. 2017, pp. 331–340. DOI: 10.1145/3110025.3110050 (cit. on pp. 75, 99).
- [90]Stephan Diehl and Carsten Görg. “Graphs, they are changing”. In: *International Symposium on Graph Drawing*. Springer. 2002, pp. 23–31. DOI: 10.1007/3-540-36151-0_3 (cit. on pp. 43, 119).
- [91]Somayeh Dodge, Robert Weibel, and Anna-Katharina Lautenschütz. “Towards a taxonomy of movement patterns”. In: *Information Visualization 7.3-4 (2008)*, pp. 240–252. DOI: 10.1057/PALGRAVE.IVS.9500182 (cit. on pp. 60, 61).
- [92]Marian Dörk, Sheelagh Cpendale, and Carey Williamson. “Edgemaps: Visualizing explicit and implicit relations”. In: *Visualization and Data Analysis 2011*. Vol. 7868. International Society for Optics and Photonics. 2011, 78680G. DOI: 10.1117/12.872578 (cit. on pp. 42, 47, 48).
- [93]Marion Dumont, Guillaume Touya, and Cécile Duchêne. “Designing multi-scale maps: Lessons learned from existing practices”. In: *International Journal of Cartography* 6.1 (2020), pp. 121–151. DOI: 10.1080/23729333.2020.1717832 (cit. on p. 33).
- [94]Cody Dunne and Ben Shneiderman. “Motif simplification: improving network visualization readability with fan, connector, and clique glyphs”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2013, pp. 3247–3256. DOI: 10.1145/2470654.2466444 (cit. on pp. 40, 42, 47, 48, 52, 54, 92, 95, 98).
- [95]David Ebert, Kelly Gaither, Yun Jang, and Sonia Lasher-Trapp. “Cross-scale, multi-scale, and multi-source data visualization and analysis issues and opportunities”. In: *Scientific Visualization*. Springer, 2014, pp. 353–360. DOI: 10.1007/978-1-4471-6497-5_28 (cit. on pp. 18, 19).
- [96]Jean-Pierre Eckmann, Elisha Moses, and Danilo Sergi. “Entropy of dialogues creates coherent structures in e-mail traffic”. In: *Proceedings of the National Academy of Sciences* 101.40 (2004), pp. 14333–14337. DOI: 10.1073/pnas.0405728101 (cit. on p. 1).
- [97]Niklas Elmqvist, Thanh-Nghi Do, Howard Goodell, Nathalie Henry, and Jean-Daniel Fekete. “ZAME: Interactive large-scale graph visualization”. In: *2008 IEEE Pacific Visualization Symposium*. IEEE. 2008, pp. 215–222. DOI: 10.1109/PACIFICVIS.2008.4475479 (cit. on pp. 28, 35, 42, 120).
- [98]Niklas Elmqvist and Jean-Daniel Fekete. “Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.3 (2009), pp. 439–454. DOI: 10.1109/TVCG.2009.84 (cit. on pp. 16, 18, 20, 33, 34, 36).

- [99] Niklas Elmqvist and Jean-Daniel Fekete. “Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.3 (2010), pp. 439–454. DOI: 10.1109/TVCG.2009.84 (cit. on pp. 75, 77, 121, 129).
- [100] Niklas Elmqvist, Nathalie Henry, Yann Riche, and Jean-Daniel Fekete. “Melange: space folding for multi-focus interaction”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2008, pp. 1333–1342. DOI: 10.1145/1357054.1357263 (cit. on p. 34).
- [101] Niklas Elmqvist, John Stasko, and Philippas Tsigas. “DataMeadow: a visual canvas for analysis of large-scale multivariate data”. In: *Information Visualization* 7.1 (2008), pp. 18–33. DOI: 10.1057/palgrave.ivs.9500170 (cit. on p. 54).
- [102] Stef van den Elzen, Danny Holten, Jorik Blaas, and Jarke J van Wijk. “Dynamic network visualization with extended massive sequence views”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.8 (2013), pp. 1087–1099. DOI: 10.1109/TVCG.2013.263 (cit. on pp. 72, 95, 97, 98).
- [103] Stef van den Elzen, Danny Holten, Jorik Blaas, and Jarke J van Wijk. “Reducing snapshots to points: A visual analytics approach to dynamic network exploration”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 1–10. DOI: 10.1109/TVCG.2015.2468078 (cit. on pp. 1, 2, 70, 72, 76, 92, 95, 98, 99, 118, 120, 121, 123, 128, 142).
- [104] Stef van den Elzen and Jarke J van Wijk. “Small multiples, large singles: A new approach for visual data exploration”. In: *Computer Graphics Forum*. Vol. 32. Wiley Online Library. 2013, pp. 191–200. DOI: 10.1111/cgf.12106 (cit. on pp. 40, 42).
- [105] Daniel Engel, Lars Hüttenberger, and Bernd Hamann. “A survey of dimension reduction methods for high-dimensional data analysis and visualization”. In: *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering-Proceedings of IRTG 1131 Workshop 2011*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2012. DOI: 10.4230/OASIcs.VLUDS.2011.135 (cit. on p. 72).
- [106] David Englert, Pablo Martinez-Blasco, Eren Cakmak, Udo Schlegel, Juri Buchmüller, and Daniel A. Keim. “Somewraps - Something with Graphs: Visual Analytics of Network Data”. In: *IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2020 Mini Challenge 1)*. 2020 (cit. on p. 11).
- [107] Paul Erdős and Alfréd Rényi. “On the evolution of random graphs”. In: *The Structure and Dynamics of Networks*. Princeton University Press, 2011, pp. 38–82. DOI: 10.1515/9781400841356.38 (cit. on pp. 107, 109).
- [108] Cesim Erten, Philip J Harding, Stephen G Kobourov, Kevin Wampler, and Gary Yee. “GraphAEL: Graph animations with evolving layouts”. In: *International Symposium on Graph Drawing*. Springer. 2003, pp. 98–110. DOI: 10.1007/978-3-540-24595-7_9 (cit. on p. 7).

- [109]Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, pp. 226–231 (cit. on pp. 61, 63).
- [110]Naser Ezzati-Jivan and Michel R Dagenais. “Multi-scale navigation of large trace data: A survey”. In: *Concurrency and Computation: Practice and Experience* 29.10 (2017), e4068. DOI: <https://doi.org/10.1002/cpe.4068> (cit. on p. 19).
- [111]Damien R Farine and Hal Whitehead. “Constructing, conducting and interpreting animal social network analysis”. In: *Journal of Animal Ecology* 84.5 (2015), pp. 1144–1163. DOI: 10.1111/1365-2656.12418 (cit. on pp. 40, 42, 43).
- [112]Michael Farrugia, Neil Hurley, and Aaron Quigley. “Exploring temporal ego networks using small multiples and tree-ring layouts”. In: *Proc. ACHI 2011* (2011), pp. 23–28 (cit. on p. 54).
- [113]Paolo Federico, Jurgen Pfeffer, Wolfgang Aigner, Silvia Miksch, and Lukas Zenk. “Visual analysis of dynamic networks using change centrality”. In: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. IEEE Computer Society. 2012, pp. 179–183. DOI: 10.1109/ASONAM.2012.39 (cit. on pp. 43, 73, 119).
- [114]Thomas MJ Fruchterman and Edward M Reingold. “Graph drawing by force-directed placement”. In: *Software: Practice and Experience* 21.11 (1991), pp. 1129–1164. DOI: 10.1002/spe.4380211102 (cit. on pp. 79, 132).
- [115]Ying-Huey Fua, M.O. Ward, and E.A. Rundensteiner. *Hierarchical parallel coordinates for exploration of large datasets*. 1999, pp. 43–508. DOI: 10.1109/VISUAL.1999.809866 (cit. on p. 29).
- [116]Johannes Fuchs, Fabian Fischer, Florian Mansmann, Enrico Bertini, and Petra Isenberg. “Evaluation of alternative glyph designs for time series data in a small multiple setting”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM. 2013, pp. 3237–3246. DOI: 10.1145/2470654.2466443 (cit. on p. 53).
- [117]Johannes Fuchs, Petra Isenberg, Anastasia Bezerianos, Fabian Fischer, and Enrico Bertini. “The influence of contour on similarity perception of star glyphs”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2251–2260. DOI: 10.1109/TVCG.2014.2346426 (cit. on p. 52).
- [118]George W. Furnas and Benjamin B. Bederson. “Space-Scale Diagrams: Understanding Multiscale Interfaces”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’95. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 234–241. DOI: 10.1145/223904.223934 (cit. on pp. 17, 19, 34).
- [119]Sarah Goodwin, Jason Dykes, Aidan Slingsby, and Cagatay Turkey. “Visualizing multiple variables across scale and geography”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015), pp. 599–608. DOI: 10.1109/TVCG.2015.2467199 (cit. on pp. 8, 16, 19, 33).

