Linköping Studies in Science and Technology
Dissertation No. 2557

Structure-Aware Representation Algorithms for Visual Analytics

Zeyang Huang

LINKÖPING UNIVERSITY

Linköping Studies in Science and Technology
Dissertations, No. 2557

Structure-Aware Representation Algorithms for Visual Analytics

Zeyang Huang

Linköping University
Department of Science and Technology
Division of Media and Information Technology
SE-581 83 Linköping, Sweden

Linköping 2026

Except where otherwise noted, this work is licensed under a Creative Commons Attribution 4.0 International License.
https://creativecommons.org/licenses/by/4.0/

Structure-Aware Representation Algorithms for Visual Analytics
© Zeyang Huang, 2026 (unless otherwise noted)

Division of Media and Information Technology
Department of Science and Technology
Campus Norrköping, Linköping University
SE–601 74 Norrköping, Sweden

ISBN 978–91–8118–759–5 (print)
ISBN 978–91–8118–760–1 (PDF)
https://doi.org/10.3384/9789181187601
ISSN 0345-7524
Published articles have been reprinted with permission from the respective copyright holder.

Cover: The cover image is an artistic interpretation of the thesis. Birds scatter in flight, each one an observation with many dimensions. Lines thread between them, sketching the relationships and structures the birds may form. Behind them, a blurred background holds the context, felt more than seen. The flock and its lines continue across the spine onto the back cover. The bird drawings are by Sihan and are used with permission.

Typeset using LaTeX
Printed by LiU-Tryck, Linköping, 2026

Edition 1:1

POPULÄRVETENSKAPLIG SAMMANFATTNING

Många frågor inom dataanalys handlar om relationer. I högdimensionella och relationella data är dessa relationer ofta implicita och alltför många för att kunna undersökas samtidigt. Algoritmer organiserar eller väljer därför ut relationer genom granngrafer, aligneringar och urval av delnätverk. Dessa beräknade representationer avgör vilka relationer som finns tillgängliga för visuell analys. Visuella representationer och interaktion avgör sedan hur analytiker kan undersöka dem. De beräknade relationerna kan dock vara missvisande eller ofullständiga. Brus kan göra att observationer som egentligen inte hör ihop betraktas som grannar, och en enda layout kan inte visa strukturer på alla skalor. Standardmetoder för dimensionsreduktion konstruerar vanligtvis en granngraf utifrån avstånden i ursprungliga data och använder sedan denna för att beräkna en enda layout. Denna avhandling presenterar strukturmedvetna representationsalgoritmer som tar hänsyn till den form som data följer och den skala på vilken grupper bildas. Den presenterar också interaktiva visualiseringstekniker som gör det möjligt för analytiker att undersöka den struktur som ett beräknat resultat bevarar. Avhandlingen bygger på fem artiklar och omfattar även två relaterade studier.

Två av artiklarna presenterar algoritmer för icke-linjär dimensionsreduktion: MAPLE, som behandlar brus, och SDMP, som behandlar skala. Algoritmen MAPLE betraktar de ursprungliga grannskapen som preliminära, lär sig ett nytt representationsrum med hjälp av ett självövervakat optimeringsmål och konstruerar om granngrafen innan en tvådimensionell layout optimeras. För bilddata och encellsdata ger metoden layouter där kluster och mindre kluster inom större grupper framträder tydligare än med UMAP. Algoritmen SDMP bygger upp varje koordinat i layouten med hjälp av spektrala komponenter från granngrafen, som läggs till stegvis från grova till fina variationer. De successiva stegen visar när kluster och trajektorier framträder och vilka komponenter som bidrar till dem. En tredje artikel, översiktsstudien VA + EMBEDDINGS, kategoriserar 122 vetenskapliga artiklar inom visuell analys och visar att få av dem förändrar hur deras embeddings konstrueras.

De återstående studierna utgår också från beräknade resultat som innehåller fler relationer än vad en enskild vy visar, men fokuserar på hur analytiker kan utforska dessa relationer genom visuella kodningar och interaktion. Studierna behandlar tvetydighet, urval av delar och detaljnivåer samt historiken bakom en utforskande analys. LINEALIGN bevarar flera möjliga matchningar mellan två tidsserier i en mjuk alignering och kopplar dessa matchningar till de ursprungliga kurvorna. Metodens vyer gör det möjligt att skilja mellan tidsförskjutning, tvetydiga korrespondenser, förändringar i förloppets hastighet och skillnader i variabelvärden. MATRIX SNAP&GO omordnar en adjacensmatris kring varje vald nod och håller den väg som hittills följts synlig. En relaterad studie kopplar samman vyer av flerpartssamtal, från sammanfattningar av samtalsavsnitt till enskilda yttranden. Den andra jämför delnätverk från sociala medier sida vid sida och registrerar varje interaktivt nätverksurval tillsammans med de val som ledde fram till det. MAPLE och SDMP utvärderades huvudsakligen med kvantitativa mått och fallstudier, medan visualiseringsteknikerna utvärderades genom fallstudier och återkoppling från experter.

Sammantaget visar studierna att beräknade och visuella representationer utgör två designsteg som är beroende av varandra. Ett gränssnitt kan endast synliggöra, omordna eller filtrera relationer som ett beräknat resultat bevarar. Om relationer saknas måste själva beräkningen ändras. Detta samband visar varför det är viktigt att ompröva hur relationer beräknas och presenteras. Nya analytiska möjligheter uppstår när algoritmer kan lära sig designval som annars är förutbestämda, eller när analytiker kan granska och ändra dessa val.

ABSTRACT

Many questions in data analysis are questions about relationships. In high-dimensional and relational data, these relationships are often implicit and too numerous to inspect at once, so algorithms organize or select them through neighbor graphs, alignments, and network samples. These computational representations determine which relationships remain available to visual analysis. Visual representation and interaction determine how analysts can inspect them. However, the computed relationships can be misleading or incomplete. Noise can make unrelated observations neighbors, and one layout cannot show structure at every scale. Standard dimensionality reduction methods build the neighbor graph once from the input distances and compute a single layout from it. This thesis presents structure-aware representation algorithms that take into account the shape along which the data lie and the scale at which groups form. The thesis also presents interactive visualization techniques that let analysts inspect the structure that a computational result preserves. It is based on five papers and draws on two related studies.

Two of the papers present algorithms for nonlinear dimensionality reduction, MAPLE for noise and SDMP for scale. MAPLE treats initial neighborhoods as tentative, learns a space with a self-supervised objective, and rebuilds the neighbor graph before optimizing a two-dimensional layout. On image and single-cell data, its layouts separate clusters and subclusters more clearly than UMAP. SDMP builds each layout coordinate from spectral components of the neighbor graph and adds them in stages from broad to fine variation. The progressive stages show when clusters and trajectories appear and which components contribute to them. A third paper, the VA + EMBEDDINGS survey, categorizes 122 visual analytics papers and finds that few change how their embeddings are constructed.

The remaining studies also start from a computational result that captures more relationships than one view shows, but focus on how analysts access these relationships through visual encodings and interaction. They address ambiguity, the selection of parts and levels of detail, and the history of an exploration. LINEALIGN keeps competing matches between two time series in a soft alignment and links these matches to the original curves. Its views distinguish temporal displacement, correspondence ambiguity, pace change, and differences in feature values. MATRIX SNAP&GO reorders an adjacency matrix around each selected node and keeps the visited path visible. One related study links views of multiparty conversations from episode summaries to individual utterances. The other compares social media subnetworks side by side and records each interactive network sample with the choices that produced it. While MAPLE and SDMP were evaluated mainly with quantitative measures and use cases, those visualization techniques were evaluated through use cases and expert feedback.

Across the studies, computational and visual representation form two dependent design stages. An interface can expose, rearrange, or filter only relationships that a computational result preserves, whereas missing relationships require changing the computation. This dependency highlights the need to reconsider how relationships are computed and presented. New analytical possibilities emerge when design choices that are typically fixed in advance can be learned by algorithms or revised by analysts.

Acknowledgments

I am truly grateful to my principal supervisor, Niklas Rönnberg. Thank you for your invaluable advice and support during the final stage of my doctoral journey, and for always being so thoughtful, generous, and understanding. Your kindness and encouragement have meant so much to me. I am also immensely grateful to my supervisor, Katerina Vrotsou. Thank you for your sharp and insightful feedback. I truly appreciate the time and care you have devoted to helping me, as well as your kindness and support throughout this process.

I would like to show my deepest gratitude to my supervisor, Takanori Fujiwara. I could not have come this far in my doctoral studies without your advice, patience, and encouragement. You are such a brilliant researcher and an incredibly kind person, and I've always felt very fortunate to have met you.

I am grateful to my thesis committee members, Remco Chang, Jaakko Peltonen, Marcela Davila, Jessica Lindblom, and Vangelis Angelakis. Thank you for your time, consideration, and feedback. I would like to thank Andreas Kerren for introducing the field of visualization and for the opportunity to pursue doctoral studies, and Kostiantyn Kucher for the guidance during the first three years of studies. My thanks also go to Jonas Unger, Anna Fredriksson, Jonas Löwgren, Camilla Forsell, Denise Wahlström, Lizzie Nichols, and the administrative teams at Linköping University for your generous help with so many practical matters. I am also very grateful to all the colleagues in the Division of Media and Information Technology. You are amazing researchers and educators, and I have learned so much from you and enjoyed our conversations.

I have greatly appreciated working with all my collaborators, and especially Angelos Chatzimparmpas and Thomas Höllt; thank you for your thoughtful insights and stimulating discussions. I am also very grateful to the ELLIIT Gender and Diversity Group for the funding opportunity, and to all the wonderful people I met and spoke with in Utrecht, Ghent, Eindhoven, and Delft. Such an inspiring research visit.

I would also like to thank Professor Michael Lamar for teaching me how to do research during my undergraduate studies. Seven years ago, I could not have imagined that your explanations in information theory would mark the beginning of the rewarding journey that eventually led to this thesis. Your passion for mathematics has continued to inspire me along the way.

To my friends in Norrköping and Linköping: thank you for the lunches, dinners, burgers, tom yum-soppa, fika, board games, and everything else that made everyday life so enjoyable. I will not name you, but if you happen to read this, please know that I appreciate you. Just drop me a message–the next lunch is on me!

Finally, I want to express my deepest thanks to my family and my friends far to the east and west. Despite the distances between us, your love, encouragement, and support have always been with me. Thank you for simply being there and giving me strength. This journey, and everything that brought me here, would not have been possible without you.

This thesis represents only one beginning stage of my research journey. Several further projects in dimensionality reduction and visualization are underway. **Please stay tuned!**

- zeyang
Norrköping, Oct 26, 2026
zeyang.huang@outlook.com

List of Publications

This thesis is based on Papers I–V. Related Studies A and B provide additional work closely connected to the thesis. Brackets are drawn to indicate groups of studies addressing closely related problems.

Representation algorithms

Dimensionality reduction

Paper I Zeyang Huang, Takanori Fujiwara, Angelos Chatzimparmpas, Wandrille Duchemin, and Andreas Kerren. "MAPLE: Self-Supervised Learning-Enhanced Nonlinear Dimensionality Reduction for Visual Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 32.7 (2026), pp. 7131–7146. DOI: 10.1109/TVCG.2026.3694000

Paper II Zeyang Huang, Angelos Chatzimparmpas, Thomas Höllt, and Takanori Fujiwara. "A Spectral Decomposition Framework for Multiscale Nonlinear Dimensionality Reduction". In: *IEEE Transactions on Visualization and Computer Graphics (Conditionally Accepted)* (2026). DOI: 10.48550/arXiv.2604.02535

Sequence alignment

Paper III Zeyang Huang, Takanori Fujiwara, Niklas Rönnberg, and Katerina Vrotsou. "LineAlign: Comparing Time Series in Line Charts through Optimal Transport-Based Soft Correspondence". Manuscript to be submitted to Computer Graphics Forum. 2026

Representation foundations

Paper IV Zeyang Huang, Daniel Witschard, Kostiantyn Kucher, and Andreas Kerren. "VA + Embeddings STAR: A State-of-the-Art Report on the Use of Embeddings in Visual Analytics". In: *Computer Graphics Forum* 42.3 (2023), pp. 539–571. DOI: 10.1111/cgf.14859

Interactive visual analysis

Matrix interaction

Paper V Zeyang Huang, Daniel Archambault, Rita Borgo, and Andreas Kerren. "Matrix Snap&Go: Visualization of Paths on Matrices". In: *EuroVis 2024 – Short Papers*. The Eurographics Association, 2024. DOI: 10.2312/evs.20241058

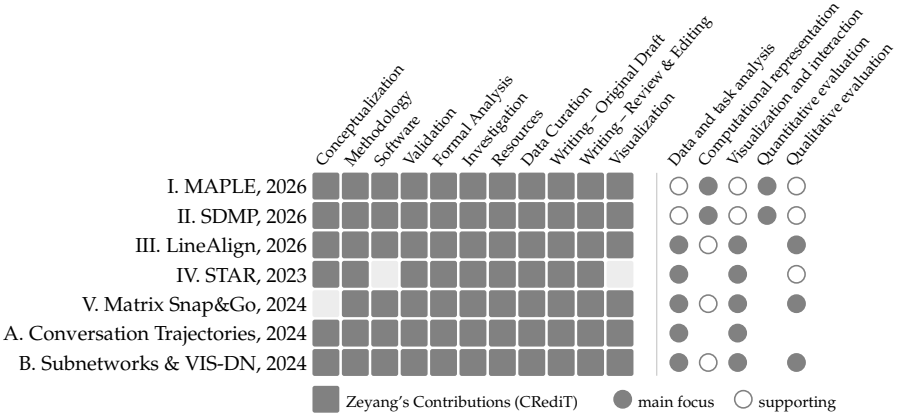
Exploratory network analysis

Related Study A Zeyang Huang, Kostiantyn Kucher, and Andreas Kerren. "Towards a Visual Analytics System for Emotion Trajectories in Multiparty Conversations". In: *EuroVis 2024 – Posters*. The Eurographics Association, 2024. DOI: 10.2312/evp.20241092

Related Study B Zeyang Huang, Kostiantyn Kucher, and Andreas Kerren. "Towards an Exploratory Visual Analytics System for Multivariate Subnetworks in Social Media Analysis". In: *Poster Abstracts, IEEE Visualization and Visual Analytics (VIS '22)*. DiVA: diva2:1707880. 2022

Zeyang Huang, Kostiantyn Kucher, and Andreas Kerren. "VIS-DN: Visual Analytics for Interactive Sampling on Dynamic Multivariate Networks". *Manuscript, 15 pages*. Full text included in this thesis; builds on the poster of Related Study B. 2024

The figure below provides an overview of Zeyang's contributions to each work and their respective research focus. The left matrix summarizes my contributions according to the CRediT taxonomy. The right matrix indicates the main and supporting focus of each work across five aspects of visual analytics.