- [120]Deborah M. Gordon. “The Ecology of Collective Behavior”. In: *PLOS Biology* 12.3 (Mar. 2014), pp. 1–4. DOI: 10.1371/journal.pbio.1001805 (cit. on p. 40).
- [121]Liang Gou, Shaoke Zhang, Jing Wang, and Xiaolong Zhang. “Tagnetlens: multiscale visualization of knowledge structures in social tags”. In: *Proceedings of the 3rd International Symposium on Visual Information Communication*. 2010, pp. 1–9. DOI: 10.1145/1865841.1865865 (cit. on p. 30).
- [122]Palash Goyal, Sujit Rokka Chhetri, Ninareh Mehrabi, Emilio Ferrara, and Arquimedes Canedo. “Dynamicgem: A library for dynamic graph embedding methods”. In: *arXiv preprint arXiv:1811.10734* (2018). DOI: 10.48550/ARXIV.1811.10734 (cit. on p. 138).
- [123]Palash Goyal and Emilio Ferrara. “Graph embedding techniques, applications, and performance: A survey”. In: *Knowledge-Based Systems* 151 (2018), pp. 78–94. DOI: 10.1016/j.knosys.2018.03.022 (cit. on pp. 73, 89, 140).
- [124]Bart N Green, Claire D Johnson, and Alan Adams. “Writing narrative literature reviews for peer-reviewed journals: secrets of the trade”. In: *Journal of Chiropractic Medicine* 5.3 (2006), pp. 101–117. DOI: 10.1016/S0899-3467(07)60142-6 (cit. on p. 20).
- [125]Martin Greilich, Michael Burch, and Stephan Diehl. “Visualizing the evolution of compound digraphs with TimeArcTrees”. In: *Computer Graphics Forum*. Vol. 28. Wiley Online Library. 2009, pp. 975–982. DOI: 10.1111/j.1467-8659.2009.01451.x (cit. on p. 43).
- [126]Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864. DOI: 10.1145/2939672.2939754 (cit. on p. 147).
- [127]Joachim Gudmundsson and Marc Van Kreveld. “Computing longest duration flocks in trajectory data”. In: *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*. New York, NY, USA: Association for Computing Machinery, 2006, pp. 35–42. DOI: 10.1145/1183471.1183479 (cit. on p. 61).
- [128]Yves Guiard, Michel Beaudouin-Lafon, Julien Bastin, Dennis Pasveer, and Shumin Zhai. “View size and pointing difficulty in multi-scale navigation”. In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. 2004, pp. 117–124. DOI: 10.1145/989863.989881 (cit. on p. 33).
- [129]Diansheng Guo and Xi Zhu. “Origin-destination flow data smoothing and mapping”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2043–2052. DOI: 10.1109/TVCG.2014.2346271 (cit. on pp. 48, 52).
- [130]Steffen Hadlak, Hans-Jorg Schulz, and Heidrun Schumann. “In situ exploration of large dynamic networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2334–2343. DOI: 10.1109/TVCG.2011.213 (cit. on pp. 43, 92, 119, 127).

- [131]Steffen Hadlak, Heidrun Schumann, Clemens H Cap, and Till Wollenberg. “Supporting the visual analysis of dynamic networks by clustering associated temporal attributes”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2267–2276. DOI: 10.1109/TVCG.2013.198 (cit. on pp. 74, 92, 96–99, 119, 120, 122, 123, 126).
- [132]Mustafa Hajij, Bei Wang, Carlos Scheidegger, and Paul Rosen. “Visual Detection of Structural Changes in Time-Varying Graphs Using Persistent Homology”. In: *2018 IEEE Pacific Visualization Symposium (PacificVis)*. 2018, pp. 125–134. DOI: 10.1109/PacificVis.2018.00024 (cit. on pp. 96, 98).
- [133]Sarkis Halladjian, Haichao Miao, David Kouřil, M Eduard Gröller, Ivan Viola, and Tobias Isenberg. “Scale Trotter: Illustrative visual travels across negative scales”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2019), pp. 654–664. DOI: 10.1109/TVCG.2019.2934334 (cit. on p. 16).
- [134]William L Hamilton, Rex Ying, and Jure Leskovec. “Representation learning on graphs: Methods and applications”. In: *arXiv preprint arXiv:1709.05584* (2017). DOI: 10.48550/ARXIV.1709.05584 (cit. on p. 147).
- [135]Jing-Dong J Han, Nicolas Bertin, Tong Hao, Debra S Goldberg, Gabriel F Berriz, Lan V Zhang, Denis Dupuy, Albertha JM Walhout, Michael E Cusick, Frederick P Roth, et al. “Evidence for dynamically organized modularity in the yeast protein–protein interaction network”. In: *Nature* 430.6995 (2004), pp. 88–93. DOI: 10.1038/nature02555 (cit. on p. 1).
- [136]Steve Hanneke, Wenjie Fu, Eric P Xing, et al. “Discrete temporal models of social networks”. In: *Electronic Journal of Statistics* 4 (2010), pp. 585–605. DOI: 10.1214/09-EJS548 (cit. on p. 94).
- [137]Martin Harrigan, Daniel Archambault, Pádraig Cunningham, and Neil Hurley. “Ego-Nav: Exploring networks through egocentric spatializations”. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. 2012, pp. 563–570. DOI: 10.1145/2254556.2254661 (cit. on pp. 95, 98).
- [138]Mark Harrower and Cynthia A Brewer. “ColorBrewer.org: an online tool for selecting colour schemes for maps”. In: *The Cartographic Journal* 40.1 (2003), pp. 27–37. DOI: 10.1179/000870403235002042 (cit. on pp. 50, 77, 100, 101, 129).
- [139]Wayne Hayes, Kai Sun, and Nataša Pržulj. “Graphlet-based measures are suitable for biological network comparison”. In: *Bioinformatics* 29.4 (2013), pp. 483–491. DOI: 10.1093/bioinformatics/bts729 (cit. on pp. 94, 101).
- [140]Christopher Healey and James Enns. “Attention and visual memory in visualization and computer graphics”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.7 (2011), pp. 1170–1188. DOI: 10.1109/TVCG.2011.127 (cit. on pp. 53, 122).
- [141]Florian Heimerl and Michael Gleicher. “Interactive analysis of word vector embeddings”. In: *Computer Graphics Forum*. Vol. 37. Wiley Online Library. 2018, pp. 253–265. DOI: 10.1111/cgf.13417 (cit. on p. 80).

- [142]Nathalie Henry and Jean-Daniel Fekete. “Matrixexplorer: a dual-representation system to explore social networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006), pp. 677–684. DOI: 10.1109/TVCG.2006.160 (cit. on p. 120).
- [143]Nathalie Henry, Jean-Daniel Fekete, and Michael J McGuffin. “NodeTrix: a hybrid visualization of social networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1302–1309. DOI: 10.1109/TVCG.2007.70582 (cit. on p. 43).
- [144]Marcel Hlawatsch, Michael Burch, and Daniel Weiskopf. “Visual adjacency lists for dynamic graphs”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.11 (2014), pp. 1590–1603. DOI: 10.1109/TVCG.2014.2322594 (cit. on p. 43).
- [145]Duong Hoang, Pavol Klacansky, Harsh Bhatia, Peer-Timo Bremer, Peter Lindstrom, and Valerio Pascucci. “A Study of the Trade-off Between Reducing Precision and Reducing Resolution for Data Analysis and Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2018), pp. 1193–1203. DOI: 10.1109/TVCG.2018.2864853 (cit. on p. 35).
- [146]Tomaž Hočevár and Janez Demšar. “A combinatorial approach to graphlet counting”. In: *Bioinformatics* 30.4 (2014), pp. 559–565. DOI: 10.1093/bioinformatics/btt717 (cit. on pp. 94, 115).
- [147]Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. “Stochastic blockmodels: First steps”. In: *Social Networks* 5.2 (1983), pp. 109–137. DOI: 10.1016/0378-8733(83)90021-7 (cit. on pp. 71, 78, 83).
- [148]Paul W Holland and Samuel Leinhardt. “A method for detecting structure in sociometric data”. In: *Social Networks*. Elsevier, 1977, pp. 411–432. DOI: 10.1016/B978-0-12-442450-0.50028-6 (cit. on pp. 94, 114).
- [149]Petter Holme and Jari Saramäki. “Temporal networks”. In: *Physics Reports* 519.3 (2012). Temporal Networks, pp. 97–125. DOI: 10.1016/j.physrep.2012.03.001 (cit. on pp. 1, 7).
- [150]K Hoppe and GJ Rodgers. “Mutual selection in time-varying networks”. In: *Physical Review E* 88.4 (2013), p. 042804. DOI: 10.1103/PhysRevE.88.042804 (cit. on p. 7).
- [151]Yucheng Huang, Lei Shi, Yue Su, Yifan Hu, Hanghang Tong, Chaoli Wang, Tong Yang, Deyun Wang, and Shuo Liang. “Eiffel: Evolutionary Flow Map for Influence Graph Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* (2019). DOI: 10.1109/TVCG.2019.2906900 (cit. on p. 48).
- [152]*InfoVis Paper Types*. Website online: <http://ieevis.org/year/2020/info/call-participation/infovis-paper-types> (cit. on pp. 22, 23).
- [153]P. Isenberg, T. Isenberg, M. Sedlmair, J. Chen, and T. Möller. “Visualization as Seen through its Research Paper Keywords”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 771–780. DOI: 10.1109/TVCG.2016.2598827 (cit. on p. 18).