Contents

Abstract v

Acknowledgments vii

List of Publications ix

Contents xi

List of Figures xiii

List of Tables xv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Structure in Computational Representations . . . 4
- 1.3 Research Questions . . . 5
- 1.4 Included Papers and Related Studies . . . 8

2 Background and Related Work 11
- 2.1 Representation in Visual Analytics . . . 11
- 2.2 Capturing Structure with Neighbor Graphs . . . 12
- 2.3 Visualizing Structure for Visual Analysis . . . 18
- 2.4 Goals for Visual Encoding and Interaction . . . 22
- 2.5 Summary . . . 23

3 Methodology and Research Process 25
- 3.1 Two Well-Known Models of Visualization Design . . . 27
- 3.2 The Research Process . . . 29

4 Summary of Included Papers and Related Studies 35
- 4.1 Paper I: MAPLE . . . 35
- 4.2 Paper II: SDMP . . . 36
- 4.3 Paper III: LINEALIGN . . . 37
- 4.4 Paper IV: VA + EMBEDDINGS STAR . . . 37

4.5 Paper V: Matrix Snap&Go 38
4.6 Related Study A: Conversation Trajectories 38
4.7 Related Study B: Multivariate Subnetworks and VIS-DN 39

5 Conceptual Framework **41**
5.1 Motivation from Analysis Tasks and Types of Relationships . . 41
5.2 The Overarching Pipeline of This Thesis 44
5.3 Comparing the Analysis Pipelines of the Studies 45

6 Synthesis **49**
6.1 Comparing MAPLE and SDMP (Papers I and II) 49
6.2 General Design Considerations 54
6.3 Future Work 56

7 Conclusions **59**

Bibliography **63**

Publications **77**
Paper I: "MAPLE: Self-Supervised Learning-Enhanced Nonlinear Dimensionality Reduction for Visual Analysis" 77
Paper II: "A Spectral Decomposition Framework for Multiscale Nonlinear Dimensionality Reduction" 95
Paper III: "LineAlign: Comparing Time Series in Line Charts through Optimal Transport-Based Soft Correspondence" 109
Paper IV: "VA + Embeddings STAR: A State-of-the-Art Report on the Use of Embeddings in Visual Analytics" 123
Paper V: "Matrix Snap&Go: Visualization of Paths on Matrices" . . . 159
Study A: "Towards a Visual Analytics System for Emotion Trajectories in Multiparty Conversations" 167
Study B: "Towards an Exploratory Visual Analytics System for Multivariate Subnetworks in Social Media Analysis" 173
Study B: "VIS-DN: Visual Analytics for Interactive Sampling on Dynamic Multivariate Networks" 175

List of Figures

1.1 Sorting a table and computing a layout from a neighbor graph . . . 3
1.2 Two difficulties in showing structure with a neighbor graph 6

2.1 Four spectral components of a small neighbor graph 16

3.1 Two models of visualization design and the research process . . . 27

5.1 From data records, to candidate pairs, and to the pairs a standard method keeps 43
5.2 A four-stage pipeline and the parts this thesis contributes to 44
5.3 How four studies extend the computed result for one task 46

6.1 Changes that Papers I and II make to a neighbor-embedding pipeline 50
6.2 Interactive visualization techniques for DC1 to DC3 54

List of Tables

3.1 Emphasis of the research process in each study 30

5.1 Tasks in the six studies and terms from task vocabularies 42
5.2 The pipeline of each study 45

1 Introduction

1.1 Motivation

Many questions in data analysis are questions about relationships within a structure, as individual observations can often be difficult to interpret in isolation. For example, the gene activity of a cell can be understood by comparing it with that of other cells. A period in a time series can be interpreted in relation to what came before and after it. A link in a network can be examined as part of a larger path or subnetwork. In each case, relationships provide context needed for interpretation, and interactive visualization gives analysts ways to inspect those relationships.

While visualization works well when a relationship involves only a few attributes, the structure in high-dimensional and relational data cannot simply be drawn straight onto a screen. A table with thousands of columns, a set of evolving sequences, or an intricate web of connections cannot be plotted directly in a form that is still meaningful to an analyst. Algorithms first need to find the patterns hidden in such data and turn them into visual forms. For instance, dimensionality reduction turns abstract high-dimensional similarity into physical distance on the screen; sequence alignment matches corresponding moments across time; and graph layouts untangle network connections into readable arrangements. These computed results serve as an intermediate layer, a *computational representation*, that directly decides which relationships analysts can actually see and explore.

In this sense, the algorithm's choice of how to construct the representation is not just preparation or preprocessing of raw data. It is also a visualization design choice. What an interface can reveal is bounded by what the underlying representation captures, and the value of those relationships depends on how easily analysts can inspect them.

This thesis studies this design problem across two kinds of data: high-dimensional data, which record many attributes for each data item, and relational data, which record how data items are ordered or linked, as in sequences and networks. Although these data types differ in what they record, analysts face a shared challenge: the collection of potential relationships is far too vast and complex to explore all at once. Making sense of them becomes, in part, a problem of representation, that is, transforming raw, implicit relationships into forms that support analysis.

Throughout this thesis, a *relationship* specifies how data elements are associated with one another, such as similarity between high-dimensional data items, correspondence between moments in different sequences, or connection and reachability between nodes in a network. The term *representation* refers to a computational form or space in which such relationships are organized or encoded, such as an embedding, a neighbor graph, or a correspondence matrix. A *visualization* presents data graphically to support analytical tasks. Examples include scatterplots, matrix layouts, and coordinated views that make selected aspects of a computational representation available for inspection. *Visual encoding* refers more specifically to the use of position, color, size, and other visual channels within a visualization.

From data transformation to data representation

Nearly three decades ago, the classic visualization reference model [9] split visualization into two distinct steps: transforming data and mapping it to visual encodings. This separation is intuitive. The first step reshapes the data itself via sorting, filtering, or aggregating values. The second step decides how those values look on screen by assigning them to visual channels like position, color, and shape.

This classic pipeline works well for simple tables. But as data becomes more complex, the single term *data transformation* bundles together two fundamentally different kinds of operations (Figure 1.1). On the one hand, routine operations like sorting, filtering, or aggregating simply change which items are shown or in what order, leaving the recorded values unchanged. On the other hand, complex data requires transformations that construct an entirely new *computational representation*: a new space designed to bring hidden patterns to the surface. Constructing a representation requires an algorithm to determine which relationships to preserve and how to encode

them. For instance, an embedding space translates high-dimensional similarity into spatial proximity; an alignment matches corresponding moments across sequences; and a graph layout reveals connectivity patterns through spatial arrangement. In each case, the algorithm defines which relationships survive to be explored on screen.

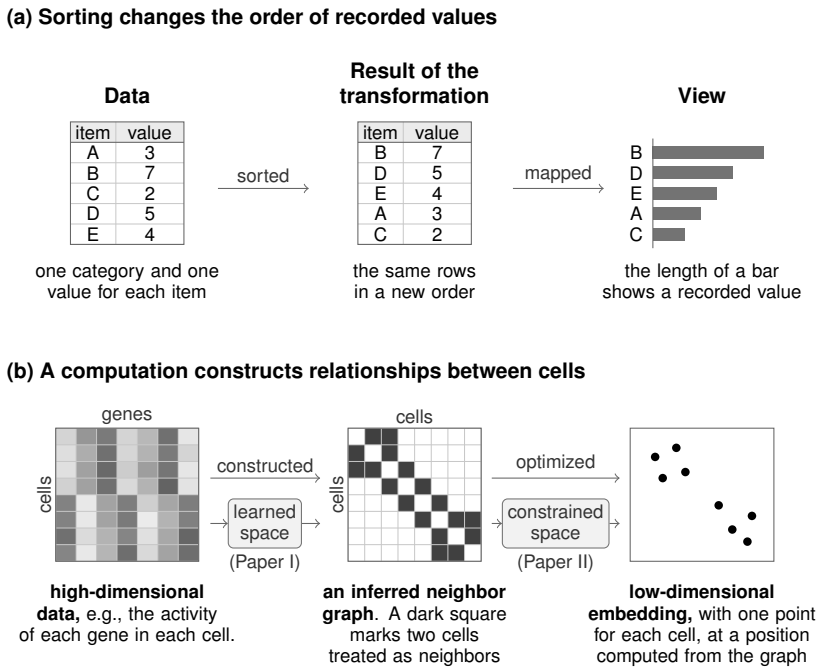


Figure 1.1: Two kinds of operation that the term *data transformation* covers. In (a), sorting changes the order of the rows, and the bar chart shows values that the table already records. In (b), the table records the activity of each gene in each cell. A computation constructs a neighbor graph, drawn here as a matrix. The positions in the scatterplot are computed from this graph by optimization. The arrows in (a) take each step directly. The arrows in (b) add a layer in the middle of each step. Paper I constructs the neighbor graph in a learned space, and Paper II optimizes the downstream layout positions in a constrained space, which it widens from broad to fine structure.

These computational representations in Figure 1.1(b) make implicit relationships available for visual exploration. Many relationships are not stored as explicit values in a table. A single-cell dataset, for example, lists thousands of gene measurements per cell, but it does not record which cells are similar to one another. An analyst cannot spot those similarities across thousands of dimensions by eye, and comparing every pair of cells manually is impossible. The relationships are hidden in the data and need to be computed and organized before any visual tool can show them.

Constructing such representations is also the subject of representation learning (RL) in machine learning, and two studies in this thesis use or examine its methods directly. Similar to RL, the representation algorithms mentioned in this thesis construct representations to compress complex data and capture its main patterns. However, their design goals diverge because their intended consumers are different. In representation learning, a learned representation usually feeds into another computational component, such as a classifier, a decoder, or a retrieval model. Its quality is judged by how well that component performs, for example, by prediction accuracy. In visual analysis, by contrast, the computational representation is handed over to human interpretation. A classifier can effortlessly utilize all fifty dimensions of an embedding, but a person looking at a screen can only track two to three spatial axes and perhaps an assigned color scheme. As a result, a representation that works well for a machine-learning model may still be difficult or impossible for an analyst to interpret, compare, or trust. Paper I meets this difference directly. It adapts its learning objective from self-supervised learning, but its result is a layout that a person reads, so the paper argues for evaluation measures closer to perception and analysis tasks.

When a computational representation reaches the visual stage, it directly defines what the visual interface can reveal. For example, a 2D embedding gives each data point a position, but nearby points on screen may not reflect true high-dimensional neighbors. Ensuring that relationships are reliable, interpretable, and relevant is partly a design choice within the algorithm itself. This motivates the need to develop "explanation-by-design" representations built specifically for visual analysis. The dependency also runs in the other direction. The specific tasks an analyst needs to perform, and the visual channels available to human eyes, can determine what the algorithm is designed to preserve.

1.2 Structure in Computational Representations

To describe local relationships and the underlying structure in high-dimensional data, a graph is a good proxy. Distances across a whole dataset are hard to interpret in many dimensions, but local similarity, which items lie close to which, can be measured. Graph connectivity builds an abstraction of high-dimensional relationships from this local information. It describes similarity as neighborhoods and larger structure as longer paths between data items. A dimensionality reduction method such as t-SNE [10] or UMAP [11] then places observations in a low-dimensional space, where spatial proximity can reveal the neighborhood structure captured by the graph.

Each edge of the graph captures one relationship, the similarity of two observations. *Structure* is the pattern that many such relationships form together.

For example, in single-cell data, cells with similar gene expression may form neighborhoods, while sequences of connected neighborhoods may reveal developmental trajectories. A neighbor graph can capture both local similarities and broader relationships among groups of cells.

However, there are issues with such graph representations. As the number of features grows, pairwise distances tend to become similar to each other, and nearest-neighbor sets become sensitive to noise [12]. For example, in single-cell data, gene expression may change gradually as cells develop, forming a curved trajectory in the high-dimensional feature space. Two cells can be close in Euclidean distance yet far apart along the developmental trajectory. A neighbor graph based only on pairwise distances may connect such cells, creating shortcuts between distinct parts of the trajectory. Thus, when structure is hard to find in the original space, the neighborhoods themselves need to be constructed better.

Large datasets also hold structure at many scales at once. A two-dimensional layout generally cannot preserve all of these equally well [13]. A single embedding does not show at which level of detail a visible group forms. With limited screen space, one view cannot capture every scale, so the scales need to be revealed separately.

A representation algorithm is *structure-aware* when it estimates a particular structure of the data and builds its result so that the structure can be read from it. This thesis considers two such structures, the shape along which the data lie and the scale at which groups form.

1.3 Research Questions

The aim of this thesis is to support exploratory visual analysis of relationships in high-dimensional and relational data. It pursues this aim by designing structure-aware representation algorithms that capture the structure analysts need, and visual interfaces that give analysts access to that structure for exploration.

Three research questions follow from this aim. The first two concern computational representation, while the third involves how analysts can explore these representations through interactive visual interfaces.

RQ1 *Refining structure.* How can the computation of relationships be refined to reduce noise and reveal structure in high-dimensional data?

RQ2 *Revealing multiscale structure.* How can the construction of a layout be adapted to reveal structure at multiple scales?

Both questions concern the computational representation, the intermediate layer that determines which relationships analysts can see. The thesis studies how these relationships can be computed to suppress noise and preserve meaningful structure (RQ1, Figure 1.2a), and how the resulting structure can be revealed at different scales (RQ2, Figure 1.2b).

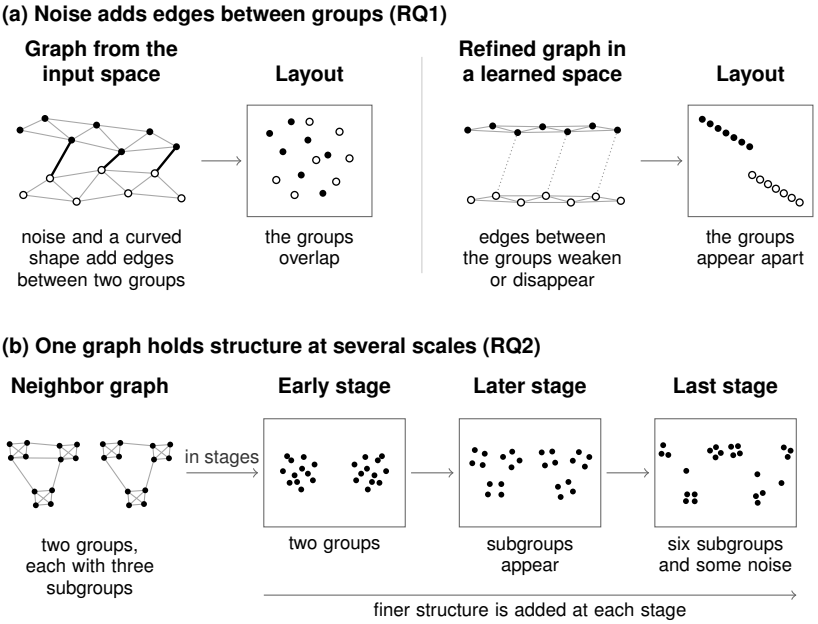


Figure 1.2: Two difficulties in capturing and showing structure with a neighbor graph. (a) Observations from two groups are drawn as filled and open points. In the input space, noise and a curved shape create edges between the groups (thick lines), and the layout mixes the groups. Refining the neighborhoods weakens or removes these edges (dotted lines), and the layout shows the groups apart. (b) A neighbor graph holds two groups, each with three subgroups. A sequence of layouts adds finer structure at each stage. The early stage shows the two groups, and later stages separate the subgroups. The last stage also shows more noise, because noise and fine structure both vary quickly between neighbors.

Beyond constructing computational representations, the thesis also considers how analysts can explore the resulting structure through visualization and interaction. To this end, it investigates a third research question.

RQ3 *Exploring structure through interactive visual analysis.* How can visualization and interaction support analysts in exploring structures in high-dimensional and relational data, particularly when faced with noise and scale?

This thesis answers RQ3 in two steps. First, it characterizes existing design approaches by surveying 122 visual analytics papers that use embeddings [4]. The survey categorizes these approaches by application domain, input data, computational method, analytical task, visualization, interaction, and evaluation. Few of them change how embeddings are constructed, so the structure an analyst can see is usually fixed before visual design begins.

Then, the thesis takes up three design challenges (DCs) within this space.

The first challenge (DC1) concerns noise, as discussed in RQ1. Structure computed from noisy data depends on how variation is treated and how noise is suppressed, so different methods and settings may produce different, similarly plausible results. Thus, downstream visual analysis should avoid presenting a single result as the unique answer and instead allow analysts to inspect the ambiguity among plausible results.

DC1 Identify the design space for visualizing ambiguous computational results, which contain several similarly plausible answers.

This thesis examines DC1 through a visual analytics approach for exploring alignment ambiguity between time series, allowing analysts to inspect multiple plausible correspondences between the original curves.

The second challenge (DC2) follows from scale, as mentioned in RQ2. Large datasets may contain structures at multiple scales, making it difficult to inspect all relevant relationships in a single view. Analysts therefore need interactive ways to access different parts of the data and levels of detail, depending on their analytical goals.

DC2 Support analysts in selecting and exploring relevant structures in large relational datasets through interactive visual analysis.

This thesis examines DC2 through visual analytics approaches for sampling and comparing parts of a large network and for viewing multiparty conversations at several levels of detail, allowing analysts to choose which part of the data to inspect and how closely.

The third challenge (DC3) follows from DC2. Since each scale and each sample shows different structure, an analyst exploring several needs to know how each result was reached. Following a path through a network raises the same need. An adjacency matrix records every link of a path but shows only direct links.

DC3 Track analytical provenance and compare the results of different exploration steps.

This thesis examines DC3 through a matrix reordering interaction technique that rearranges rows and columns around a selected node, and through a provenance graph of earlier network samples in a visual analytics system, allowing analysts to follow a path one link at a time and to compare the results of different sampling steps.

In summary, the three research questions organize the rest of this thesis. RQ1 and RQ2 are addressed through representation algorithms that refine structure and reveal it across scales. RQ3 is addressed through a survey of existing visual analytics approaches and through visualization prototypes and interaction techniques that investigate three design challenges. DC1 concerns ambiguity arising from noise, DC2 concerns exploring structure at different scales, and DC3 concerns analytical provenance during exploration. The thesis addresses these design challenges through example prototypes. They illustrate each challenge and do not form an exhaustive taxonomy of the design space.

Scope. This thesis focuses on high-dimensional data and on two forms of relational data, sequences and networks, because each highlights a distinct representation challenge: high-dimensional data requires preserving similarity across many dimensions, sequential data requires establishing correspondences across time, and network data requires following paths and selecting parts of a large network. These three cases serve as testbeds for examining design questions and do not form an exhaustive taxonomy of data forms. The included studies do not attempt to cover every relationship or characterize every structure that can arise within them.

In addition, this thesis is grounded in designing and constructing algorithms and visual interfaces, evaluated through established computational and qualitative practices. The included studies assess their contributions through quantitative metrics of relationship preservation, qualitative comparisons, real-world analysis scenarios, and domain expert feedback. For the datasets and cases examined, this evidence shows that the algorithms preserve the targeted relationships and that the proposed interfaces provide functional ways to inspect them. However, this work does not measure how accurately, quickly, or reliably analysts in general perform these tasks compared to alternative tools. Establishing such empirical claims would require formal, controlled user studies, which are outside the scope of this thesis.

1.4 Included Papers and Related Studies

This thesis is based on five papers, Papers I to V, and draws on two related studies, Studies A and B. Study A is a poster on emotion trajectories in multiparty conversations. Study B consists of two publications on the same net-

work of online communities, a poster on comparing subnetworks and the manuscript VIS-DN on interactive sampling. Together, they represent three methodological groups:

- **Algorithmic design of representations (Papers I and II):** These studies focus on the computational stage. Paper I refines structure by learning the space in which relationships are computed (RQ1), and Paper II reveals structure from broad to fine scales (RQ2).
- **Visual encoding and interaction techniques (Papers III and V, and Studies A and B):** These studies focus on the interface stage (RQ3). Taking underlying representations as given, they develop visual encodings and interaction techniques that help analysts read alignments with several plausible matches (Paper III, DC1), choose which part of a large network or dialogue collection to show and at which level of detail (Studies A and B, DC2), and keep track of paths and earlier selections during exploration (Paper V and Study B, DC3).
- **Design space (Paper IV):** Paper IV provides a state-of-the-art survey reviewing how visual analytics papers use embeddings, and finds that many of them give limited detail on how their embeddings were chosen. It characterizes the design space of RQ3 for one kind of computational representation, embeddings.

2 Background and Related Work

Visual analysis of relationships uses computational methods to construct neighbor graphs, embeddings, and alignments, along with visual techniques that help analysts work with these results. This chapter places both within established models of visual analytics. It then reviews the methods and techniques on which the studies of this thesis build, followed by goals for visual encoding and interaction.

2.1 Representation in Visual Analytics

The reference model of Card, Mackinlay, and Shneiderman describes visualization as a sequence of transformations from raw data to data tables, visual structures, and views [9]. The model separates the organization of data from its mapping to graphical form.

Visual analytics models add automated analysis and the analyst to this account. Sacha et al. distinguish the construction of models from data and the construction of visualizations from data, and they describe how the analyst interacts with both [14]. In these models, a display is produced from an intermediate organization of the data. This thesis calls such an organization a computational representation when it records relationships among data items.

A neighbor graph captures which observations a method treats as related. An embedding captures selected similarities as distances between points. An alignment indicates which time points in two series correspond. An adjacency matrix records which nodes in a network are directly linked. A visual mapping of one of these results can show only what the result records, unless the interface returns to the original data.

What a representation needs to preserve depends on the analytical task. Brehmer and Munzner describe a task by why it is performed, how it is carried out, and what data it concerns [15]. A representation suited to identifying local groups may be poorly suited to comparing distances between those groups.

2.2 Capturing Structure with Neighbor Graphs

Dimensionality reduction (DR) is the computational setting of Papers I and II. A DR method maps observations described by many features to positions in two or three dimensions [16]. The positions are meant to keep selected relationships among the observations. A low-dimensional result cannot keep all of them, so each method decides which relationships to favor.

This section first describes neighbor graphs and neighbor embeddings. It then reviews work on refining the structure that a neighbor graph captures when the data are noisy, and on revealing that structure at several scales. It ends with methods that explain how an embedding was formed and with the measures used to assess what a DR result preserves.

2.2.1 Neighbor graphs and neighbor embeddings

Linear methods such as principal component analysis express each output coordinate as a weighted sum of input features, which makes the contribution of each feature explicit [17]. A single linear map cannot faithfully unfold a curved data manifold. Many nonlinear methods address this limitation with a neighbor graph, in which each observation is connected to nearby observations. Isomap uses shortest paths in such a graph to approximate distances along the data [18]. Laplacian Eigenmaps derives coordinates from eigenvectors of the graph Laplacian [19]. t-SNE and UMAP optimize point positions so that neighbors in the graph remain close in the layout [10, 11]. These optimization-based methods are known as neighbor embeddings.

UMAP is widely used in biology [20], and both computational studies in this thesis build on it. UMAP works in two phases: graph construction and graph layout [11].

Graph construction. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times d}$ denote the input data, with N observations and d features. UMAP first finds the k nearest neighbors of each observation under a distance function $\mathrm{dist}(\cdot, \cdot)$, by default the Euclidean distance. It then assigns a weight to each edge from $\mathbf{x}_i$ to a neighbor $\mathbf{x}_j$,

$$w_{ij} = \exp\left(-\frac{\max\bigl(0,\, \mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j) - \rho_i\bigr)}{\sigma_i} \right), \tag{2.1}$$

where, under UMAP's default local-connectivity setting, ρ_i is the distance from $\mathbf{x}_i$ to its nearest neighbor at positive distance. The scale σ_i is chosen separately for each observation so that neighborhoods with different distance ranges receive comparable total connection strength. A missing directed edge has weight zero. The directed weights are then made symmetric,

$$w_{ij} \leftarrow w_{ij} + w_{ji} - w_{ij}\, w_{ji}. \tag{2.2}$$

The result is a weighted graph $\mathbf{W}$ with entries w_{ij}. It records which observations UMAP treats as related and how strongly.

Graph layout. In the second phase, UMAP places the observations at positions $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top \in \mathbb{R}^{N \times m}$, usually with $m = 2$. The similarity between two positions is modeled as

$$v_{ij} = \left(1 + a\, \|\mathbf{y}_i - \mathbf{y}_j\|^{2b}\right)^{-1}, \tag{2.3}$$

where a and b are derived from `min_dist` and `spread`, which control the shape of the low-dimensional similarity curve. The positions are optimized to minimize the cross-entropy between the graph weights and the layout similarities,

$$\mathcal{L}(\mathbf{Y}) = \sum_{i \neq j} \left[w_{ij} \log \frac{w_{ij}}{v_{ij}} + (1 - w_{ij}) \log \frac{1 - w_{ij}}{1 - v_{ij}} \right]. \tag{2.4}$$

The first term pulls connected observations together. The second term pushes apart observations that are weakly connected or not connected. UMAP initializes the layout with a spectral embedding of the graph and optimizes it by stochastic gradient descent with negative sampling.

Böhm et al. show that t-SNE, UMAP, and related methods form one family that differs mainly in the balance between attraction and repulsion [21]. Stronger attraction produces continuous structures. Stronger repulsion produces compact, separated clusters. Damrich and Hamprecht show that negative sampling weakens repulsion considerably relative to the stated objective [22]. The layout then depends strongly on which edges the graph contains.

Two properties of this formulation matter for the thesis. The attractive term in Equation 2.4 acts on relationships recorded as graph edges. Repulsion and the optimization process also shape the final arrangement, so a visible pattern cannot be attributed to the graph alone. The final coordinates do not show how these factors produced the pattern. The next two subsections examine the graph (Section 2.2.2) and the scales of structure that a layout shows (Section 2.2.3).

2.2.2 Refining structure in noisy data

Standard graph construction relies on distances in the input space, and these distances can be unreliable when the data are noisy or lie on curved surfaces (Section 1.2). Equation 2.1 rescales distances locally, but UMAP still begins with a fixed number of neighbors found under the chosen input-space distance [11]. In data with varying density, these neighborhoods can cover regions of very different size.

Several methods modify the layout objective or help choose its parameters. densMAP adds a term that preserves local density [23]. scDEED compares neighborhoods before and after layout to identify unreliable points and guide choices such as k [24]. These methods can change the resulting graph or layout, but they still define neighbors through the chosen input-space distance.

Self-supervised learning offers a different way to define similarity. A model is trained on signals derived from the data itself, without labels [25]. Multi-view methods create augmented versions of each item, called views in this literature, and train an encoder to represent views of the same item consistently.

Maximum manifold capacity representations (MMCR) belong to this family [26]. They draw on manifold capacity theory [27]. In that theory, each object corresponds to a manifold, the set of representations of the object under variations such as changes of view. Capacity measures how many such manifolds a linear classifier can separate under random labels, relative to the number of dimensions. It grows as each manifold becomes smaller and occupies fewer dimensions. Capacity is hard to optimize directly, so MMCR uses the nuclear norm, the sum of a matrix's singular values, as a tractable surrogate. A small nuclear norm for an item's views means they are compressed into a few directions. A large nuclear norm for the item-centroid matrix means the centroids spread across many directions.

Several DR methods use self-supervised learning to improve similarity estimates. t-SimCNE combines contrastive learning with a t-SNE-like objective, but its augmentations are specific to images [28]. TLDR uses each observa-

tion's nearest neighbor as a training signal and targets compression for later processing, not visualization [29]. Paper I, MAPLE, adapts MMCR to neighbor graphs. It treats each observation and its k nearest neighbors as one manifold, minimizes the nuclear norm of each neighborhood, and maximizes the nuclear norm of the matrix of neighborhood centroids. The neighbor graph is rebuilt in the learned space and laid out with the UMAP objective [1]. In standard UMAP, the neighbor graph is fixed before optimization. In MAPLE, the neighbor graph is itself a learned result of the method.

2.2.3 Revealing structure at several scales

The tension between broad and fine structure in one layout (Section 1.2) is known as the global–local trade-off [30]. In neighbor embeddings, the balance between attraction and repulsion (Section 2.2.1) sets where a layout falls on this trade-off. The global arrangement of clusters also depends strongly on initialization [31]. Visible separation between clusters in a t-SNE or UMAP layout may come partly from the optimization and not only from the data.

Methods can move the compromise by changing the objective, the sampling of point pairs, or the parameters. PaCMAP, for example, balances near, mid-range, and far pairs [30]. Each parameter setting of such a method produces a separately optimized layout. Hierarchical methods such as HSNE define scale through sampling [32]. Coarse levels embed a small set of representative points, and finer levels add more points. Not every observation appears at every level.

Spectral graph methods describe scale in a different way. Consider a weighted neighbor graph with adjacency matrix $\mathbf{W}$ and diagonal degree matrix $\mathbf{D}$, where $D_{ii} = \sum_j w_{ij}$. With $\mathbf{I}$ the identity matrix, the symmetric normalized graph Laplacian is

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\,\mathbf{W}\,\mathbf{D}^{-1/2}. \tag{2.5}$$

Its eigenvectors $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}$ are ordered by eigenvalue, $0 = \lambda_0 \leqslant \lambda_1 \leqslant \cdots \leqslant \lambda_{N-1}$. A connected graph has exactly one zero eigenvalue, and each additional connected component adds another. Each eigenvector assigns one value to each observation. Eigenvectors with small eigenvalues vary slowly across the graph, so strongly connected observations receive similar values. Eigenvectors with larger eigenvalues can change quickly between neighbors and describe finer variation [33]. The eigenvectors thus form an ordered set of spectral components, from broad to fine structure. Figure 2.1 shows four of them for a connected graph. Laplacian Eigenmaps uses components with the smallest nonzero eigenvalues as coordinates [19]. Its layout is fixed by the eigendecomposition and shows mainly broad structure.

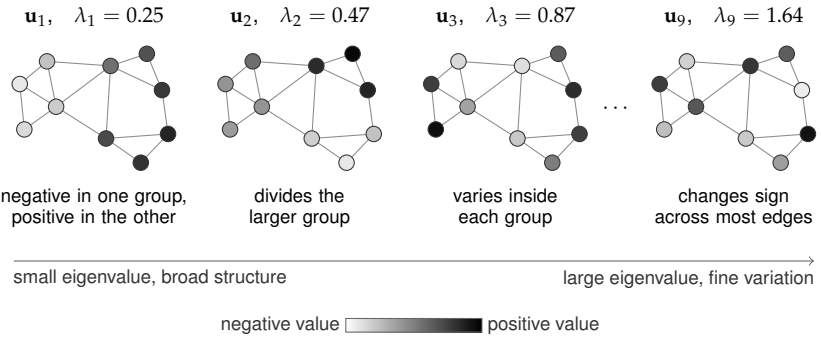


Figure 2.1: Four spectral components of a neighbor graph with ten observations and 16 edges of equal weight. Each panel shades the observations by the values that one eigenvector of the normalized graph Laplacian assigns to them. The component $\mathbf{u}_1$ changes sign across 3 of the 16 edges, and $\mathbf{u}_9$, which has the largest eigenvalue, across 12. The shades are computed from the graph drawn here. Figure 5.3 uses the same ten observations.

Diffusion-based methods use a related spectral ordering. The heat operator $e^{-t\mathbf{L}}$ keeps each eigenvector $\mathbf{u}_i$ and scales it by $e^{-t\lambda_i}$, so fine components fade as the diffusion time t grows. Diffusion Maps similarly scales eigenvectors of a diffusion operator by powers of their eigenvalues [34]. PHATE derives distances from a diffusion process and embeds them with multidimensional scaling [35]. Its final axes no longer correspond one to one to the spectral components used to construct the distances. The same spectral structure appears in neighbor embeddings. The early phase of t-SNE optimization behaves like power iteration on a graph Laplacian [36].

Spectral components offer a way to state at what level of detail an embedding represents relationships. Paper II, SDMP, writes each embedding coordinate as a weighted combination of spectral components and learns the weights with a UMAP-style objective [2]. Components are added in stages, from broad to fine. The method produces a sequence of embeddings, and the contribution of each component to each point can be examined.

2.2.4 Explaining how an embedding was formed

Analysts use embeddings to reason about clusters, trajectories, and relationships among groups [37]. Interpreting a visible pattern may require knowing what produced it. A linear projection makes the contribution of each input feature to a coordinate explicit (Section 2.2.1). A neighbor embedding optimizes the coordinates themselves, and the finished layout has no comparable set of global feature weights. An explanation can then be sought in two places, after the layout is computed or inside its construction.