- [154] Petra Isenberg, Florian Heimerl, Steffen Koch, Tobias Isenberg, Panpan Xu, Charles D Stolper, Michael Sedlmair, Jian Chen, Torsten Möller, and John Stasko. “vispubdata.org: A metadata collection about iee visualization (vis) publications”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.9 (2016), pp. 2199–2206. DOI: 10.1109/TVCG.2016.2615308 (cit. on p. 18).
- [155] Petra Isenberg, Tobias Isenberg, Michael Sedlmair, Jian Chen, and Torsten Möller. *Toward a Deeper Understanding of Visualization Through Keyword Analysis*. Tech. rep. RR-8580. France: INRIA, Aug. 2014 (cit. on p. 18).
- [156] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. “ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software”. In: *PloS One* 9.6 (2014). DOI: 10.1371/journal.pone.0098679 (cit. on p. 101).
- [157] Mikkel R Jakobsen and Kasper Hornbæk. “Interactive visualizations on large and small displays: The interrelation of display size, information space, and scale”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2336–2345. DOI: 10.1109/TVCG.2013.170 (cit. on p. 32).
- [158] Waqas Javed and Niklas Elmqvist. “Exploring the design space of composite visualization”. In: *2012 IEEE Pacific Visualization Symposium*. IEEE. 2012, pp. 1–8. DOI: 10.1109/PacificVis.2012.6183556 (cit. on pp. 22, 23).
- [159] Waqas Javed and Niklas Elmqvist. “Stack zooming for multifocus interaction in skewed-aspect visual spaces”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.8 (2012), pp. 1362–1374. DOI: 10.1109/TVCG.2012.323 (cit. on pp. 16, 121).
- [160] Waqas Javed, Sohaib Ghani, and Niklas Elmqvist. “GravNav: Using a gravity model for multi-scale navigation”. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. 2012, pp. 217–224. DOI: 10.1145/2254556.2254597 (cit. on p. 35).
- [161] Waqas Javed, Sohaib Ghani, and Niklas Elmqvist. “Polyzoom: multiscale and multifocus exploration in 2d visual spaces”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 287–296. DOI: 10.1145/2207676.2207716 (cit. on pp. 32, 34).
- [162] Susanne Jul and George W Furnas. “Critical zones in desert fog: aids to multiscale navigation”. In: *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*. 1998, pp. 97–106. DOI: 10.1145/288392.288578 (cit. on pp. 32–35).
- [163] Ilir Jusufi, Andreas Kerren, and Björn Zimmer. “Multivariate network exploration with JauntyNets”. In: *2013 17th International Conference on Information Visualisation*. IEEE. 2013, pp. 19–27. DOI: 10.1109/IV.2013.3 (cit. on p. 42).
- [164] Linda T Kaastra and Brian Fisher. “Field experiment methodology for pair analytics”. In: *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*. ACM. 2014, pp. 152–159. DOI: 10.1145/2669557.2669572 (cit. on p. 55).

- [165]Tomihisa Kamada, Satoru Kawai, et al. “An algorithm for drawing general undirected graphs”. In: *Information processing letters* 31.1 (1989), pp. 7–15. DOI: 10.1016/0020-0190(89)90102-6 (cit. on p. 134).
- [166]Dominik P Käser, Maneesh Agrawala, and Mark Pauly. “FingerGlass: efficient multi-scale interaction on multitouch screens”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 1601–1610. DOI: 10.1145/1978942.1979175 (cit. on p. 32).
- [167]L Kaufman and PJ Rousseeuw. “Agglomerative nesting (program AGNES) in finding groups in data: an introduction to cluster analysis”. In: *Hoboken* (1990). DOI: 10.1002/9780470316801 (cit. on p. 67).
- [168]Johannes Kehler and Helwig Hauser. “Visualization and visual analysis of multifaceted scientific data: A survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.3 (2012), pp. 495–513. DOI: 10.1109/TVCG.2012.110 (cit. on p. 19).
- [169]Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. “Visual analytics: Definition, process, and challenges”. In: *Information Visualization*. Springer, 2008, pp. 154–175. DOI: 10.1007/978-3-540-70956-5_7 (cit. on p. 8).
- [170]Daniel A Keim. “Designing pixel-oriented visualization techniques: Theory and applications”. In: *IEEE Transactions on Visualization and Computer Graphics* 6.1 (2000), pp. 59–78. DOI: 10.1109/2945.841121 (cit. on pp. 73, 74, 92, 96, 103).
- [171]Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. “Visual analytics: Scope and challenges”. In: *Visual Data Mining*. Springer, 2008, pp. 76–90. DOI: 10.1007/978-3-540-71080-6_6 (cit. on p. 8).
- [172]Daniel A Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. “Challenges in visual data analysis”. In: *Tenth International Conference on Information Visualisation (IV’06)*. IEEE. 2006, pp. 9–16. DOI: 10.1109/IV.2006.31 (cit. on p. 31).
- [173]David Kempe, Jon Kleinberg, and Éva Tardos. “Maximizing the spread of influence through a social network”. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2003, pp. 137–146. DOI: 10.1145/956750.956769 (cit. on p. 1).
- [174]Natalie Kerracher, Jessie Kennedy, and Kevin Chalmers. “A task taxonomy for temporal graph visualisation”. In: *IEEE Transactions on Visualization and Computer Graphics* 21.10 (2015), pp. 1160–1172. DOI: 10.1109/TVCG.2015.2424889 (cit. on pp. 72, 93, 96, 119).
- [175]Andreas Kerren, Achim Ebert, and Jörg Meyer. *Human-Centered Visualization Environments: GI-Dagstuhl Research Seminar, Dagstuhl Castle, Germany, March 5-8, 2006, Revised Papers*. Vol. 4417. Springer, 2007. DOI: 10.1007/978-3-540-71949-6 (cit. on p. 122).
- [176]Ethan Kerzner, Alexander Lex, Crystal Lynn Sigulinsky, Timothy Urness, Bryan W Jones, Robert E Marc, and Miriah Meyer. “Graffinity: Visualizing connectivity in large graphs”. In: *Computer Graphics Forum*. Vol. 36. Wiley Online Library. 2017, pp. 251–260. DOI: 10.1111/cgf.13184 (cit. on pp. 40, 42).

- [177]Seokyeon Kim, Seongmin Jeong, Insoo Woo, Yun Jang, Ross Maciejewski, and David S Ebert. “Data flow analysis and visualization for spatiotemporal statistical data without trajectory information”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.3 (2017), pp. 1287–1300. DOI: 10.1109/TVCG.2017.2666146 (cit. on pp. 43, 48).
- [178]Andrew J King, Gaëlle Fehlmann, Dora Biro, Ashley J Ward, and Ines Fürtbauer. “Re-wilding collective behaviour: an ecological perspective”. In: *Trends in Ecology & Evolution* 33.5 (2018), pp. 347–357. DOI: 10.1016/j.tree.2018.03.004 (cit. on p. 60).
- [179]Slava Kisilevich, Florian Mansmann, Mirco Nanni, and Salvatore Rinzivillo. “Spatio-temporal clustering”. In: *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer, 2010, pp. 855–874. DOI: 10.1007/978-0-387-09823-4_44 (cit. on p. 61).
- [180]Donald E Knuth. *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley Professional, 2014 (cit. on pp. 107, 109).
- [181]Sungahnn Ko, Shehzad Afzal, Simon Walton, Yang Yang, Junghoon Chae, Abish Malik, Yun Jang, Min Chen, and David Ebert. “Analyzing high-dimensional multivariate network links with integrated anomaly detection, highlighting and exploration”. In: *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2014, pp. 83–92. DOI: 10.1109/VAST.2014.7042484 (cit. on pp. 42, 48, 50, 54).
- [182]Wiebke Köpp and Tino Weinkauff. “Temporal treemaps: Static visualization of evolving trees”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2018), pp. 534–543. DOI: 10.1109/TVCG.2018.2865265 (cit. on p. 73).
- [183]David Kouřil, Tobias Isenberg, Barbora Kozlíková, Miriah Meyer, M Eduard Gröller, and Ivan Viola. “HyperLabels—Browsing of Dense and Hierarchical Molecular 3D Models”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.8 (2021), pp. 3493–3504. DOI: 10.1109/TVCG.2020.2975583 (cit. on p. 32).
- [184]Paul L Krapivsky and Sidney Redner. “Organization of growing random networks”. In: *Physical Review E* 63.6 (2001), p. 066123. DOI: 10.1103/PhysRevE.63.066123 (cit. on pp. 107, 109).
- [185]Jens Krause, Stefan Krause, Robert Arlinghaus, Ioannis Psorakis, Stephen Roberts, and Christian Rutz. “Reality mining of animal social systems”. In: *Trends in Ecology & Evolution* 28.9 (2013), pp. 541–551. DOI: 10.1016/j.tree.2013.06.002 (cit. on p. 44).
- [186]Jens Kruger, Jens Schneider, and Rudiger Westermann. “Clearview: An interactive context preserving hotspot visualization technique”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006), pp. 941–948. DOI: 10.1109/TVCG.2006.124 (cit. on p. 27).
- [187]Oleksii Kuchaiev, Tijana Milenković, Vesna Memišević, Wayne Hayes, and Nataša Pržulj. “Topological network alignment uncovers biological function and phylogeny”. In: *Journal of the Royal Society Interface* 7.50 (2010), pp. 1341–1354. DOI: 10.1098/rsif.2010.0063 (cit. on p. 101).