Most work explains a finished layout. Visual analytics approaches help analysts relate visible regions to input features. DimReader draws axes that show how each feature changes point positions [38]. DimBridge describes visible patterns with logical predicates over features [39]. These methods work in the interface. They help analysts judge a layout without changing how it was constructed.

A smaller line of work builds interpretable quantities into the method. Locality-preserving projections and linear t-SNE compute a layout with one linear map of the input features and optimize that map to keep local neighborhoods [40, 41]. The feature weights are global and apply to every point. Couplet et al. relax this constraint. Their natively interpretable t-SNE learns separate feature weights for each point and keeps the weights of neighboring points similar [42]. FeatureMAP estimates local feature directions within a UMAP-style embedding [43]. In these methods, the quantities used to construct the embedding also support its interpretation.

Both groups of methods explain coordinates through input features. They do not say at which level of detail a pattern forms. Paper II explains coordinates through spectral components (Section 2.2.3), so an explanation states at which level of detail a structure forms [2]. Chapter 5 returns to the choice between explaining a layout after its construction and during it (Section 5.2).

2.2.5 Assessing what a representation preserves

DR results are usually assessed by comparing relationships in the input with relationships in the layout. Local measures ask whether neighbors are kept. Trustworthiness penalizes points that appear as neighbors in the layout but are not neighbors in the data. Continuity penalizes neighbors in the data that the layout separates [44]. Rank-based criteria extend this comparison to neighborhoods of every size [45]. Global measures, such as stress and the rank correlation of pairwise distances, ask whether larger distances are kept [46]. When class labels exist, neighborhood hit and k-nearest-neighbor classification accuracy measure how well labeled groups remain separated [47]. No single measure covers every aspect of a layout. Different measures emphasize different properties, and there is no agreed ground truth for the structure of high-dimensional data [13]. Aggregate scores also say little about where a layout is reliable. Two layouts with similar trustworthiness can distort different regions. A survey of multidimensional projection connects distortion measures to analytical tasks and to visual methods for inspecting distortion [37]. The measures show whether selected relationships are kept. They do not show whether an analyst can tell which relationships a visible pattern reflects. Chapter 3 describes how the included studies combine such measures with use cases.

2.3 Visualizing Structure for Visual Analysis

A relationship captured in a representation still has to be reached by the analyst. Visual encoding maps recorded information to position, color, size, shape, and connection. Interaction lets the analyst select, filter, navigate, reconfigure, and connect views [48]. In network visualization, interaction can act on the view, the visual structure (such as the layout), or the data itself. This section first reviews how visual analytics uses embeddings. It then reviews techniques for showing competing time-series matches, choosing the part and scale of a structure to display, and supporting path following and comparison.

Embeddings show how computational and visual representations meet in practice. Paper IV surveys visual analytics approaches that use embeddings [4]. In some approaches, embeddings are computed within the analysis pipeline and used as an intermediate representation for later analysis. In others, embeddings are computed outside the tool and enter as input data. When an embedding enters as input data, its construction lies outside the interface, and the interface works with the result as given. This division is reasonable in many systems. It becomes a limitation when the task requires a relationship that the adopted embedding does not capture.

2.3.1 Ambiguous results in time-series alignment

Time-series alignment gives an example of a computed result that can contain several plausible answers. When two series are drawn in one chart, the vertical gap between them compares values at the same horizontal position. This assumes that points at the same time correspond. If one process runs later or at a different pace, the gap mixes a difference in value with a difference in timing. An alignment identifies which time points correspond before comparing values.

How the series are aligned before display can change comparison accuracy considerably, and the best alignment depends on the task [49]. Perceived similarity between two series can also differ from similarity computed by a distance measure [50]. When an alignment is computed, it is usually applied as a hidden global step, or it is shown as a path over a cost matrix or as links between matched points.

Dynamic time warping (DTW) is a common alignment method [51]. It computes a cost for every pair of time steps and selects the path of minimum total cost that keeps the temporal order. The path covers every time step, but one time step may correspond to several steps in the other series. DTW does not record how strongly the selected path was preferred. Flat stretches and repeated patterns can produce several paths of similar cost, and DTW keeps

only one of them. Soft-DTW smooths the minimum over path costs so that it can serve as a differentiable loss [52]. It still defines the result through paths.

A soft alignment records the alternatives. It assigns a nonnegative weight to every pair of time steps. For a time step in one series, the weights describe how its correspondence is spread over the other series. The weights may concentrate on one target, split between separate regions, or spread over a wide interval. Optimal transport (OT) computes such weights [53]. Let $\mathbf{C}$ be a cost matrix between the time steps of two series, and let $\mathbf{a}$ and $\mathbf{b}$ be nonnegative weight vectors with equal total mass. Entropy-regularized OT finds the transport matrix

$$\mathbf{T}^* = \arg\min_{\mathbf{T} \in U(\mathbf{a},\mathbf{b})} \sum_{s,t} T_{st}\, C_{st} - \varepsilon H(\mathbf{T}), \tag{2.6}$$

where $U(\mathbf{a}, \mathbf{b})$ is the set of nonnegative matrices whose rows sum to $\mathbf{a}$ and whose columns sum to $\mathbf{b}$, and $H(\mathbf{T}) = -\sum_{s,t} T_{st} \log T_{st}$ is the entropy, with $0 \log 0 = 0$, and $\varepsilon > 0$ sets the weight of the entropy term. Standard OT ignores temporal order. The order-preserving Wasserstein distance (OPW) adds terms that favor matches at similar relative positions and in local order [54].

A transport matrix has the same shape as a DTW cost matrix, but its entries mean something different. A cost entry states whether two time steps are similar in isolation. A transport entry indicates how strongly they correspond after considering all matches and constraints together. In OPW, a strong weight can reflect feature similarity, temporal preference, or both. Paper III, LINEALIGN, treats the transport matrix as the alignment result and connects its weights to the original curves [3]. With this choice, the result records competing matches, and showing them becomes a visualization problem.

Soft alignment adds another requirement. For each time step, the display needs to show which match is favored, how decisively it is favored, and which alternatives remain. Common uncertainty displays, such as error bars, bands, and hypothetical outcome plots, show variation in a value at a known position [55]. Designs for uncertain time intervals show variation in start and end times [56]. In a soft alignment, the variation lies in where a match falls along another ordered time axis. A heatmap of the transport matrix shows every weight, but it separates the weights from the curves that produced them. Paper III uses coordinated views that link correspondence weights back to the original series [3].

2.3.2 Choosing which part and scale to show

Large networks and text collections cannot be shown in full detail at once. A common strategy is to begin with an overview, narrow to parts of interest, and show details on demand [57]. Coordinated multiple views support this strategy by linking displays, so that a selection in one view identifies the corresponding items in the others [58]. Network visualization applies the same strategy by grouping nodes and navigating between levels of a grouping [59].

A network records its links, so an analyst does not need to compute which nodes are related. Displaying a large network still depends on computation. A layout decides where nodes appear, and an analyst reads nearby nodes as related. Embedding-based layouts show this dependency most clearly. Methods such as node2vec learn a vector for each node from random walks over the network, so that nodes that appear near each other in the walks receive similar vectors [60]. A neighbor embedding such as t-SNE then places the vectors in two dimensions (Section 2.2.1). The layout inherits the choices of both steps, and the display does not show which step produced a visible gap between groups. The poster of Study B positions online communities with t-SNE applied to node embeddings from earlier work [7].

A second computation reduces a network before any layout is drawn. Graph sampling selects a subset of nodes and links that keeps chosen properties of the original network, and different methods keep different properties [61]. On a network with a skewed degree distribution, uniform node sampling can yield a sparse sample with few links, while link-based sampling can concentrate on hubs [8]. Visualization uses sampling to keep computation and displays manageable as data grow [62]. When links carry timestamps, the length of the time window adds another choice because it decides which links appear together in one snapshot [63]. A link left out of a sample cannot appear in any view built on it.

Conversation data add time and text to network structure. Sentiment visualization studies how attitudes expressed in text change and how they relate to other features. Many approaches treat a conversation as a monologue or as an exchange between two parties. In a multiparty conversation, any speaker can respond to any other, and the conversation forms a multigraph. Study A links a speaker network, emotion flows within individual dialogues, and the text of single utterances. An analyst can move from an overview of a dialogue collection to the close reading of one utterance [6].

Large networks raise the same need to move from an overview to a selected part. Systems such as ManyNets and SocialAction let analysts select subnetworks of interest [64, 65]. The Study B poster lets analysts select subnetworks

from a large network and retrieve the full neighborhood of a node of interest for closer comparison [7]. VIS-DN links views of time, link attributes, and node degree, and it shows the timeline both in segments of a chosen length and in single days. Analysts choose a time window, filters, and sampling methods [8].

Thus, the level of detail appears on both sides of this thesis. In Studies A and B, the interface provides it through navigation between views. In Paper II, the computation builds it into a sequence of embeddings (Section 2.2.3).

2.3.3 Following paths and comparing results

Node-link diagrams and adjacency matrices are the two main visual representations of networks. In a matrix, each row and column represents a node, and each cell shows whether the two nodes are linked. Matrices avoid the edge crossings and occlusion of node-link diagrams and support accurate reading of dense graphs [66]. What a matrix shows depends on the order of its rows and columns. Reordering algorithms place strongly connected nodes near each other, so that clusters appear as blocks [67]. Interactive tools let analysts steer reordering or manually rearrange rows and columns [68].

Paths are harder to read. A matrix records every direct link, but the links that form a path lie in different rows and columns. Studies find matrices weaker than node-link diagrams for path-related tasks [66]. An ordering chosen to reveal clusters does not bring the cells of a path together. MatLink combines a matrix with node-link elements [69]. Shen and Ma draw paths over a matrix as connected horizontal and vertical lines [70]. In node-link diagrams, Bring&Go brings the neighbors of a selected node into view to support navigation along paths [71].

Paper V adapts this idea to matrices [5]. It reorders the matrix around a selected node so that possible next steps appear nearby, and it keeps the traversed path visible on the matrix. The network data do not change. The technique changes how an analyst follows links that the matrix already records.

Exploring a large structure produces a sequence of states, such as successive samples, levels of detail, or steps along a path. Provenance research preserves this history so analysts can recall, replicate, or recover earlier states [72]. VIS-DN records each choice of time window, filters, and sampling methods in a provenance graph. Its nodes can be ordered by the sequence of selections or by the similarity of the resulting samples, an arrangement that the paper traces to Time Curves [73, 8]. A node-link view then compares the current sample with an earlier one.

There are three arrangements for visual comparison. Juxtaposition places the compared items side by side. Superposition places them in one coordinate system. Explicit encoding computes a comparison quantity, such as a difference, and shows it directly. Each arrangement suits different comparison targets, and all three remain common in practice [74].

Comparison also arises in networks. Multivariate network visualization shows a network's structure along with attributes of its nodes and edges, and balancing both in a single display remains difficult [75]. The Study B poster places selected subnetworks side by side and encodes textual attributes on their links. Analysts can then compare structure and attributes across parts of a large network [7].

2.4 Goals for Visual Encoding and Interaction

The computational and visual work reviewed in Sections 2.2 to 2.3 uses different forms of evaluation. Computational methods are evaluated by the preservation measures of Section 2.2.5. Visual techniques are judged by whether an encoding or interaction supports an analytical action. Visualization research describes this judgment through goals for visual encoding, interaction, and the fit between a design and a task.

Mackinlay proposed two criteria for graphical presentations [76]. *Expressiveness* asks whether an encoding shows all the facts in a set and only those facts. It requires the type of a visual channel to match the type of an attribute, so that ordered attributes use ordered channels. *Effectiveness* asks which expressive encoding lets a reader perceive the information most readily. It gives the most important attributes the channels that readers judge most accurately, such as position [77, 78].

The two criteria can fail independently. Paper IV describes a common design for embedding views [4]. Position encodes similarity in the embedding, and color encodes a separate grouping of the same items. When the color groups cut across the clusters formed by position, readers struggle to read the two encodings together. The survey compares this conflict to the Stroop effect. Both kinds of information are present in the view, so the design is expressive. The difficulty lies in reading them, which affects effectiveness.

A relationship can be present in a display and still be costly to reach. Lam derives seven kinds of interaction cost from Norman's stages of action [79]. These include the cost of deciding what to do next, the cost of physical motion, and the cost of reinterpreting a display after the view changes. Following a path in an adjacency matrix carries such costs (Section 2.3.3).

Access also depends on what the analyst is trying to do. This thesis focuses on three recurring actions. An analyst may identify a relationship, compare relationships, or follow a chain of relationships through the data. Identify and compare appear among the query actions in the task typology of Brehmer and Munzner [15]. Following a path appears in the graph-visualization task taxonomy of Lee et al. [80]. These actions describe the work in this thesis. They are not proposed as a complete task taxonomy.

Each action places a different demand on a display. Identifying a relationship requires that it be visible and distinguishable from other relationships. Comparing requires that the relationships be placed where they can be read against each other (Section 2.3.3). Following requires that each step lead visibly to the next.

The same action can occur at different levels of detail. Bertin distinguishes three levels of reading, namely a single element, a group of elements, and the whole display [81]. An analyst may identify a single point, a cluster, or the overall arrangement of clusters.

These criteria and goals assess the visual representation and interaction after the computational result has been constructed. They do not by themselves distinguish a relationship that is missing from the result from one that the result records but the interface makes difficult to use.

2.5 Summary

The reviewed work covers two sides of visual analysis. Computational methods construct neighbor graphs, embeddings, alignments, and network samples. Visual techniques help analysts identify, compare, and follow relationships in such results. Researchers commonly describe and evaluate the two sides using different terms. Chapter 5 brings them into one account by separating which relationships a computational result preserves from how an analyst reaches them.

3 Methodology and Research Process

The research questions in Section 1.3 ask how to refine structure, reveal it across scales, and make it accessible for exploratory analysis. These are relevant design questions because the data do not determine a single useful representation. The data types studied in this thesis show why. For instance, mapping high-dimensional observations to two dimensions preserves some distances and neighborhoods and distorts others. Noise in the data makes these choices harder because it can create unstable or misleading relationships. The screen adds a further limit. Its space is small, so every view needs to choose what to show.

In dimensionality reduction, some of this distortion is unavoidable. High-dimensional data are often approximately low rank. Their singular values decay quickly, so the data lie close to a linear subspace spanned by a few components. Large data matrices often have this property [82], and so do the matrices that describe many real networks [83]. Many analysis pipelines rely on it and first project the data onto a few tens of principal components, which often keep most of the variance. However, reducing further to two dimensions is different. A finite set of points keeps all its pairwise Euclidean distances in a two-dimensional layout only if the points already lie in a plane. Data whose main variation spans more than two directions do not, so every two-dimensional layout changes some of their distances.

Mapping such data to two dimensions is ill-posed because no layout preserves all distances. Dimensionality-reduction methods for visual analysis solve a substitute problem. Each method defines a loss function that decides which distortions count most [37]. MDS fits all pairwise distances at once [46]. t-SNE and UMAP mainly preserve local neighborhoods and let large distances change [10, 11]. Their loss functions are non-convex, so the resulting layout depends on initialization and parameters [31]. Different runs on the same data can produce different layouts.

Alignment and sampling raise a related problem, because several results can be equally plausible. There is no single representation that is correct for every exploratory goal. The representation still needs to capture structure supported by the data, but its value also depends on the analytical task. In this thesis, the main task is exploratory analysis. The studies help analysts find patterns, examine alternatives, and form hypotheses.