- [188]Srijan Kumar, William L Hamilton, Jure Leskovec, and Dan Jurafsky. “Community interaction and conflict on the web”. In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2018, pp. 933–943. DOI: 10.1145/3178876.3186141 (cit. on pp. 85, 86, 92, 110, 112, 113, 133, 134, 138, 140, 146).
- [189]Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. “Edge weight prediction in weighted signed networks”. In: *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE. 2016, pp. 221–230. DOI: 10.1109/ICDM.2016.0033 (cit. on p. 110).
- [190]Oh-Hyun Kwon, Tarik Crnovrsanin, and Kwan-Liu Ma. “What would a graph look like in this layout? a machine learning approach to large graph visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), pp. 478–488. DOI: 10.1109/TVCG.2017.2743858 (cit. on pp. 95, 98).
- [191]Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. “Empirical studies in information visualization: Seven scenarios”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.9 (2011), pp. 1520–1536. DOI: 10.1109/TVCG.2011.279 (cit. on pp. 22, 23).
- [192]Tatiana von Landesberger, Sebastian Bremm, Jürgen Bernard, and Tobias Schreck. “Smart Query Definition for Content-Based Search in Large Sets of Graphs.” In: *EuroVAST@ EuroVis*. 2010. DOI: 10.2312/PE/EuroVAST/EuroVAST10/007–012 (cit. on pp. 95, 98).
- [193]Tatiana von Landesberger, Melanie Gorner, and Tobias Schreck. “Visual analysis of graphs with multiple connected components”. In: *2009 IEEE Symposium on Visual Analytics Science and Technology*. IEEE. 2009, pp. 155–162. DOI: 10.1109/VAST.2009.5333893 (cit. on pp. 95, 98).
- [194]Tatiana von Landesberger, Melanie Görner, Robert Rehner, and Tobias Schreck. “A system for interactive visual analysis of large graphs using motifs in graph editing and aggregation”. In: *Proc. Int. Fall Workshop on Vision, Modeling, and Visualization*. Vol. 9. Otto-von-Guericke-Universität, Magdeburg. 2009, pp. 331–339 (cit. on p. 95).
- [195]Alexandra Lee, Daniel Archambault, and Miguel Nacenta. “Dynamic Network Plaid: A Tool for the Analysis of Dynamic Networks”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–14. DOI: 10.1145/3290605.3300360 (cit. on pp. 96, 98).
- [196]Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. “Task taxonomy for graph visualization”. In: *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*. ACM. 2006, pp. 1–5. DOI: 10.1145/1168149.1168168 (cit. on p. 2).
- [197]Etienne Lein and Alex Jordan. “Studying the evolution of social behaviour in one of Darwin’s Dreamponds: a case for the Lamprologine shell-dwelling cichlids”. In: *Hydrobiologia* 848.16 (2021), pp. 1–28. DOI: 10.1007/s10750-020-04473-x (cit. on p. 61).

- [198]Fritz Lekschas, Benjamin Bach, Peter Kerpedjiev, Nils Gehlenborg, and Hanspeter Pfister. “HiPiler: visual exploration of large genome interaction matrices with interactive small multiples”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), pp. 522–531. DOI: 10.1109/TVCG.2017.2745978 (cit. on pp. 95, 98).
- [199]Olav Lenz, Frank Keul, Sebastian Bremm, Kay Hamacher, and Tatiana von Landesberger. “Visual analysis of patterns in multiple amino acid mutation graphs”. In: *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE. 2014, pp. 93–102. DOI: 10.1109/VAST.2014.7042485 (cit. on pp. 95, 98).
- [200]Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. <http://snap.stanford.edu/data>. June 2014 (cit. on p. 110).
- [201]Omer Levy, Yoav Goldberg, and Ido Dagan. “Improving distributional similarity with lessons learned from word embeddings”. In: *Transactions of the Association for Computational Linguistics* 3 (2015), pp. 211–225. DOI: 10.1162/tac1_a_00134 (cit. on p. 76).
- [202]Antoine Lhuillier, Christophe Hurter, and Alexandru Telea. “State of the art in edge and trail bundling techniques”. In: *Computer Graphics Forum*. Vol. 36. Wiley Online Library. 2017, pp. 619–645. DOI: 10.1111/cgf.13213 (cit. on pp. 42, 52).
- [203]Chenhui Li, George Baciu, and Yunzhe Wang. “Module-based visualization of large-scale graph network data”. In: *Journal of visualization* 20.2 (2017), pp. 205–215. DOI: 10.1007/s12650-016-0375-5 (cit. on p. 48).
- [204]Quan Li, Kristanto Sean Njotoprawiro, Hammad Haleem, Qiaoan Chen, Chris Yi, and Xiaojuan Ma. “Embeddingvis: A visual analytics approach to comparative network embedding inspection”. In: *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE. 2018, pp. 48–59. DOI: 10.1109/VAST.2018.8802454 (cit. on p. 80).
- [205]Zhenhui Li, Bolin Ding, Jiawei Han, and Roland Kays. “Swarm: Mining Relaxed Temporal Moving Object Clusters”. In: *Proceedings of the VLDB Endowment* 3.1 (2010), pp. 723–734. DOI: 10.14778/1920841.1920934 (cit. on p. 61).
- [206]Yang Liu, Eunice Jun, Qisheng Li, and Jeffrey Heer. “Latent space cartography: Visual analysis of vector space embeddings”. In: *Computer Graphics Forum*. Vol. 38. Wiley Online Library. 2019, pp. 67–78. DOI: 10.1111/cgf.13672 (cit. on pp. 79, 80).
- [207]Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. “Graph summarization methods and applications: A survey”. In: *ACM Computing Surveys (CSUR)* 51.3 (2018), pp. 1–34. DOI: 10.1145/3186727 (cit. on pp. 120, 125, 127).
- [208]David Lloyd and Jason Dykes. “Human-centered approaches in geovisualization design: Investigating multiple methods through a long-term case study”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2498–2507. DOI: 10.1109/TVCG.2011.209 (cit. on pp. 48, 52–55).
- [209]Ugo Lopez, Jacques Gautrais, Iain D Couzin, and Guy Theraulaz. “From behavioural analyses to models of collective motion in fish schools”. In: *Interface Focus* 2.6 (2012), pp. 693–707. DOI: 10.1098/rsfs.2012.0033 (cit. on p. 17).