Because the data do not determine one correct representation, this thesis studies representations by building them. It follows a constructive research approach. The six studies, aside from the survey, each build a representation algorithm, an interactive visual interface, or both. Each study begins by identifying a limitation in prior research, constructs an artifact to address it, and examines the result. This approach is technique-driven because the problems arise primarily from existing computational and visual methods, not from a formal requirements study with one group of domain experts [84]. It also follows the high-level logic of design research, in which knowledge is developed by constructing an artifact for a stated purpose and evaluating what it enables [85, 86].

Pragmatism provides the philosophical position for this work because it judges an idea through its consequences in inquiry [87]. In this thesis, this means evaluating whether a refined neighbor graph reduces misleading connections, whether a multiscale layout reveals structures at different scales, and whether an interface helps analysts to perform the identified task. Quantitative measures assess properties that can be measured numerically, while use cases and expert feedback examine whether the results are meaningful in their application domains. Together, these forms of evidence support a pragmatic approach to research [88].

Further, the research process of this thesis builds on two established models of visualization, the information visualization reference model and the nested model of visualization design. This chapter first introduces both models and explains how they determine the stages of the research process and the order of this chapter. It then describes the research approaches used at each stage, from characterizing data and tasks to evaluating the results.

3.1 Two Well-Known Models of Visualization Design

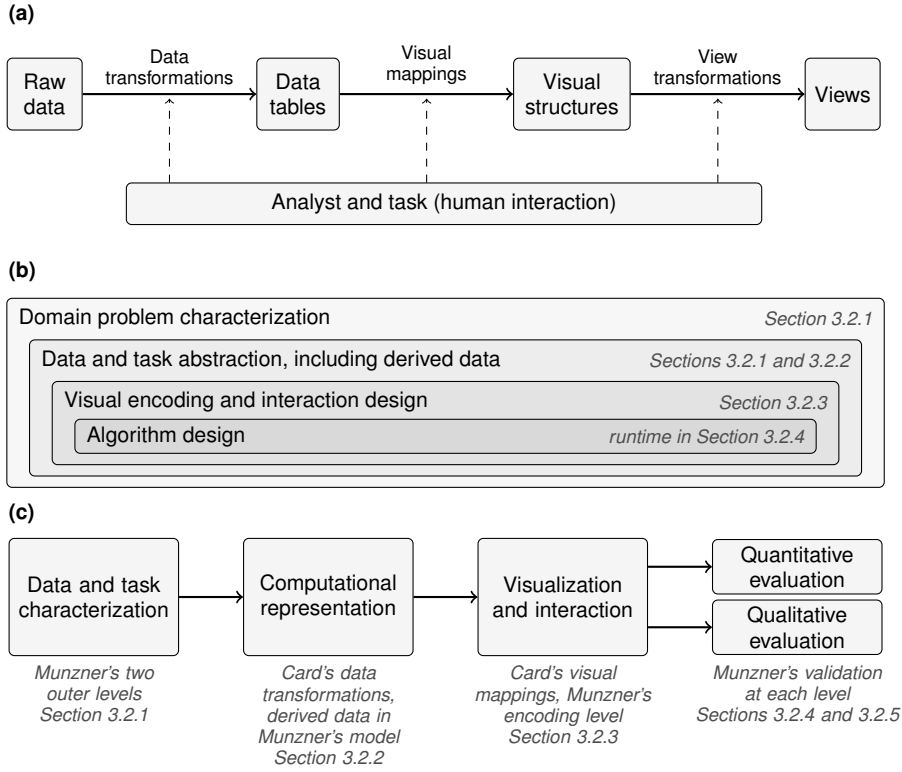


Figure 3.1: Two models of visualization design and the research process of this thesis that builds on them. (a) The information visualization reference model [9]. Solid arrows are transformations of the data, and dashed arrows show that the analyst can change each transformation. (b) The nested model of visualization design [89]. Each level is designed within the decisions of the level around it. The labels on the right name the sections of this chapter that treat each level. (c) The research process of this thesis, which combines the two models into five stages. The note under each stage names the part of (a) or (b) that the stage comes from and the section that describes it.

The information visualization reference model of Card, Mackinlay, and Shneiderman describes how a visualization turns data into a view [9]. It connects four states of the data by three transformations (Figure 3.1a). Data transformations turn raw data into data tables, which describe data items and their attributes in a structured form. Visual mappings turn the data tables into visual structures by assigning attributes to marks and to channels such as position and color. View transformations, such as zooming and panning, decide which part of the visual structures appears on the screen.

Through interaction, the analyst can change each of the three transformations.

This chapter uses the analysis of single-cell data as an example. Biologists often use dimensionality reduction on such data to explore and annotate cell types. Following the reference model (Figure 3.1a), the raw data are high-dimensional gene-expression values measured for many cells. A data transformation computes a neighbor graph that records which cells have similar expression and derives a two-dimensional embedding from that graph. A visual mapping draws each cell as a point at its embedded position and colors the point by an annotated cell type. A view transformation lets the analyst zoom into one group of points.

The reference model describes how data flow through a finished system. It does not state in which order the parts of a system are designed or how each part can be checked. Munzner's nested model addresses these two questions [89]. It divides visualization design into four levels (Figure 3.1b). The outermost level characterizes the problem of a domain, that is, who the analysts are, what data they have, and what questions they ask. The second level abstracts this problem into data types and generic tasks, such as identifying groups or following a path in a network. Munzner counts derived data, such as a neighbor graph computed from measured values, as part of this abstraction [78]. The third level designs the visual encoding and the interaction that support the tasks. The innermost level designs the algorithms that compute the encoding and respond to interaction efficiently.

The levels are nested. Each inner level takes the output of the level around it as its input, so a mistake at an outer level carries into every level inside it. The single-cell analysis shows how this happens. Suppose that noise in the expression values leads the dimensionality-reduction method to connect cells of two different types in the neighbor graph. The two types then overlap in the embedding, and the scatterplot shows one mixed group. This wrong neighbor graph is a mistake at the data abstraction level. A better color scale or a faster layout algorithm cannot separate the two types, because both start from a neighbor graph that has already merged them. Only a corrected neighbor graph, computed again at the data abstraction level, can separate them.

The nested model also assigns its own validation methods to each level [89]. A domain characterization can be checked by observing and interviewing the target analysts. An abstraction can be checked when target analysts use the finished tool and report whether it helps with their questions. An encoding or interaction design can be justified with known perceptual principles and checked by inspecting the resulting images, qualitatively or with quan-

titative measures. An algorithm can be checked by analyzing its complexity and measuring its runtime and memory.

3.2 The Research Process

The two models answer different questions, and this thesis is based on a mixture of both. The reference model gives the order in which an analysis computes its results. The nested model gives the order in which design decisions depend on each other and states how each decision can be validated. The research process combines the two into five stages (Figure 3.1c), which the remaining sections of this chapter follow.

Data and task characterization comes first because it corresponds to the two outer levels of the nested model. Every later decision depends on which data the analyst has and which task the analyst needs to perform. In the technique-driven studies of this thesis, this stage draws on the literature and on the known limits of established methods (Section 3.2.1).

Computational representation comes second. In the reference model, it is the output of the data transformations. In the nested model, it is derived data within the data abstraction. This thesis treats it as a stage of its own for two reasons. Section 1.1 argues that constructing a neighbor graph, an embedding, or an alignment differs from sorting or filtering values because the construction determines which relationships the result records. In addition, two of the studies contribute mainly to this stage. The stage comes before visualization because the visual mapping receives only what the computational representation captures (Section 3.2.2).

Visualization and interaction come third and correspond to the encoding and interaction level (Section 3.2.3). The algorithm level of the nested model does not form a separate stage, because none of the studies contributes mainly to computational speed. Runtime is reported as part of the quantitative evaluation.

Evaluation comes last and has two parts, quantitative and qualitative (Sections 3.2.4 and 3.2.5). The split follows the validation methods of the nested model. Measures of layout quality and runtime check a computational representation and the algorithm that produces it. Use cases and expert feedback check whether a view supports the intended task and whether the patterns it shows can be interpreted in the application domain.

The studies do not give each stage the same weight. A study that contributes at an inner level of the nested model takes the outer levels as given, and its evidence validates only the levels it works on. A study that changes how a neighbor graph is built needs measures of the resulting layouts more than

sessions with analysts, and a study of an interface needs the opposite. Table 3.1 shows the stages on which each study focuses. It tells the reader which kind of evidence to expect from each study in Chapter 4.

Stage	I	II	III	IV	V	A	B
Data and task characterization	○	○	●	●	●	●	●
Computational representation	●	●	○		○		○
Visualization and interaction	○	○	●	●	●	●	●
Quantitative evaluation	●	●					
Qualitative evaluation	○	○	●	○	●		●

Table 3.1: Emphasis of the research process in the five included papers and two related studies. A filled circle denotes a main focus of a study, and an open circle denotes a supporting part.

3.2.1 Data and Task Characterization

Each study in this thesis begins with a literature review. Prior work shows the limits of established methods, and each study restates one such limit in terms of its data and the exploratory task it supports. For dimensionality reduction, the starting points are known limits of neighbor embeddings. For time-series alignment, the starting point is that a hard alignment discards competing correspondences. For networks, the starting points are that an adjacency matrix shows dense connections well but makes paths hard to follow, and that large, attributed networks are hard to explore across levels of detail. The survey is itself a structured literature review. It develops and refines a categorization of papers matching the inclusion-exclusion criteria through screening, independent coding, and discussion of agreements and disagreements.

High-dimensional and relational data raise different problems, though they overlap. High-dimensional observations contain many measured attributes, such as pixels or gene-expression values. Relationships between observations must first be computed, for example as a neighbor graph built from distances. However, as dimensionality grows, it becomes harder to distinguish relationships using standard distance metrics. Measurement noise can change which observations appear to be neighbors, and certain geometric variation on manifolds can place semantically different observations close together in the input space. The same data can also contain structure at several scales. A broad group, a subtype, and variation between individual observations may all be present, and a two-dimensional layout cannot show each of them equally well. In summary, noise motivates RQ1, and structure at several scales motivates RQ2.

Relational data record order or connection directly. Two time series can differ in timing, and repeated or flat patterns can support several plausible matches. The networks studied in this thesis combine topology with attributes, time, text, or repeated links. Their size, skewed degree distributions, and nested levels of detail make it hard to read a complete display. These properties motivate the three design challenges under RQ3. Analysts need to inspect alternative results, decide what to show and at what scale, and track how one selection led to the next.

The application domains show where these problems arise in practice. For high-dimensional data, the thesis covers image collections and single-cell transcriptomics, where dimensionality reduction is commonly used to explore cell types, developmental trajectories, and possible rare populations. For relational data, it covers gene-expression trajectories and financial exchange rates, where similar patterns may occur at different times. It also covers a citation network for path exploration, emotions in multiparty dialogues for natural-language-processing research, and time-stamped, attributed links between online communities for social-media analysis. Across the literature the survey covers, embeddings appear in nine application domains, so the design problem is broader than these examples. The selected domains test the methods in varied settings, but whether the findings hold in other domains remains an open question.

3.2.2 Computational Representation

The studies on computational representation share a common research method. Each study replaces one component of an established method and keeps the others fixed, so that a difference in the result can be attributed to that component. For neighbor embedding, two studies change consecutive stages of the same pipeline. One changes how the neighbor graph is constructed and keeps the UMAP layout objective. The other keeps the neighbor graph and changes how the layout is built from it.

In the remaining studies, the research focuses on visual encodings and interaction, and the computation supports them. For time-series alignment, the thesis adopts an established order-preserving optimal-transport formulation because its weighted correspondences keep competing matches available for visual analysis. Such computations are necessary parts of the interfaces, but the contribution of these studies lies mainly in how analysts control and inspect them.

3.2.3 Visualization and Interaction

The research method for visualization and interaction differs from the method for the computational representation. A visualization design has no

single component that can be replaced while the others stay fixed, because tasks, visual encodings, and interactions depend on one another. Thus, the work starts from what the computational result captures and what the analyst needs to do with it. For the included studies, prior visualization research then suggests how to meet this need.

Two recurring design principles address the three design challenges under RQ3. The first keeps each derived result connected to the data it came from. For example, alignment views show the weighted correspondences next to the original curves. The second principle makes the analyst's choices visible and revisitable. The network sampling interface keeps each sample with the filters that produced it, so an analyst can return to an earlier sample and compare it with a later one.

In the dimensionality-reduction studies, visualization has a narrower role. Static plots and inspection views serve the qualitative evaluation of the computational representation (Section 3.2.5).

3.2.4 Quantitative Evaluation

Quantitative evaluation in this thesis applies to the two dimensionality-reduction studies. Each measure favors one property of a layout, so each study uses measures that can detect its own change. One study changes which observations become neighbors, so it uses label-based measures. These check whether observations with the same label end up close together and whether known groups become better separated. The other study changes how graph structure appears at successive scales, so it uses label-free measures of neighborhood preservation, distance order, and layout stress. Both studies compare against established dimensionality-reduction methods, vary key parameters, and measure runtime. Label-based measures cover only classes that are already known, and label-free measures cannot show whether a visible group is meaningful in the application domain.

3.2.5 Qualitative Evaluation

Most studies in this thesis rely on qualitative evaluation. It checks whether a pattern in a layout or view can be traced back to the data and interpreted in its application context. The check takes a different form in the two kinds of studies. Dimensionality-reduction studies compare layouts side by side and inspect observations within visible groups. For single-cell data, for example, cell-type annotations show whether a visible group contains one cell type or several. The interface studies rely on use cases. A use case follows a concrete question through the prototype and checks each derived view against the original data.

In addition to use cases, a few studies draw on domain experts directly. A use case shows that a pattern can be found, but the authors cannot always judge whether the pattern is meaningful in the domain. Domain experts bring knowledge that the data themselves do not contain. In the network-sampling study, for example, three computational social scientists used the interface in semi-structured interviews and judged whether its workflow fits how they analyze online communities and where it falls short.

4 Summary of Included Papers and Related Studies

This chapter summarizes the five included papers and the two related studies in the order of the List of Publications. Each section states the problem, the approach and its evidence, and the study's role in the thesis. Table 3.1 shows which stages of the research process each study focuses on.

Paper I refines neighborhood structure and answers RQ1. Paper II reveals structure from broad to fine scales and answers RQ2. Paper IV characterizes the design space of RQ3. Papers III and V and Studies A and B address the three design challenges within that space. Chapter 6 compares the studies.

4.1 Paper I: MAPLE

Problem. UMAP builds a weighted k-nearest-neighbor graph from a fixed distance function and then optimizes a low-dimensional layout [11]. The graph is not revised during layout optimization. Noise, distance concentration, and curved manifolds can create spurious neighbors, and the layout inherits those connections [12, 1].

Method. MAPLE treats the initial neighbors as tentative self-supervised signals [1]. Encoder and projector networks learn an intermediate embedding with a maximum manifold capacity objective. The method then builds

a refined neighbor graph in this learned space and applies UMAP's layout optimization.

Tests on image, synthetic, and single-cell data combine quantitative measures, visual comparisons, and expert-guided analysis. They show clearer cluster separation and finer subcluster resolution than UMAP, with the largest gains on data whose local neighborhoods contain substantial high-dimensional variation [1].

Role in this thesis. MAPLE answers RQ1 by changing how neighborhood relationships are computed. Its unique contribution is to treat the initial neighbors as tentative and refine them in a learned space before constructing the final graph. This targets spurious connections caused by noise and curved manifolds before they reach the layout.

4.2 Paper II: SDMP

Problem. Nonlinear dimensionality reduction faces two connected trade-offs. Neighbor embeddings [21, 31] preserve fine neighborhoods but do not separately represent broad and fine structure. Spectral methods [90, 19] expose broad graph structure but offer less freedom to fit local detail. Freely optimized coordinates are flexible, yet they do not show which graph scales produced a visible pattern [2].

Method. SDMP expresses each embedding coordinate as a learned combination of graph Laplacian eigenvectors, called spectral modes [2]. Low-frequency modes describe broad variation over the neighbor graph. Higher-frequency modes permit finer changes between neighbors. The method starts with low-frequency modes and expands the available spectral subspace in stages while keeping the data, graph, and objective fixed.