- [210]Tiancheng Lou, Jie Tang, John Hopcroft, Zhanpeng Fang, and Xiaowen Ding. “Learning to predict reciprocity and triadic closure in social networks”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 7.2 (2013), pp. 1–25. DOI: 10.1145/2499907.2499908 (cit. on p. 94).
- [211]Ryan Lukeman, Yue-Xian Li, and Leah Edelstein-Keshet. “Inferring individual rules from collective behavior”. In: *Proceedings of the National Academy of Sciences* 107.28 (2010), pp. 12576–12580. DOI: 10.1073/pnas.1001763107 (cit. on p. 52).
- [212]Avi Ma’ayan, Sherry L Jenkins, Ryan L Webb, Seth I Berger, Sudarshan P Purushothaman, Noura S Abul-Husn, Jeremy M Posner, Tony Flores, and Ravi Iyengar. “SNAVI: Desktop application for analysis and visualization of large-scale signaling networks”. In: *BMC Systems Biology* 3.1 (2009), pp. 1–11. DOI: 10.1186/1752-0509-3-10 (cit. on pp. 95, 98).
- [213]Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2605 (cit. on pp. 72, 79, 80).
- [214]Tran Hai Dang Mai, Felix Hinderer, Eren Cakmak, Udo Schlegel, Juri Buchmüller, and Daniel A. Keim. “VisMe: Visual and Uncertainty Analysis tool in Object Recognition”. In: *IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2020 Mini Challenge 2)*. 2020 (cit. on p. 11).
- [215]Yury A Malkov and Dmitry A Yashunin. “Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs”. In: *IEEE transactions on Pattern Analysis and Machine Intelligence* (2018). DOI: 10.1109/TPAMI.2018.2889473 (cit. on p. 128).
- [216]Yi Mao, Joshua Dillon, and Guy Lebanon. “Sequential Document Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1208–1215. DOI: 10.1109/TVCG.2007.70592 (cit. on p. 27).
- [217]Ali Masoudi-Nejad, Falk Schreiber, and Zahra Razaghi Moghadam Kashani. “Building blocks of biological networks: a review on major network motif discovery algorithms”. In: *IET Systems Biology* 6.5 (2012), pp. 164–174. DOI: 10.1049/iet-syb.2011.0011 (cit. on p. 114).
- [218]Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. “UMAP: Uniform Manifold Approximation and Projection”. In: *Journal of Open Source Software* 3.29 (2018), p. 861. DOI: 10.21105/joss.00861 (cit. on p. 80).
- [219]Miriah Meyer, Tamara Munzner, and Hanspeter Pfister. “MizBee: a multiscale synten browser”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 897–904. DOI: 10.1109/TVCG.2009.167 (cit. on p. 34).
- [220]Haichao Miao, Elisa De Llano, Johannes Sorger, Yasaman Ahmadi, Tadija Kekic, Tobias Isenberg, M Eduard Gröller, Ivan Barišić, and Ivan Viola. “Multiscale visualization and scale-adaptive modification of DNA nanostructures”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), pp. 1014–1024. DOI: 10.1109/TVCG.2017.2743981 (cit. on pp. 19, 24, 35).

- [221]Haichao Miao, Tobias Klein, David Kouřil, Peter Mindek, Karsten Schatz, M Eduard Gröller, Barbora Kozlíková, Tobias Isenberg, and Ivan Viola. “Multiscale molecular visualization”. In: *Journal of Molecular Biology* 431.6 (2019), pp. 1049–1070. DOI: 10.1016/j.jmb.2018.09.004 (cit. on pp. 8, 16, 18).
- [222]Tijana Milenković, Weng Leong Ng, Wayne Hayes, and Nataša Pržulj. “Optimal network alignment with graphlet degree vectors”. In: *Cancer Informatics* 9 (2010), CIN–S4744. DOI: 10.4137/CIN.S4744 (cit. on p. 94).
- [223]Jarrod Millman and Travis Vaught. “The State of SciPy”. In: *Proceedings of the 7th Python in Science Conference*. Pasadena, CA USA, 2008, pp. 5–10 (cit. on p. 107).
- [224]Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. “Superfamilies of evolved and designed networks”. In: *Science* 303.5663 (2004), pp. 1538–1542 (cit. on pp. 94, 100, 107, 114).
- [225]Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. “Network motifs: simple building blocks of complex networks”. In: *Science* 298.5594 (2002), pp. 824–827. DOI: 10.1126/science.298.5594.824 (cit. on pp. 93, 94).
- [226]Adiel Mittmann, Aldo von Wangenheim, and Alckmar Luiz dos Santos. “A multi-level visualization scheme for poetry”. In: *2016 20th International Conference Information Visualisation (IV)*. IEEE. 2016, pp. 312–317. DOI: 10.1109/IV.2016.64 (cit. on p. 30).
- [227]Haneen Mohammed, Ali K Al-Awami, Johanna Beyer, Corrado Cali, Pierre Magistretti, Hanspeter Pfister, and Markus Hadwiger. “Abstractocyte: A visual tool for exploring nanoscale astroglial cells”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), pp. 853–861. DOI: 10.1109/TVCG.2017.2744278 (cit. on pp. 29, 35).
- [228]James Moody, Daniel McFarland, and Skye Bender-deMoll. “Dynamic Network Visualization”. In: *American Journal of Sociology* 110.4 (2005), pp. 1206–1241 (cit. on p. 120).
- [229]Tamara Munzner. *Visualization analysis and design*. CRC Press, 2014 (cit. on pp. 21–23).
- [230]Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. “graph2vec: Learning distributed representations of graphs”. In: *arXiv preprint arXiv:1707.05005* (2017). DOI: 10.48550/ARXIV.1707.05005 (cit. on pp. 70, 71, 75, 76, 81, 83, 84, 86, 118, 124, 128, 134, 138).
- [231]Mark Newman. *Networks*. Oxford university press, 2018 (cit. on p. 120).
- [232]Mark EJ Newman. “The structure and function of complex networks”. In: *SIAM Review* 45.2 (2003), pp. 167–256. DOI: 10.1126/science.1089167 (cit. on pp. 94, 107).
- [233]C Nobre, M Meyer, M Streit, and A Lex. “The state of the art in visualizing multivariate networks”. In: *Computer Graphics Forum*. Vol. 38. 3. Wiley Online Library. 2019, pp. 807–832. DOI: 10.1111/cgf.13728 (cit. on pp. 40, 42, 47, 72, 93, 119, 122, 123, 129).

- [234]S. Openshaw. *The Modifiable Areal Unit Problem*. Concepts and Techniques in Modern Geography. Geo Books, 1983 (cit. on p. 19).
- [235]Mark Ortmann and Ulrik Brandes. “Efficient orbit-aware triad and quad census in directed and undirected graphs”. In: *Applied Network Science* 2.1 (2017), pp. 1–17. DOI: 10.1007/s41109-017-0027-2 (cit. on p. 100).
- [236]Juraj Pálenik, Jan Byška, Stefan Bruckner, and Helwig Hauser. “Scale-Space Splatting: Reforming Spacetime for Cross-Scale Exploration of Integral Measures in Molecular Dynamics”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2019), pp. 643–653. DOI: 10.1109/TVCG.2019.2934258 (cit. on p. 28).
- [237]Ashwin Paranjape, Austin R Benson, and Jure Leskovec. “Motifs in temporal networks”. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 2017, pp. 601–610. DOI: 10.1145/3018661.3018731 (cit. on pp. 138, 140, 146).
- [238]Ji Hwan Park, Saad Nadeem, and Arie Kaufman. “GeoBrick: exploration of spatiotemporal data”. In: *The Visual Computer* 35.2 (2019), pp. 191–204. DOI: 10.1007/s00371-017-1461-y (cit. on pp. 47, 48).
- [239]Iulian Peca, Georg Fuchs, Katerina Vrotsou, Natalia V Andrienko, and Gennady L Andrienko. “Scalable Cluster Analysis of Spatial Events”. In: *EuroVA@ EuroVis* 6 (2012), pp. 19–23. DOI: 10.2312/PE/EuroVAST/EuroVA12/019-023 (cit. on pp. 50, 51, 63).
- [240]Nikos Pelekis, Gennady Andrienko, Natalia Andrienko, Ioannis Kopanakis, Gerasimos Marketos, and Yannis Theodoridis. “Visually exploring movement data via similarity-based analysis”. In: *Journal of Intelligent Information Systems* 38.2 (2012), pp. 343–391. DOI: 10.1007/s10844-011-0159-2 (cit. on p. 45).
- [241]Grace CY Peng, Mark Alber, Adrian Buganza Tepole, William R Cannon, Suvranu De, Savador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W Lytton, Paris Perdikaris, et al. “Multiscale Modeling Meets Machine Learning: What Can We Learn?” In: *Archives of Computational Methods in Engineering* (2020), pp. 1–21. DOI: 10.1007/s11831-020-09405-5 (cit. on pp. 16, 17).
- [242]Vsevolod Peysakhovich, Christophe Hurter, and Alexandru Telea. “Attribute-driven edge bundling for general graphs with applications in trail analysis”. In: *2015 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2015, pp. 39–46. DOI: 10.1109/PACIFICVIS.2015.7156354 (cit. on p. 40).
- [243]Nicola Pezzotti, Jean-Daniel Fekete, Thomas Höllt, Boudewijn PF Lelieveldt, Elmar Eisemann, and Anna Vilanova. “Multiscale visualization and exploration of large bipartite graphs”. In: *Computer Graphics Forum*. Vol. 37. 3. Wiley Online Library, 2018, pp. 549–560. DOI: 10.1111/cgf.13441 (cit. on pp. 28, 35).
- [244]Robert Pienta, Fred Hohman, Alex Endert, Acar Tamersoy, Kevin Roundy, Chris Gates, Shamkant Navathe, and Duen Horng Chau. “VIGOR: interactive visual exploration of graph query results”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), pp. 215–225. DOI: 10.1109/TVCG.2017.2744898 (cit. on pp. 40, 42).

- [245] Emmanuel Pietriga, Caroline Appert, and Michel Beaudouin-Lafon. “Pointing and beyond: an operationalization and preliminary evaluation of multi-scale searching”. In: *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*. 2007, pp. 1215–1224. DOI: 10.1109/TVCG.2019.2934555 (cit. on p. 32).
- [246] Isabel Piljek, Giuliana Dehn, Jannik Frauendorf, Ziad Salem, Yezahn Niyazbayev, Juri Buchmüller, Eren Cakmak, Wolfgang Jentner, Florian Stoffel, and Daniel A. Keim. “Identifying Patterns and Anomalies within Spatiotemporal Water Sampling Data”. In: *IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC2)*. 2018 (cit. on p. 13).
- [247] Nataša Pržulj. “Biological network comparison using graphlet degree distribution”. In: *Bioinformatics* 23.2 (2007), e177–e183. DOI: 10.1093/bioinformatics/bt1301 (cit. on p. 94).
- [248] Helen C Purchase, Eve Hoggan, and Carsten Görg. “How important is the mental map?—an empirical investigation of a dynamic graph layout algorithm”. In: *International Symposium on Graph Drawing*. Springer. 2006, pp. 184–195. DOI: 10.1007/978-3-540-70904-6_19 (cit. on pp. 43, 71, 119, 122, 131).
- [249] Peter Ranacher and Katerina Tzavella. “How to compare movement? A review of physical movement similarity measures in geographic information science and beyond”. In: *Cartography and Geographic Information Science* 41.3 (2014), pp. 286–307. DOI: 10.1080/15230406.2014.890071 (cit. on p. 66).
- [250] Craig W Reynolds. “Flocks, herds and schools: A distributed behavioral model”. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: Association for Computing Machinery, 1987, pp. 25–34. DOI: 10.1145/37401.37406 (cit. on p. 62).
- [251] Pedro Ribeiro, Pedro Paredes, Miguel EP Silva, David Aparicio, and Fernando Silva. “A Survey on Subgraph Counting: Concepts, Algorithms, and Applications to Network Motifs and Graphlets”. In: *ACM Computing Surveys (CSUR)* 54.2 (2021), pp. 1–36. DOI: 10.1145/3433652 (cit. on pp. 93, 94, 115).
- [252] Gaëlle Richer, Joris Sansen, Frédéric Lalanne, David Auber, and Romain Bourqui. “Enabling hierarchical exploration for large-scale multidimensional data with abstract parallel coordinates”. In: *International Workshop on Big Data Visual Exploration and Analytics 2018*. Vol. 2083. 2018, pp. 76–83 (cit. on p. 29).
- [253] George Robertson, David Ebert, Stephen Eick, Daniel Keim, and Ken Joy. “Scale and complexity in visual analytics”. In: *Information Visualization* 8.4 (2009), pp. 247–253. DOI: 10.1057/ivs.2009.23 (cit. on p. 34).
- [254] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. “Effectiveness of animation in trend visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1325–1332. DOI: 10.1109/TVCG.2008.125 (cit. on pp. 43, 47).

- [255] Daniel Romero and Jon Kleinberg. “The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter”. In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 4. 2010. DOI: 10.48550/ARXIV.1003.2469 (cit. on p. 99).
- [256] Mikkel Rønne Jakobsen and Kasper Hornbæk. “Sizing up visualizations: effects of display size in focus+ context, overview+ detail, and zooming interfaces”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 1451–1460. DOI: 10.1145/1978942.1979156 (cit. on p. 32).
- [257] Sara Brin Rosenthal, Colin R Twomey, Andrew T Hartnett, Hai Shan Wu, and Iain D Couzin. “Revealing the hidden networks of interaction in mobile animal groups allows prediction of complex behavioral contagion”. In: *Proceedings of the National Academy of Sciences* 112.15 (2015), pp. 4690–4695. DOI: 10.1073/pnas.1420068112 (cit. on pp. 40, 45, 53).
- [258] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. “Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs”. In: *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. ACM. 2020. DOI: 10.1145/3340531.3412757 (cit. on pp. 76, 138).
- [259] Sébastien Rufiange and Michael J McGuffin. “DiffAni: Visualizing dynamic graphs with a hybrid of difference maps and animation”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2556–2565. DOI: 10.1109/TVCG.2013.149 (cit. on pp. 119, 123).
- [260] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North, and Daniel A Keim. “Visual interaction with dimensionality reduction: A structured literature analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2016), pp. 241–250. DOI: 10.1109/TVCG.2016.2598495 (cit. on p. 72).
- [261] Fatemeh Salehi Rizi and Michael Granitzer. “Properties of vector embeddings in social networks”. In: *Algorithms* 10.4 (2017), p. 109. DOI: 10.3390/a10040109 (cit. on pp. 80–82, 89).
- [262] Roeland Scheepens, Huub Van De Wetering, and Jarke J Van Wijk. “Non-overlapping aggregated multivariate glyphs for moving objects”. In: *2014 IEEE Pacific Visualization Symposium*. IEEE. 2014, pp. 17–24. DOI: 10.1109/PacificVis.2014.13 (cit. on pp. 47, 48, 52).
- [263] Udo Schlegel, Eren Cakmak, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A. Keim. “Towards Visual Debugging for Multi-Target Time Series Classification”. In: *Proceedings of the 25th International Conference on Intelligent User Interfaces*. IUI '20. Cagliari, Italy: Association for Computing Machinery, 2020, pp. 202–206. DOI: 10.1145/3377325.3377528 (cit. on p. 12).
- [264] Udo Schlegel, Eren Cakmak, Juri Buchmüller, and Daniel A Keim. “G-Rap: Interactive text synthesis using recurrent neural network suggestions”. In: *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning: ESANN 2018*. 2018 (cit. on p. 13).

- [265]Udo Schlegel, Eren Cakmak, and Daniel A. Keim. “ModelSpeX: Model Specification Using Explainable Artificial Intelligence Methods”. In: *Machine Learning Methods in Visualisation for Big Data*. The Eurographics Association, 2020. DOI: 10.2312/mlvis.20201100 (cit. on p. 12).
- [266]Udo Schlegel, Wolfgang Jentner, Juri Buchmueller, Eren Cakmak, Giuliano Castiglia, Renzo Canepa, Simone Petralli, Luca Oneto, Daniel A. Keim, and Davide Anguita. “Visual Analytics for Supporting Conflict Resolution in Large Railway Networks”. In: *Recent Advances in Big Data and Deep Learning*. Cham: Springer International Publishing, 2020, pp. 206–215 (cit. on p. 12).
- [267]Hansjörg Schmauder, Michael Burch, and Daniel Weiskopf. “Visualizing Dynamic Weighted Digraphs with Partial Links”. In: *International Conference on Information Visualization Theory and Applications*. Vol. 2. SCITEPRESS. 2015, pp. 123–130 (cit. on p. 40).
- [268]Falk Schreiber and Henning Schwöbbermeyer. “MAVisto: a tool for the exploration of network motifs”. In: *Bioinformatics* 21.17 (2005), pp. 3572–3574. DOI: 10.1093/bioinformatics/bti556 (cit. on pp. 95, 98).
- [269]Hans-Jorg Schulz, Steffen Hadlak, and Heidrun Schumann. “The design space of implicit hierarchy visualization: A survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.4 (2010), pp. 393–411. DOI: 10.1109/TVCG.2010.79 (cit. on p. 20).
- [270]Hans-Jörg Schulz and Christophe Hurter. “Grooming the hairball-how to tidy up network visualizations?” In: *INFOVIS 2013, IEEE Information Visualization Conference*. 2013 (cit. on p. 120).
- [271]Rita Sevastjanova, Eren Cakmak, Shauli Ravfogel, Ryan Cotterell, and Mennatallah El-Assady. “Visual Comparison of Language Model Adaptation”. In: *IEEE Trans. on Visualization and Computer Graphics* 3 (2022) (cit. on p. 11).
- [272]Zeqian Shen and Kwan-Liu Ma. “Mobivis: A visualization system for exploring mobile data”. In: *2008 IEEE Pacific Visualization Symposium*. IEEE. 2008, pp. 175–182. DOI: 10.1109/PACIFICVIS.2008.4475474 (cit. on p. 54).
- [273]Shai S Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. “Network motifs in the transcriptional regulation network of *Escherichia coli*”. In: *Nature Genetics* 31.1 (2002), pp. 64–68. DOI: /10.1137/S003614450342480 (cit. on p. 96).
- [274]Weiqi Shi, Julie Dorsey, and Holly Rushmeier. “Learning-Based Inverse Bi-Scale Material Fitting from Tabular BRDFs”. In: *IEEE Transactions on Visualization and Computer Graphics* (2020). DOI: 10.1109/TVCG.2020.3026021 (cit. on pp. 22, 23).
- [275]Jamin Shin, Andrea Madotto, and Pascale Fung. “Interpreting word embeddings with eigenvector analysis”. In: *32nd Conference on Neural Information Processing Systems (NIPS 2018), IRASL workshop*. 2018 (cit. on pp. 73, 81).
- [276]Hadi Shiravi, Ali Shiravi, and Ali A Ghorbani. “A survey of visualization systems for network security”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.8 (2011), pp. 1313–1329. DOI: 10.1109/TVCG.2011.144 (cit. on p. 40).