Quantitative tests and case studies on synthetic, image, and single-cell data examine neighborhood preservation, manifold continuity, and compactness in reduced spectral subspaces. The case studies also trace cluster and trajectory patterns to spectral contributions at the dataset and point levels [2].

Role in this thesis. SDMP answers RQ2 by producing a staged sequence of embeddings in progressively larger spectral subspaces. Each stage adds higher-frequency modes and continues optimization from the preceding result. Early stages restrict the layout to smooth variation over the neighbor graph and emphasize broad organization. Later stages permit bigger differences between neighbors and can express finer local structure. The sequence lets analysts see when a cluster or trajectory becomes visible. The spectral

decomposition also shows which modes shape the layout overall and which contribute most strongly at individual points.

4.3 Paper III: LINEALIGN

Problem. A superposed line chart compares values at the same horizontal position and assumes that the two time axes already correspond. A timing difference can then appear as a value difference [49]. Dynamic time warping computes a correspondence, but its single path omits competing matches [51]. Flat stretches and repeated patterns can give one time step several plausible partners [3].

Method. LINEALIGN visualizes soft alignments produced by order-preserving optimal transport [3]. A transport matrix assigns a weight to every candidate pair of time steps, recording dominant and competing matches. The paper derives three visualization challenges and seven analytical tasks from this output, and then coordinates correspondence profiles with the original curves.

Two use cases, on single-cell trajectories and currency exchange rates, show how the interface separates temporal displacement, correspondence ambiguity, pace change, and remaining feature mismatch. They also show how observed similarity and temporal preference contribute to a match [3].

Role in this thesis. LINEALIGN addresses RQ3 and DC1. Its unique contribution is to formulate soft time-series alignment as a correspondence-ambiguity visualization problem and to design coordinated views for it. The interface shows competing matches, timing displacement, pace change, and feature mismatch against the original curves, making the alignment visible and checkable.

4.4 Paper IV: VA + EMBEDDINGS STAR

Problem. Embedding-based visual analytics spans many application domains and uses embeddings at different points in the analytical process. Before this report, the field lacked a systematic account covering both the computational and visual aspects of these systems [4].

Method. This report reviews 122 peer-reviewed papers that compute embeddings in a visual analytics pipeline or explicitly show them in a visual representation [4]. The authors screened 1,704 candidate papers. Three annotators independently assessed a pilot set, discussed disagreements, and used the results to refine the inclusion criteria and categorization schema. The

final schema covers application domain, input data, computation, analytical task, visual representation, interaction, and evaluation. It distinguishes *embeddings4VA*, where embeddings support a visual analytics system, from *VA4embeddings*, where visualization supports the study of an embedding. The report identifies 15 subfields across nine application domains and provides an interactive browser for the categorized corpus.

Role in this thesis. Paper IV provides the field-level answer to RQ3. Its unique contribution is a categorization of 122 visual analytics papers across application domains, data, embedding methods, tasks, visual representations, interactions, and evaluations. The distinction between *embeddings4VA* and *VA4embeddings* shows where an embedding enters the visual analytics process and where design work is concentrated. It also shows that few surveyed systems change how they construct embeddings [4].

4.5 Paper V: Matrix Snap&Go

Problem. Adjacency matrices avoid the edge clutter of node-link diagrams [66], but they show graph paths as separate cells. Following a path requires finding the next edge at each step and remembering the route [66, 91]. Existing matrix reordering methods mainly expose blocks and do not support this task directly [67].

Method. Matrix Snap&Go brings the neighbors of a selected node next to it and records the traversed route as a polyline over the matrix [5]. A synchronized radial node-link view gives a second view of the same path. Tests on synthetic graphs and a citation-network use case demonstrate the technique across different node and edge densities.

Role in this thesis. Paper V addresses RQ3 and DC3. Its unique contribution is an interaction that reorders a matrix around the selected node while drawing the accumulated exploration path. The network relationships already exist in the matrix. The contribution makes them easier to follow and keeps the analyst's route visible.

4.6 Related Study A: Conversation Trajectories

Problem. Multiparty conversations combine speaker relationships, utterance order, emotion labels, and text. A view of the speaker network alone does not support moving from a collection of episodes to one conversation, then to the utterance that produced a pattern [6].

Method. The prototype uses the Multimodal EmotionLines Dataset and represents successive utterances as links between speakers [6]. Episode summaries show dominant emotion transitions. Selecting an episode isolates its local network, selecting a conversation opens an emotion timeline, and selecting an utterance opens its text. The poster reports an implemented prototype and data-processing pipeline.

Role in this thesis. Study A addresses RQ3 and DC2. Its unique contribution is a linked, multilevel interface for moving from episode summaries to a local conversation network, emotion distributions, and individual utterances. It shows how a large conversation collection can be reduced step by step while preserving access to the text behind each selection.

4.7 Related Study B: Multivariate Subnetworks and VIS-DN

Study B consists of two publications on the same Reddit hyperlink network. The poster compares selected subnetworks [7]. The VIS-DN manuscript supports the earlier decisions that produce a sample [8].

Problem. Large dynamic networks cannot be shown in full [57, 62]. Their links can carry temporal and multivariate attributes [92, 63]. Analysts need to select manageable subnetworks, understand how each selection was produced, and compare the resulting topology and attributes [61, 8].

Method. The Study B poster introduces side-by-side comparisons of two user-defined subnetworks [7]. Node-link views show their structure, while linked histograms, timelines, and attribute encodings support comparisons of linguistic properties and ego networks.

VIS-DN builds on the poster by moving upstream from comparing selected subnetworks to supporting how analysts produce those selections [8]. It links global statistics for time, topology, and link attributes to twelve sampling methods. Analysts choose time windows and structural filters, inspect a sample, and revise their choices. A provenance graph records each state, and a comparison view distinguishes shared and sample-specific nodes.

The poster demonstrates its interface on 40 months of Reddit data [7]. VIS-DN adds use cases on Reddit and online-course activity networks, interviews with three computational social scientists, and performance tests on synthetic networks with up to about 500,000 links. The manuscript does not report a controlled user study [8].

Role in this thesis. Study B addresses RQ3, DC2, and DC3 through two successive contributions. The poster shows how analysts can compare subnetworks at different structural and multivariate scales. The manuscript extends this work into a sampling workflow that uses global distributions to guide local selection and records the resulting exploration history.

5 Conceptual Framework

The thesis develops one algorithm that learns the space in which a neighbor graph is constructed and another that controls which spectral components form its layout (Papers I and II). The other studies contribute interactive visualization techniques or a survey. Chapter 3 reviewed each study's research methodology under both the reference model [9] and the nested model [89]. Building on the five-stage research process (Figure 3.1c), this chapter takes a closer look at each study's contributions and places them within a common conceptual framework, so that they can be compared. The framework describes each study by the tasks it supports and by the part of a shared pipeline it changes. The chapter first motivates the framework from the tasks of the studies, then places the studies within the pipeline, and finally describes what each study adds to the methods it builds on.

5.1 Motivation from Analysis Tasks and Types of Relationships

The works in this thesis share two common motivations: one from the tasks analysts perform and one from the relationships the data contain. This section takes up the two in turn.

Visualization researchers have characterized visualization tasks at several levels, from high-level goals to low-level actions on data items. High-level

Study	Tasks stated in the study	Closest established terms
Paper I (MAPLE)	Separate groups, resolve subgroups within a group, and reveal rare populations	Cluster, Find Anomalies [93]
Paper II (SDMP)	See groups at broad and fine levels of detail. Find the stage at which a visible pattern appears and the spectral components that produce it	Cluster, Explain [94]
Paper III (LineAlign)	Locate the match of a time step and its displacement. Characterize how concentrated its candidate matches are and identify competing matches. Compare timing across regions. Check a match against the original curves	Eight of the ten low-level tasks, all except Sort and Cluster
Paper V (Matrix Snap&Go)	Create and follow a path in an adjacency matrix	Follow Path [80]
Study A (Conversation trajectories)	Navigate from a network of speakers to single utterances and identify dialogues of interest	Navigate, Filter [15]
Study B (VIS-DN)	Discover and describe observations, identify their causes, collect evidence, compare sampled subnetworks, and explain their differences	Analysis goals of Lam et al. [95]

Table 5.1: Tasks from the six studies in this thesis and the closest terms in established task vocabularies.

goals, such as discovering patterns or forming hypotheses, describe why an analyst uses a visualization [15, 95]. As stated in Section 1.3, this thesis aims to facilitate exploratory data analysis.

However, pursuing a high-level goal directly is hard, and analysts reach it through mid- and low-level tasks, such as *Cluster*, *Find Anomalies*, or *Follow Path*, which describe what they do with the data [93, 80]. Each study's tasks are mapped to these established low-level terms (Table 5.1). In the nested model, this mapping occurs at the abstraction level (Section 3.1), where a domain question becomes a generic task. For example, finding rare cell populations in single-cell data becomes *Find Anomalies*. Because the tasks are generic, studies across different datasets can be compared, and the mapping shows that they serve the same goal, exploratory analysis, through different lower-level tasks. The abstraction level also sits above visual encoding and algorithm design, so these tasks are what each study's design has to support.

In this sense, they indicate the analytical capacity of visual analysis, and a new algorithm or technique increases this capacity when it lets analysts carry out more of them. For example, revealing structure at several scales lets analysts find the stage at which a visible pattern appears (Paper II).

The tasks fall into two groups that follow the research questions (Table 5.1). The tasks of the dimensionality-reduction studies concern which groups appear and at what level of detail (RQ1 and RQ2), whereas the tasks of the other studies concern locating, following, and comparing relationships within a view (RQ3).

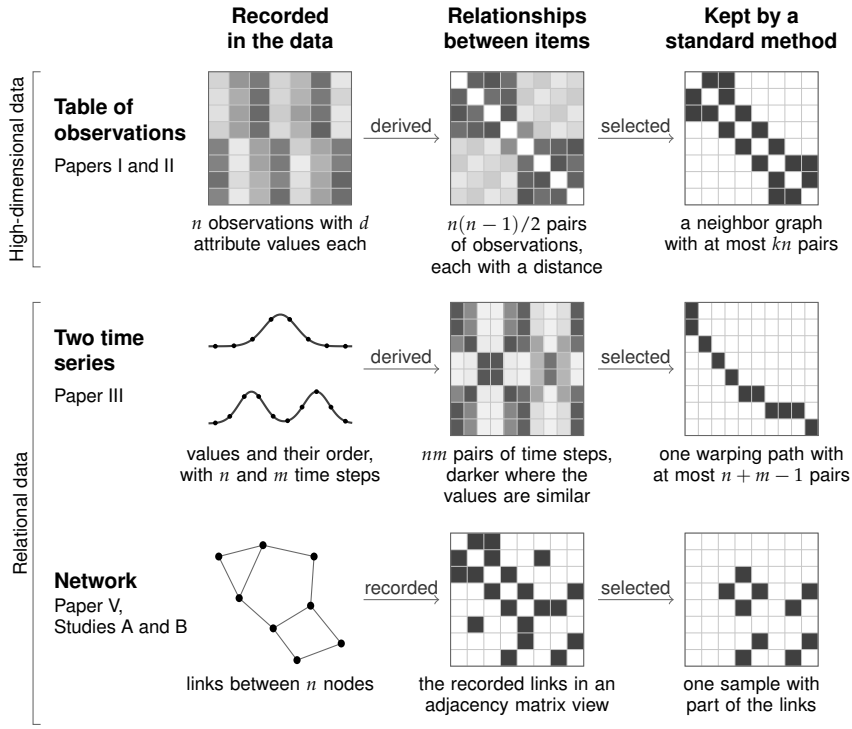


Figure 5.1: From what the data record (left) to all candidate pairs of items (middle) and the pairs that a standard method keeps (right). Each row is one case of data in this thesis. In the upper two rows, darker cells mark more similar pairs. In the network row, dark cells mark linked nodes.

The second motivation is about what these analytical tasks act on. Each lower-level task acts on a specific object, and in this thesis, that object is a relationship between data items. For instance, *Cluster* depends on which observations are similar, *Follow Path* on which links connect the nodes, and *Compare* on how two items correspond.

These relationships do not reach the analyst directly. Figure 5.1 traces them from the data to the result of a standard method. The data often do not record the relationships that the tasks need (left), so the relationships must first be derived (middle). Their number grows quickly, and a standard method keeps only a small, sparse subset (right). For instance, a table of n observations yields $n(n-1)/2$ pairwise distances, of which a neighbor graph keeps at most kn. Time series and networks follow the same pattern (lower rows).

A visualization is usually built from what the standard method keeps (Figure 5.1, right), and that result is one selection among many possible ones. In the nested model [89], this selection is derived data at the abstraction level (Section 3.1). A pair left out there, or a wrong pair kept, carries into every inner level, as in the single-cell example where noise merged two cell types. Thus, whether an analyst can perform a task depends on two things: which relationships the computation keeps and how the view lets the analyst reach them.

5.2 The Overarching Pipeline of This Thesis

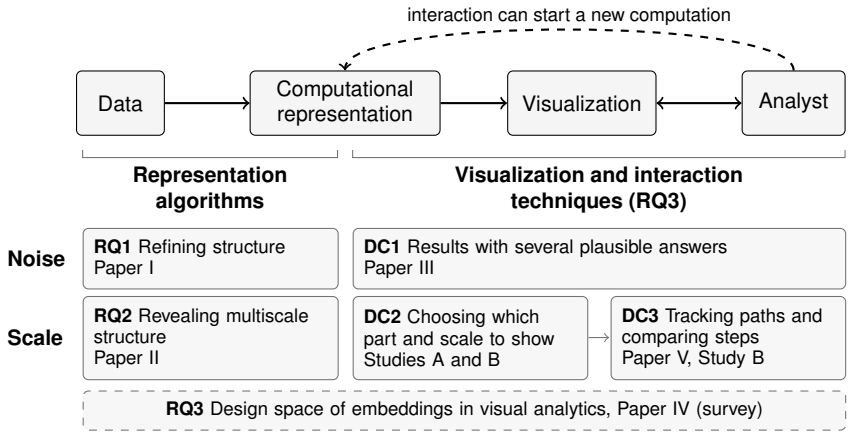


Figure 5.2: A four-stage pipeline condensed from the reference model of Section 3.1 (top) and the parts of it that this thesis contributes to (bottom).

This section maps the research questions and design challenges of Section 1.3 onto the reference model (Figure 3.1a). It first condenses that model into a pipeline of four stages, from the data through a computational representation and a visualization to the analyst (Figure 5.2, top). The pipeline has two parts, corresponding to Section 5.1. In the first part, a representation algorithm decides which relationships the computational representation keeps. In the second, visual encoding and interaction determine how the analyst reaches them. The two parts also feed back into each other, as when an ana-

lyst changes a hyperparameter and the representation is recomputed. It then places each question and challenge under the part it addresses, as introduced in Section 1.2 (Figure 5.2, bottom).

Study	Pipeline	Standard method	Focus of the study
Paper I (MAPLE)	$\mathbf{X} \rightarrow \boxed{\mathbf{E}} \rightarrow \mathbf{W} \rightarrow \mathbf{Y}$	Neighbor graph from distances in the input space	Neighbor graph from a learned space
Paper II (SDMP)	$\mathbf{X} \rightarrow \mathbf{W} \rightarrow \boxed{\mathbf{U}_S\mathbf{P}_S} = \mathbf{Y}$	One embedding	A sequence of embeddings from broad to fine structure
Paper III (LineAlign)	$(\mathbf{x}_1, \mathbf{x}_2) \rightarrow \boxed{\mathbf{T}} \rightarrow \boxed{V(\mathbf{T}; \mathbf{x}_1, \mathbf{x}_2)}$	One warping path	Weights for every candidate match
Paper IV (Survey)	$\mathbf{X} \rightarrow \mathbf{E} \rightarrow V(\mathbf{E})$	Surveys other work	
Paper V (Matrix Snap&Go)	$\mathcal{G} \rightarrow \boxed{V_v(\mathcal{G})}$	Adjacency matrix in one fixed order	The same matrix, reordered around the selected node
Study A (Conversation trajectories)	$\mathcal{G}(t) \rightarrow \boxed{V_1 \leftrightarrow \cdots \leftrightarrow V_4}$	Views of episodes, speakers, and utterances, kept apart	The same views, linked across levels of detail
Study B (Subnetworks)	$\mathcal{G}(t) \rightarrow \mathbf{E} \rightarrow \mathbf{Y} \rightarrow \boxed{\mathcal{G}_1 \parallel \mathcal{G}_2}$	One view of the whole network over a t-SNE layout	Two selected subnetworks side by side
Study B (VIS-DN)	$\mathcal{G}(t) \rightarrow \boxed{\theta} \rightarrow \mathcal{G}_\theta \rightarrow \boxed{\mathcal{G}_\theta \parallel \mathcal{G}_{\theta'}}$	One sample from settings chosen once	Samples recorded together with their settings

Table 5.2: The pipeline of each study, with the result of a standard method next to the result that the study focuses on. A solid box marks the computational representation that a study works on, and a dashed box marks the focus of its visualization or interaction technique. Paper IV surveys other work and has no box. $\mathbf{X}$ is the input data, $\mathbf{E}$ an embedding, $\mathbf{W}$ a neighbor graph, $\mathbf{Y}$ layout coordinates, $\mathbf{T}$ a correspondence matrix, and $\mathbf{U}_S\mathbf{P}_S$ a combination of the first S spectral components of $\mathbf{W}$. $\mathcal{G}$ is a network, and $\mathcal{G}(t)$ a network whose links carry timestamps and attributes. θ holds the sampling settings, and V_v is a view ordered around the node v. V is a view, $\leftrightarrow$ links views, and $\parallel$ places views side by side.

5.3 Comparing the Analysis Pipelines of the Studies

Each study sits in one part of the pipeline (Figure 5.2). The same placement is given in more detail by writing each study's pipeline in a common notation, from the input data on the left to the view on the right (Table 5.2). A

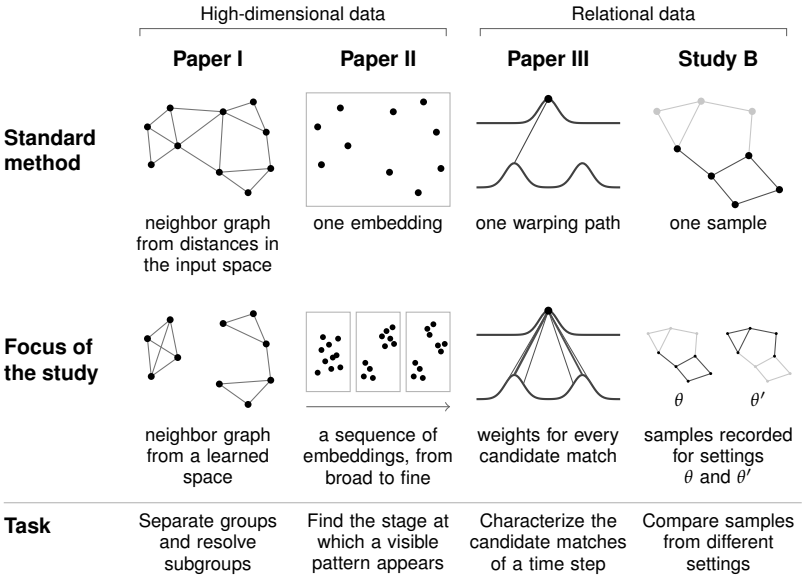


Figure 5.3: How four studies in this thesis extend the computed result for one of their tasks.

box marks the part that a study changes, and the unboxed parts come from existing methods.

The boxed part is described twice, once as a standard method builds it and once as the study builds it (last two columns of Table 5.2). The comparison returns to the selection step (Figure 5.1). A standard method selects a few relationships from many candidates before the analysis begins, using a criterion fixed in advance, such as the distance between observations in the input space or the cost of a match. This early selection can go wrong in two ways, corresponding to the two research problems in Section 1.2.

Under noise, the method cannot tell which relationships to keep. Scale creates another problem, as no single selection serves every step of an exploration. How much detail to show, and for which part of a large network, depends on what the analyst is examining. Moreover, the analyst makes these decisions only after seeing the first views, so a selection made beforehand cannot anticipate them.

The studies in this thesis respond to the two problems in different ways. When noise is the problem, they either learn a better space for constructing neighbors (Paper I) or keep the candidate matches that the method cannot distinguish (Paper III). When scale is the problem, they postpone the selec-

tion so that the analyst can make it during the analysis (Papers II and V, Studies A and B).

Apart from Paper I, these studies leave the analyst to choose among alternatives, and this comes at a cost. A method that chooses for the analyst computes and returns one answer, leaving the analyst with no alternatives to compare. Handing the choice to the analyst sacrifices this simplicity in one of two ways. Candidates kept in the computation must also be shown, so the view draws several weighted matches for a time step where a standard view draws one line. An interactive selection keeps each view simple, but the analyst sees one alternative at a time and loses earlier ones unless the system records them. Study B records each sample together with its settings for this reason. In four studies, Papers I to III and Study B, the change reaches the computation, and each change serves one task of the study (Figure 5.3).

6 Synthesis

What analysts can see depends on how a method processes the data and displays the result. For example, the choice of neighbors affects which points a layout brings together. Restricting where points can be placed affects which patterns the layout can show. A computed result may contain several plausible matches between two time series, but a view showing only one match leaves the alternatives unseen. The studies in this thesis examine changes to these computations, views, and interactions. Comparing the studies shows what each change makes possible, what remains unchanged, and how its effects can be evaluated.

6.1 Comparing MAPLE and SDMP (Papers I and II)

Papers I and II begin from two decisions that UMAP fixes before layout optimization. UMAP constructs a neighbor graph $\mathbf{W}$ from observations $\mathbf{X}$ in the input feature space, then optimizes an independent coordinate for every point in the layout $\mathbf{Y}$. Paper I reopens the first decision by learning a space $\mathbf{E}$ in which the graph is rebuilt. Paper II reopens the second by restricting the layout to an expanding spectral subspace $\mathbf{U}_S\mathbf{P}_S$ (Figure 6.1(a)). Both papers keep UMAP's cross-entropy layout objective. Paper I changes the relationships supplied to that objective. Paper II changes the set of layouts over which the objective is optimized.

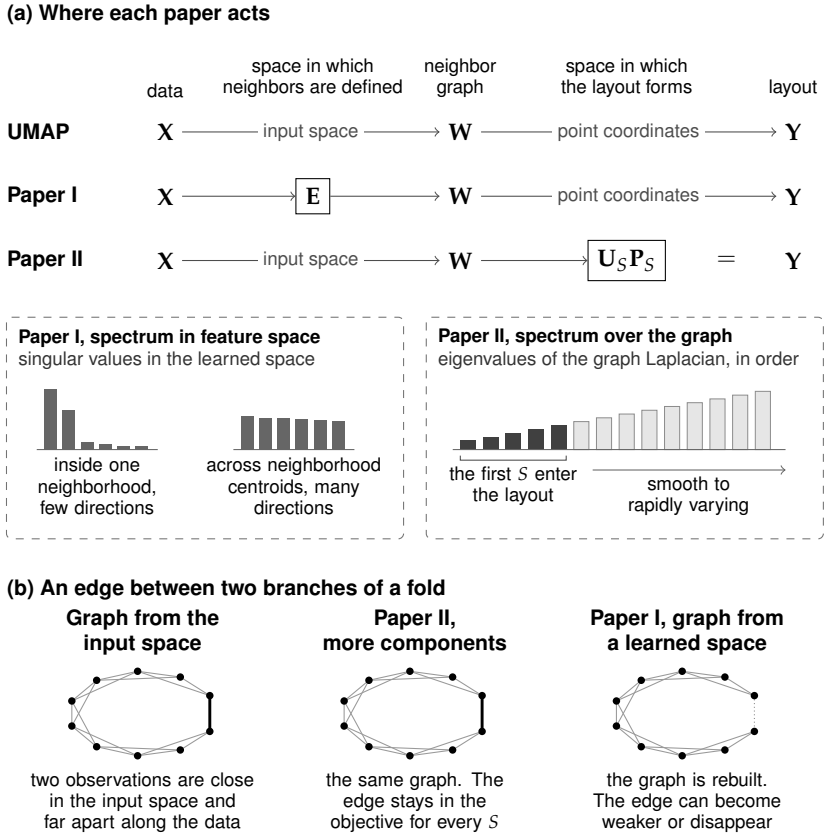


Figure 6.1: Papers I and II change two different representations in a standard neighbor-embedding pipeline. (a) UMAP constructs a neighbor graph $\mathbf{W}$ from the input space and optimizes point coordinates $\mathbf{Y}$. Paper I adds a learned space $\mathbf{E}$ in which neighbors are defined. Paper II lets the layout form in an expanding spectral space $\mathbf{U}_S\mathbf{P}_S$, from broad to fine structure. The dashed frames locate the two spectra involved. (b) If the input-space graph connects two branches of a folded set, changing S changes how the fixed graph becomes a layout but keeps the edge in the objective. Learning the space in which the graph is built can change that edge.

Paper I separates observation from analysis [1]. The matrix $\mathbf{X}$ records the data as measured, for example image pixel values or gene counts of cells, and distances in that space need not reflect the similarity that the analysis needs. Paper I treats each observation and its nearest neighbors as views of one local manifold and learns $\mathbf{E}$ by comparing these views, as multi-view self-supervised learning does with augmented images [96, 97, 26]. The neighborhoods are rebuilt in $\mathbf{E}$ during training, so $\mathbf{X}$ sets only the starting neighborhoods.

Paper II separates the relationships in a graph from the positions that express them [2]. In UMAP every point moves freely, so the coordinate parameterization does not distinguish broad variation over the graph from finer variation. Paper II writes each layout coordinate as a weighted combination of the first S spectral components of the graph, ordered from broad to fine variation. The level of detail becomes a parameter, and each position can be traced to the components that formed it. Paper I changes which relationships the graph records. Paper II changes how much broad and fine structure of a fixed graph can enter the layout. These are two aspects of computational representation, the relationships present in a result and their level of detail.

How neighbor averaging relates feature-space variance to graph smoothness. The word *spectrum* refers to different quantities in the two papers. In Paper I, singular values describe how variance is distributed among directions in the learned feature space. The MMCR objective [26] reduces the sum of the singular values for each neighborhood and increases it for the matrix of neighborhood centroids. Each neighborhood concentrates its variance in a few directions, while the centroids spread across many directions. These changes affect which observations become neighbors when $\mathbf{W}$ is rebuilt.

In Paper II, the eigenvalues of the graph Laplacian $\mathbf{L}$ order signals by how rapidly they change across graph edges. For a normalized eigenvector $\mathbf{u}_s$, the eigenvalue

$$\lambda_s = \mathbf{u}_s^\mathsf{T} \mathbf{L} \mathbf{u}_s$$

measures how strongly the signal changes across graph edges. Small eigenvalues belong to smooth signals that describe broad structure. Large eigenvalues belong to signals that change quickly between neighbors. The Laplacian is symmetric and positive semidefinite, so its eigenvalues equal its singular values numerically. Their meaning still differs from the singular values in Paper I. One spectrum describes variance in feature space. The other describes variation over a graph.

Neighbor averaging suggests a possible connection between the spectra. In Paper I, each centroid averages an observation with its k nearest neighbors

in **E**. Coordinates that vary little within a neighborhood change little under this averaging, while some within-neighborhood variation can cancel. One possible interpretation is that the centroid term favors variation that remains visible after neighborhood averaging. Whether this amounts to a preference for graph-smooth variation depends on the averaging operator.

For the original MMCR loss, a precise spectral connection holds under the view-group construction and normalization assumptions analyzed by Yerxa et al. [26]. In that setting, an optimal embedding can have left singular vectors given by eigenvectors of the averaging matrix, with singular values proportional to its largest eigenvalues. These eigenvectors correspond to the smallest eigenvalues of the associated Laplacian.

This result does not directly apply to Paper I. Its k-nearest-neighbor graph used to compute the MMCR centroids is not necessarily symmetric; the graph is rebuilt every N epochs by default, and the loss includes an additional neighborhood term [1]. The theorem motivates a possible interpretation of the centroid term, but does not establish its effect in Paper I. Paper II directly adds the smoothest Laplacian eigenvectors of its fixed weighted graph first. Graph smoothness may thus offer a useful perspective for comparing the methods, although their objectives and graphs differ. Neither paper derives this connection. Paper II suggests that different objectives and graph modifications may amplify or suppress different parts of the spectrum [2].

Limits of each method. Each method depends on a particular kind of variation in the data. Paper I needs variance that it can redistribute. When the input has been heavily whitened and the neighborhoods are fuzzy, the centroids have no additional variance to spread, and the layout contracts into compact clusters [1]. Paper II needs few components only when the structure over the graph is smooth. A Swiss roll can be laid out with few components. Single-cell data need a large part of the spectrum, which indicates that much of their structure varies quickly between neighboring cells [2]. Both methods also depend on the graph built from the input. If neighborhoods are close to random, the graph has few smooth components beyond the trivial eigenvector. The first stages of Paper II then have little broad structure to show, and the centroid term of Paper I has little smooth structure to favor.

What each method can and cannot change. The spectral components and the objective of Paper II both come from **W**, so changing S cannot repair a missing or spurious edge. An edge between two branches of a fold stays in the objective for every S, while Paper I can weaken or remove it by rebuilding the graph in **E** (Figure 6.1(b)). Noise and fine structure both vary quickly between neighbors, so raising S to show fine structure also lets

in noise (Figure 1.2(b)). By weakening edges associated with noise, Paper I acts before any spectral component is computed. It does not show how the refined graph contributes to the layout at different scales. When the graph already records the intended relationships and a restricted layout omits fine detail, adding components is enough, and learning another feature space is unnecessary. The two operations are complementary but not interchangeable. Paper I changes the relationship structure that reaches the layout objective. Paper II changes the coordinate functions available to satisfy that objective.

Combining the two methods. Because the two papers act at consecutive points, their changes can compose into one pipeline,

$$\mathbf{X} \longrightarrow \boxed{\mathbf{E}} \longrightarrow \mathbf{W}_E \longrightarrow \boxed{\mathbf{U}_{E,S}\mathbf{P}_S} = \mathbf{Y}, \tag{6.1}$$

in which Paper II decomposes the graph that Paper I learns. Paper I proposes the same role, as a graph-refinement stage, for t-SNE and Laplacian Eigenmaps, which also begin from a neighbor graph [1]. This thesis evaluates neither the combination nor these uses.

Manifold capacity theory motivates one possible outcome of the combination. MMCR aims to improve the linear separability of manifolds associated with different items [26]. If Paper I weakens connections between groups while preserving strong connections within them, the graph may develop small Laplacian eigenvalues associated with those groups. The corresponding low-frequency subspace may then express group membership more clearly [98]. Group distinctions that require high-variation components in the graph built from $\mathbf{X}$ might become expressible with fewer low-frequency components in $\mathbf{W}_E$. Paper I could also separate the groups without reducing the number of components needed, or change which groups require high-variation components. Comparing the eigenvalues, response profiles, and reconstruction errors of the two graphs could help assess these possibilities (Section 6.3).