- [277]Ben Shneiderman. “The eyes have it: A task by data type taxonomy for information visualizations”. In: *The Craft of Information Visualization*. Elsevier, 2003, pp. 364–371. DOI: 10.1016/B978-155860915-0/50046-9 (cit. on p. 122).
- [278]Ben Shneiderman and Aleks Aris. “Network visualization by semantic substrates”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006), pp. 733–740. DOI: 10.1109/TVCG.2006.166 (cit. on p. 42).
- [279]Daniel J Simons, Steven L Franconeri, and Rebecca L Reimer. “Change blindness in the absence of a visual disruption”. In: *Perception* 29.10 (2000), pp. 1143–1154. DOI: 10.1068/p3104 (cit. on p. 53).
- [280]Mike Sips, Patrick Köthur, Andrea Unger, Hans-Christian Hege, and Doris Dransch. “A visual analytics approach to multiscale exploration of environmental time series”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2899–2907. DOI: 10.1109/TVCG.2012.191 (cit. on p. 31).
- [281]Aidan Slingsby and E Van Loon. “Exploratory visual analysis for animal movement ecology”. In: *Computer Graphics Forum*. Vol. 35. 3. Wiley Online Library, 2016, pp. 471–480. DOI: 10.1111/cgf.12923 (cit. on pp. 44, 46).
- [282]Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B Viégas, and Martin Wattenberg. “Embedding projector: Interactive visualization and interpretation of embeddings”. In: *arXiv preprint arXiv:1611.05469* (2016). DOI: 10.48550/ARXIV.1611.05469 (cit. on p. 79).
- [283]Tom AB Snijders. “The statistical evaluation of social network dynamics”. In: *Sociological Methodology* 31.1 (2001), pp. 361–395. DOI: 10.1111/0081-1750.00099 (cit. on p. 94).
- [284]Olaf Sporns, Rolf Kötter, and Karl J Friston. “Motifs in brain networks”. In: *PLoS Biology* 2.11 (2004). DOI: 10.1371/journal.pbio.0020369 (cit. on p. 94).
- [285]David Spretke, Peter Bak, Halldor Janetzko, Bart Kranstauber, Florian Mansmann, and Sarah Davidson. “Exploration through enrichment: a visual analytics approach for animal movement”. In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011, pp. 421–424. DOI: 10.1145/2093973.2094038 (cit. on p. 55).
- [286]Klaus Stein, Rene Wegener, and Christoph Schlieder. “Pixel-oriented visualization of change in social networks”. In: *2010 International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 2010, pp. 233–240. DOI: 10.1109/ASONAM.2010.18 (cit. on pp. 73, 96, 98).
- [287]Chris Stolte, Diane Tang, and Pat Hanrahan. “Multiscale visualization using data cubes”. In: *IEEE Transactions on Visualization and Computer Graphics* 9.2 (2003), pp. 176–187. DOI: 10.1109/TVCG.2003.1196005 (cit. on pp. 17, 19, 34, 35).
- [288]David JT Sumpter. *Collective animal behavior*. Princeton University Press, 2010 (cit. on p. 52).

- [289]Wenbo Tao, Xinli Hou, Adam Sah, Leilani Battle, Remco Chang, and Michael Stonebraker. “Kyrix-S: Authoring Scalable Scatterplot Visualizations of Big Data”. In: *IEEE Transactions on Visualization and Computer Graphics* (2020). DOI: 10.1109/TVCG.2020.3030372 (cit. on p. 34).
- [290]Wenbo Tao, Xiaoyu Liu, Yedi Wang, Leilani Battle, Çağatay Demiralp, Remco Chang, and Michael Stonebraker. “Kyrix: Interactive pan/zoom visualizations at scale”. In: *Computer Graphics Forum*. Vol. 38. 3. Wiley Online Library. 2019, pp. 529–540. DOI: 10.1111/cgf.13708 (cit. on p. 34).
- [291]Christian Tominski, Georg Fuchs, and Heidrun Schumann. “Task-driven color coding”. In: *2008 12th International Conference Information Visualisation*. IEEE. 2008, pp. 373–380. DOI: 10.1109/IV.2008.24 (cit. on pp. 77, 130).
- [292]Christian Tominski, Stefan Gladisch, Ulrike Kister, Raimund Dachselt, and Heidrun Schumann. “Interactive lenses for visualization: An extended survey”. In: *Computer Graphics Forum*. Vol. 36. 6. Wiley Online Library. 2017, pp. 173–200. DOI: 10.1111/cgf.12871 (cit. on p. 34).
- [293]Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander Bronstein, and Emmanuel Müller. “Netlsd: hearing the shape of a graph”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2347–2356. DOI: 10.1145/3219819.3219991 (cit. on p. 81).
- [294]John W Tukey. “Exploratory data analysis”. In: *Addison-Wesley Series in Behavioral Science: Quantitative Methods* (1977) (cit. on p. 1).
- [295]Çagatay Turkay, Aidan Slingsby, Helwig Hauser, Jo Wood, and Jason Dykes. “Attribute signatures: Dynamic visual summaries for analyzing multivariate geographical data”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2033–2042. DOI: 10.1109/TVCG.2014.2346265 (cit. on p. 16).
- [296]Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. “Animation: can it facilitate?” In: *International Journal of Human-Computer Studies* 57.4 (2002), pp. 247–262. DOI: 10.1006/ijhc.2002.1017 (cit. on pp. 43, 71, 86, 96, 97, 122).
- [297]Jarke J Van Wijk and Wim AA Nuij. “A model for smooth viewing and navigation of large 2D information spaces”. In: *IEEE Transactions on Visualization and Computer Graphics* 10.4 (2004), pp. 447–458. DOI: 10.1109/TVCG.2004.1 (cit. on p. 35).
- [298]Ricardo Manuel Millán Vaquero, Jan Rzepecki, Karl-Ingo Friese, and Franz-Erich Wolter. “Visualization and user interaction methods for multiscale biomedical data”. In: *3D Multiscale Physiological Human*. Springer, 2014, pp. 107–133. DOI: 10.1007/978-1-4471-6275-9_5 (cit. on pp. 19, 35).
- [299]A Vazquez, R Dobrin, D Sergi, J-P Eckmann, Zoltan N Oltvai, and A-L Barabási. “The topological relationship between the large-scale attributes and local interaction patterns of complex networks”. In: *Proceedings of the National Academy of Sciences* 101.52 (2004), pp. 17940–17945. DOI: 10.1073/pnas.0406024101 (cit. on p. 92).

- [300]Corinna Vehlow, Fabian Beck, and Daniel Weiskopf. “The State of the Art in Visualizing Group Structures in Graphs”. In: *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015, pp. 21–40. DOI: 10.2312/eurovisstar.20151110 (cit. on p. 93).
- [301]Corinna Vehlow, Fabian Beck, and Daniel Weiskopf. “Visualizing group structures in graphs: A survey”. In: *Computer Graphics Forum*. Vol. 36. Wiley Online Library. 2017, pp. 201–225. DOI: 10.1111/cgf.12872 (cit. on pp. 47, 51).
- [302]Rafael Veras and Christopher Collins. “Optimizing hierarchical visualizations with the minimum description length principle”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2016), pp. 631–640. DOI: 10.1109/TVCG.2016.2598591 (cit. on p. 29).
- [303]Saurabh Verma and Zhi-Li Zhang. “Hunt for the unique, stable, sparse and fast feature learning on graphs”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 88–98 (cit. on pp. 70, 76, 81, 83, 84, 86, 88, 128, 138, 139).
- [304]Eduardo Faccin Vernier, Rafael Garcia, Iron Prando da Silva, João L. D. Comba, and Alexandru C. Telea. “Quantitative Evaluation of Time-Dependent Multidimensional Projection Techniques”. In: *Computer Graphics Forum* (2020). DOI: 10.1111/cgf.13977 (cit. on p. 72).
- [305]Nguyen Xuan Vinh, Julien Epps, and James Bailey. “Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance”. In: *The Journal of Machine Learning Research* 11 (2010), pp. 2837–2854 (cit. on p. 63).
- [306]Ivan Viola and Tobias Isenberg. “Pondering the concept of abstraction in (illustrative) visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.9 (2017), pp. 2573–2588. DOI: 10.1109/TVCG.2017.2747545 (cit. on pp. 18, 19, 33).
- [307]VIS Paper Submission Keywords. Online <http://ieevis.org/year/2020/info/call-participation/paper-keywords> (cit. on pp. 22, 23).
- [308]Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P Gummadi. “On the evolution of user interaction in facebook”. In: *Proceedings of the 2nd ACM Workshop on Online Social Networks*. 2009, pp. 37–42. DOI: 10.1145/1592665.1592675 (cit. on p. 110).
- [309]Tatiana Von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J van Wijk, J-D Fekete, and Dieter W Fellner. “Visual analysis of large graphs: state-of-the-art and future research challenges”. In: *Computer Graphics Forum*. Vol. 30. 6. Wiley Online Library. 2011, pp. 1719–1749. DOI: 10.1111/j.1467-8659.2011.01898.x (cit. on pp. 7, 93, 116).
- [310]Junpeng Wang, Xiaotong Liu, Han-Wei Shen, and Guang Lin. “Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2016), pp. 81–90. DOI: 10.1109/TVCG.2011.195 (cit. on p. 31).

- [311]Yong Wang, Daniel Archambault, Hammad Haleem, Torsten Moeller, Yanhong Wu, and Huamin Qu. “Nonuniform Timeslicing of Dynamic Graphs Based on Visual Complexity”. In: *2019 IEEE Visualization Conference (VIS)*. IEEE. 2019, pp. 1–5. DOI: 10.1109/VISUAL.2019.8933748 (cit. on pp. 75, 122, 123, 125).
- [312]Ashley JW Ward, David JT Sumpter, Iain D Couzin, Paul JB Hart, and Jens Krause. “Quorum decision-making facilitates information transfer in fish shoals”. In: *Proceedings of the National Academy of Sciences* 105.19 (2008), pp. 6948–6953. DOI: 10.1073/pnas.0710344105 (cit. on p. 61).
- [313]Matthew O Ward, Georges Grinstein, and Daniel Keim. *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2010 (cit. on p. 105).
- [314]Colin Ware, Daniel Bolan, Ricky Miller, David H Rogers, and James P Ahrens. “Animated versus static views of steady flow patterns”. In: *Proceedings of the ACM Symposium on Applied Perception*. ACM. 2016, pp. 77–84. DOI: 10.1145/2931002.2931012 (cit. on p. 43).
- [315]Martin Wattenberg. “Visual exploration of multivariate graphs”. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM. 2006, pp. 811–819. DOI: 10.1145/1124772.1124891 (cit. on pp. 40, 42).
- [316]Duncan J Watts. “Networks, dynamics, and the small-world phenomenon”. In: *American Journal of Sociology* 105.2 (1999), pp. 493–527. DOI: 10.1086/210318 (cit. on p. 107).
- [317]Duncan J Watts and Steven H Strogatz. “Collective dynamics of ‘small-world’ networks”. In: *Nature* 393.6684 (1998), pp. 440–442. DOI: 10.1038/30918 (cit. on p. 85).
- [318]Peter A. H. Westley, Andrew M. Berdahl, Colin J. Torney, and Dora Biro. “Collective movement in ecology: from emerging technologies to conservation and management”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 373.1746 (2018), p. 20170004. DOI: 10.1098/rstb.2017.0004 (cit. on p. 46).
- [319]Nelson Wong, Sheelagh Carpendale, and Saul Greenberg. “Edgelens: An interactive method for managing edge congestion in graphs”. In: *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714)*. IEEE. 2003, pp. 51–58. DOI: 10.1109/INFVIS.2003.1249008 (cit. on p. 40).
- [320]Pak Chung Wong, Patrick Mackey, Kristin A Cook, Randall M Rohrer, Harlan Foote, and Mark A Whiting. “A multi-level middle-out cross-zooming approach for large graph analytics”. In: *2009 IEEE Symposium on Visual Analytics Science and Technology*. IEEE. 2009, pp. 147–154. DOI: 10.1109/VAST.2009.5333880 (cit. on p. 120).
- [321]Jonathan Woodring and Han-Wei Shen. “Multiscale Time Activity Data Exploration via Temporal Clustering Visualization Spreadsheet”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.1 (2009), pp. 123–137. DOI: 10.1109/TVCG.2008.69 (cit. on p. 35).

- [322] Liwenhan Xie, James O'Donnell, Benjamin Bach, and Jean-Daniel Fekete. "Interactive Time-Series of Measures for Exploring Dynamic Networks". In: *Proceedings of the International Conference on Advanced Visual Interfaces*. 2020, pp. 1–9. DOI: 10.1145/3399715.3399922 (cit. on pp. 96–98).
- [323] Jin Xu, Yubo Tao, Yuyu Yan, and Hai Lin. "Exploring Evolution of Dynamic Networks via Diachronic Node Embeddings". In: *IEEE Transactions on Visualization and Computer Graphics* (2018). DOI: 10.1109/TVCG.2018.2887230 (cit. on pp. 119, 123).
- [324] Ke Xu, Meng Xia, Xing Mu, Yun Wang, and Nan Cao. "Ensemblelens: Ensemble-based visual exploration of anomaly detection algorithms with multidimensional data". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2018), pp. 109–119. DOI: 10.1109/TVCG.2018.2864825 (cit. on pp. 22, 23).
- [325] Jing Yang, Matthew O Ward, and Elke A Rundensteiner. "Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets". In: *Computers & Graphics* 27.2 (2003), pp. 265–283. DOI: 10.1016/S0097-8493(02)00283-2 (cit. on p. 20).
- [326] Yalong Yang, Tim Dwyer, Sarah Goodwin, and Kim Marriott. "Many-to-many geographically -embedded flow visualisation: An evaluation". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2016), pp. 411–420. DOI: 10.1109/TVCG.2016.2598885 (cit. on p. 42).
- [327] Xin Yao, Lun Wu, Di Zhu, Yong Gao, and Yu Liu. "Visualizing spatial interaction characteristics with direction-based pattern maps". In: *Journal of Visualization* 22.3 (2019), pp. 555–569. DOI: 10.1007/s12650-018-00543-4 (cit. on pp. 47, 48).
- [328] Guan Yuan, Penghui Sun, Jie Zhao, Daxing Li, and Canwei Wang. "A review of moving object trajectory clustering algorithms". In: *Artificial Intelligence Review* 47.1 (2017), pp. 123–144. DOI: 10.1007/s10462-016-9477-7 (cit. on p. 61).
- [329] Wei Zeng, Chengqiao Lin, Juncong Lin, Jincheng Jiang, Jiazhi Xia, Cagatay Turkey, and Wei Chen. "Revisiting the Modifiable Areal Unit Problem in Deep Traffic Prediction with Visual Analytics". In: *IEEE Transactions on Visualization and Computer Graphics* (2020). DOI: 10.1109/TVCG.2020.3030410 (cit. on p. 35).
- [330] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. "Network representation learning: A survey". In: *IEEE transactions on Big Data* (2018). DOI: 10.1109/TBDATA.2018.2850013 (cit. on pp. 73, 74, 76, 119, 128).
- [331] Jiawei Zhang, Abish Malik, Benjamin Ahlbrand, Niklas Elmqvist, Ross Maciejewski, and David S Ebert. "Topogroups: Context-preserving visual illustration of multi-scale spatial aggregates". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 2940–2951. DOI: 10.1145/3025453.3025801 (cit. on pp. 28, 29, 35).
- [332] Jian Zhao, Fanny Chevalier, Emmanuel Pietriga, and Ravin Balakrishnan. "Exploratory analysis of time-series with chronolenses". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2422–2431. DOI: 10.1109/TVCG.2011.195 (cit. on p. 34).

- [333]Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. “Understanding mobility based on GPS data”. In: *Proceedings of the 10th International Conference on Ubiquitous Computing*. New York, NY, USA: Association for Computing Machinery, 2008, pp. 312–321. DOI: 10.1145/1409635.1409677 (cit. on p. 61).
- [334]Xi Zhu and Diansheng Guo. “Mapping Large Spatial Flow Data with Hierarchical Clustering”. In: *Transactions in GIS* 18.3 (), pp. 421–435. DOI: 10.1111/tgis.12100 (cit. on p. 43).
- [335]Michael Zinsmaier, Ulrik Brandes, Oliver Deussen, and Hendrik Strobel. “Interactive level-of-detail rendering of large graphs”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2486–2495. DOI: 10.1109/TVCG.2012.238 (cit. on p. 120).
- [336]Lingbo Zou and Stephen Brooks. “A dynamic approach for presenting local and global information in geospatial network visualizations”. In: *GeoInformatica* (2019), pp. 1–25. DOI: 10.1007/s10707-019-00350-5 (cit. on pp. 42, 45, 47, 48).