Different references for evaluation. The two papers evaluate their layouts against different references, and each choice follows from the stage that the paper changes. Paper II keeps the graph, so the input space is the reference that its layout should match. It compares the layout with the input space using label-free measures and avoids label-based ones, which can reward compact classes while the layout distorts the geometry of the data [2]. Paper I changes the graph on purpose, so agreement with the input space cannot show whether the change is correct. Measures such as trustworthiness penalize neighbors gained or lost relative to the input k-nearest-neighbor graph, so they count the neighbors that Paper I corrects as errors.

Paper I uses class labels as the reference [1]. An evaluation of the combination would have to judge the two stages separately, the change from $\mathbf{W}$ to $\mathbf{W}_E$ against an external reference such as labels, and the layout against $\mathbf{W}_E$.

6.2 General Design Considerations

Paper IV maps the design space across various visual analytics approaches that use embeddings (Section 4.4). Papers III and V and Studies A and B take up three design challenges within this space. Figure 6.2 compares one technique for each challenge with the view that the study takes as given.

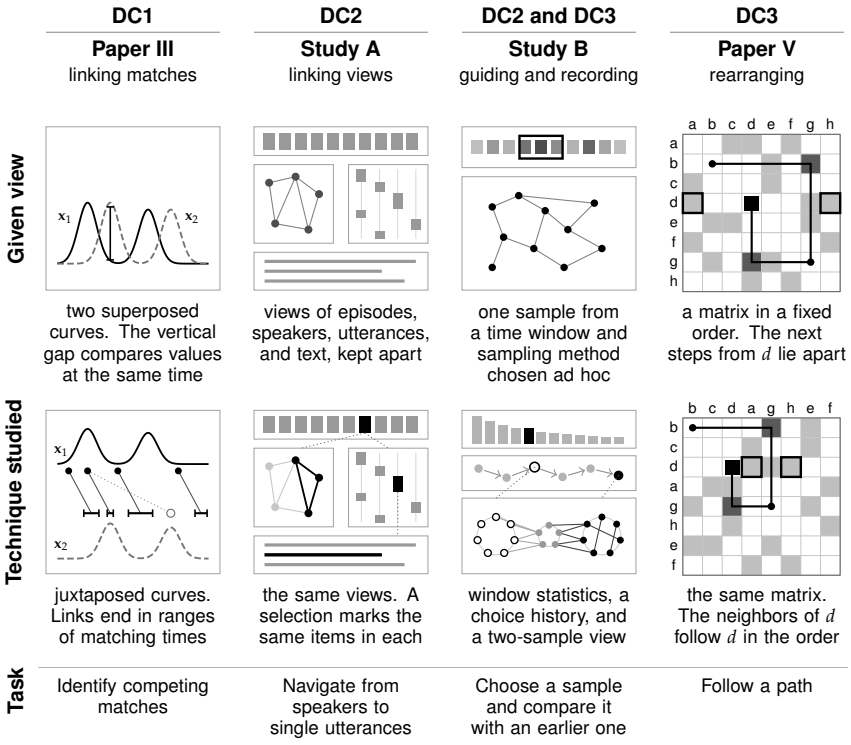


Figure 6.2: Four interactive visualization techniques studied for DC1 to DC3. Each column begins with the view or views taken as given and then shows the technique the study develops.

Across the computational methods and visualization techniques, a recurring move is to reopen a decision that a standard pipeline fixes too early. The appropriate stage for that decision is the stage that has the information needed to make it. Later stages may still need enough information to inspect its consequences.

The comparisons lead to five considerations for work on similar pipelines. Each is supported by the studies named with it and remains limited to their data and tasks.

1. *Check whether the computational result records a relationship before designing access to it.* An interface can rearrange, link, or filter what a result records. It cannot show a relationship that the result omits. Where the result already records the needed links, the problem lies in access, and a visualization or interaction technique is enough (Paper V). Where it does not, the computation has to change first (Paper III, and Papers I and II within one pipeline). With an inherited computation, the same check bounds what the interface can claim (Study B).

2. *When a method changes relationships on purpose, keep a reference to what it changed.* A learned graph, a sample, and a fitted alignment depart from their input by design, so a visible pattern can come from the data or from the method. A reference beside the result, or an explicit encoding of the difference, lets analysts tell the two apart (Paper III, Study B). An explicit difference is easier to read but shows the departure from only one reference. Papers I and II do not provide such a reference.

3. *Summarize plausible alternatives without collapsing them to one answer.* When a result keeps several weighted answers, showing all of them overloads the display, and showing only the strongest discards what the representation was chosen to record. A single summary position can also mislead when the weight splits between separate answers. Paper III, the only study that supports this consideration directly, pairs a representative match with its spread and its alternatives. The spread also depends on model parameters, so part of the ambiguity shown belongs to the model.

4. *Record the choice of part and scale, whether the computation or the analyst makes it.* A finding about a large structure holds for the part and the level of detail under inspection. When the computation sets the level, the setting makes the result reproducible (Paper II). When the analyst sets it, the choice can follow the question, but the display may no longer show which selection produced a finding (Studies A and B). Recording the selections ties each finding to the step that produced it and lets an analyst check whether a pattern persists across samples (Study B, Paper V).

5. *State which stages are inherited, and match the evidence to the stage that was changed.* Many visual analytics papers report little about the embeddings on which their views depend (Paper IV), and each study that works on visualization techniques in this thesis inherits some stages.

Naming them tells a reader which claims the evidence covers. Structural measures can show how a representation algorithm changes a graph or a layout (Papers I and II). Use cases and expert feedback can show how a visualization or interaction technique works in a specific analysis (Papers III and V, Studies A and B). Neither form of evidence supports claims about general analyst performance.

6.3 Future Work

This section describes directions for future work that follow from the studies in this thesis. The first direction concerns the theory behind MAPLE (Paper I) and SDMP (Paper II). MAPLE learns which observations are treated as neighbors, and SDMP builds a layout from the spectral components of a neighbor graph. Section 6.1 suggests a close relation between the two methods. However, this relation holds only in a special case, and future work could develop it formally. A spectral theory of dimensionality reduction for visual analysis could close this gap by treating learning the neighbors and building the layout as two operations on the same graph. Such a theory would also need to account for intrinsic dimension. MAPLE compresses each neighborhood into a few directions. A neighborhood on a two-dimensional surface needs two, but a neighborhood in a more complex region needs more. Manifold capacity theory, and possibly information theory, could relate this number to the intrinsic dimension of the data around each neighborhood.

A second direction concerns the layout itself. SDMP uses a fixed number of spectral components and leaves out the rest, so each component is either fully included or excluded. A spectral graph neural network could learn a filter over the whole spectrum and decide how strongly each component contributes. Furthermore, the same spectral view applies to layouts produced by other methods. Each layout coordinate is a signal over the observations, so any layout can be decomposed into the spectral components of the graph. The decomposition would show how much of a layout follows broad structure and how much follows fine structure. Layouts from different methods could then be compared component by component.

A third direction concerns the data that these methods receive. MAPLE and SDMP are general methods, designed without domain experts or a domain-specific pipeline. However, some fields already have such pipelines. In single-cell analysis, data pass through established steps, such as normalization and selection of highly variable genes, before dimensionality reduction. The effects of these steps are known, yet a general method treats their output like any other table of numbers. Thus, an open question is how to design domain-specific dimensionality reduction that builds on these steps and their known effects.

The last direction moves from the computational methods to the interactive techniques. In the interactive studies, analysts make decisions, for example, which of several competing matches to trust in LineAlign. Decision theory could describe how an analyst chooses among the options a view presents. Furthermore, provenance could record these choices, as VIS-DN already does for every selection. Adaptive interfaces could then use these records to respond to each analyst.

7 Conclusions

This thesis examined how structure-aware representation algorithms can support visual analytics of high-dimensional and relational data. RQ1 asked how a neighbor graph can be refined when distances in the input space mix structure with noise. RQ2 asked how a layout can reveal the structure of one graph at several scales. RQ3 broadened the view and asked how computational methods, visualization techniques, and interaction techniques are used to inspect structure in a visual analytical pipeline, and which design challenges noise and scale raise in such a pipeline.

A neighbor graph can be refined (RQ1) by learning the space in which neighbors are computed, allowing the graph to express relationships that noise or curved structure obscure in the input space (Paper I). Once the graph is built, a layout can reveal its structure at several scales (RQ2). The layout coordinates are treated as graph signals and built in an expanding spectral subspace, so structure appears progressively as finer variation is introduced (Paper II). Both answers change the computation. Whether analysts can inspect the resulting structure also depends on the visualization and interaction techniques applied to it (RQ3). A survey of embedding-based visual analytics characterizes the design space where these techniques combine with computational methods (Paper IV). Within that space, four studies address three design challenges raised by noise and scale (Papers III and V, Studies A and B). The challenges are inspecting competing results, choosing which part

of the data and which scale to show, and tracking the steps of an exploration. Together, the algorithms and techniques change several stages of a standard visual analysis pipeline, from the neighbor graph to the view, so analysts can carry out tasks the standard pipeline omits or makes hard, such as finding the level of detail at which a group appears.

Contributions. The thesis contributes two representation algorithms for neighbor embedding. In standard UMAP, the neighbor graph is fixed before the layout is optimized. The first algorithm treats these neighbors as tentative and rebuilds the graph in a space learned with a self-supervised objective, so the graph itself becomes a learned result (Paper I) [1]. The objective treats each neighborhood as a set of views, so it applies to tabular data such as gene expression, where earlier self-supervised dimensionality-reduction methods target images or compression. The second algorithm makes the level of detail a parameter of the layout (Paper II) [2]. Each stage adds spectral components of the neighbor graph, from broad to fine structure, and each position can be traced to the components that formed it. Existing explanations relate a layout to input features and do not show at which level of detail a group forms.

The thesis also contributes interactive visualization techniques for cases where a standard view commits to one result before the analyst can examine it. A single warping path keeps one match per time step, and a superimposed line chart can show timing differences as value differences. Visualizing a soft alignment, with a weight for every candidate match, shows where several matches are plausible and separates timing displacement from pace change and feature mismatch (Paper III) [3]. An adjacency matrix shows a path as separate cells, and reordering it around each selected node while drawing the route taken makes the path easier to follow (Paper V) [5]. A sample of a large network depends on settings chosen before the sample is seen. Linking the whole network's statistics to these settings, and recording every sample with its settings, lets analysts revise a sample and compare it with earlier ones (Study B) [7, 8]. Linked views from summaries of a dialogue collection down to single utterances keep the text behind each pattern within reach (Study A) [6].

A survey of visual analytics approaches that use embeddings shows that few of them change how their embeddings are constructed, so the structure an analyst can see is usually fixed before visual design begins (Paper IV) [4]. Comparing all studies yields five design considerations for similar pipelines (Section 6.2). First, an interface can reorder, link, or filter only what a computation records. A relationship missing from the computational result, such as a competing match between two time series, has to be added in the computation before any view can show it.

Scope of the Findings. The thesis does not attempt a general solution for representing high-dimensional data, time series, and networks. It uses noise and multiscale structure, two characteristics common to the relationships derived or obtained from these data, to show how computational representation and the interface together determine which structure an analyst can inspect.

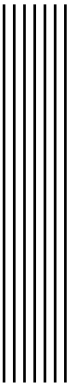
The kind of evidence also limits the claims. The two algorithms were evaluated quantitatively and through use cases (Papers I and II). The remaining studies propose and evaluate interactive visualization techniques through prototypes, usage scenarios, use cases, and limited expert feedback, and none includes a controlled user study. The studies show what the methods and techniques make possible in the cases examined. The thesis does not claim that analysts generally work more accurately, quickly, or reliably with these techniques than with alternatives.

Broader Relevance. The studies address practical problems across several application domains. In single-cell biology, a two-dimensional layout is often where analysts first propose cell types and developmental lineages. Decisions made before the layout appears, about which cells count as neighbors and how much of the neighbor graph's structure is shown, can determine whether a subtype or a transitional cell is visible at all (Papers I and II). Such decisions are usually treated as preprocessing. The two studies indicate that they belong to the analysis itself, because they shape the hypotheses that follow.

For visualization, the same studies suggest a different way to think about an embedding. A neighbor embedding is often read as a finished picture of the data. It can also be read as the result of two stages that can each be designed and examined, the construction of the neighbor graph and the layout of that graph across scales. Treating an embedding as two designed stages also affects evaluation. A quality measure that takes the input neighbors as correct cannot fairly judge a method whose purpose is to question those neighbors. In time-series alignment and network sampling, competing matches and sampling settings also shape what an analyst sees, and the surveyed literature reports such choices unevenly.

Closing. Many questions in visual data analysis are questions about relationships, and many of these relationships are hidden in the data. Noise, high dimensionality, and structure at several scales make them hard to see directly. Method designers and analysts have to decide which observations count as neighbors, at which scale structure is shown, which of several plausible matches holds, and which path, level of detail, or sample is worth following. In a standard visual analysis pipeline, many of these decisions sit

inside a computation or a fixed view that the analyst cannot change. Each study in this thesis takes one such decision and turns it into something an algorithm can learn or an analyst can see and change. As data grow in dimension and connectivity, what an analyst sees depends more on decisions like these, and visual analytics can benefit from designing the computational representation, visual encodings, and interaction together. The methods in this thesis often begin at points where an analytical pipeline appears already settled. The broader lesson is that new analytical possibilities do not always require a more sophisticated objective. They can emerge from recognizing which decision or assumption was fixed too early, and making that decision available for investigation again.



Bibliography

[1] Zeyang Huang, Takanori Fujiwara, Angelos Chatzimparmpas, Wandrille Duchemin, and Andreas Kerren. “MAPLE: Self-Supervised Learning-Enhanced Nonlinear Dimensionality Reduction for Visual Analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 32.7 (2026), pp. 7131–7146. DOI: 10.1109/TVCG.2026.3694000.

[2] Zeyang Huang, Angelos Chatzimparmpas, Thomas Höllt, and Takanori Fujiwara. “A Spectral Decomposition Framework for Multiscale Nonlinear Dimensionality Reduction”. Manuscript under revision for IEEE Transactions on Visualization and Computer Graphics. 2026. DOI: 10.48550/arXiv.2604.02535. arXiv: 2604.02535 [cs.LG].

[3] Zeyang Huang, Takanori Fujiwara, Niklas Rönnberg, and Katerina Vrotsou. “LineAlign: Comparing Time Series in Line Charts through Optimal Transport-Based Soft Correspondence”. Manuscript to be submitted to Computer Graphics Forum. 2026.

[4] Zeyang Huang, Daniel Witschard, Kostiantyn Kucher, and Andreas Kerren. “VA + Embeddings STAR: A State-of-the-Art Report on the Use of Embeddings in Visual Analytics”. In: *Computer Graphics Forum* 42.3 (2023), pp. 539–571. DOI: 10.1111/cgf.14859.

[5] Zeyang Huang, Daniel Archambault, Rita Borgo, and Andreas Kerren. “Matrix Snap&Go: Visualization of Paths on Matrices”. In: *EuroVis 2024 – Short Papers*. The Eurographics Association, 2024. DOI: 10.2312/evs.20241058.

[6] Zeyang Huang, Kostiantyn Kucher, and Andreas Kerren. “Towards a Visual Analytics System for Emotion Trajectories in Multiparty Conversations”. In: *EuroVis 2024 – Posters*. The Eurographics Association, 2024. DOI: 10.2312/evp.20241092.

[7] Zeyang Huang, Kostiantyn Kucher, and Andreas Kerren. “Towards an Exploratory Visual Analytics System for Multivariate Subnetworks in Social Media Analysis”. In: *Poster Abstracts, IEEE Visualization and Visual Analytics (VIS ’22)*. DiVA: diva2:1707880. 2022.

[8] Zeyang Huang, Kostiantyn Kucher, and Andreas Kerren. “VIS-DN: Visual Analytics for Interactive Sampling on Dynamic Multivariate Networks”. Unpublished Manuscript. Builds on the poster of Related Study B, 15 pages. 2024.

[9] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, eds. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999. URL: https://shop.elsevier.com/books/readings-in-information-visualization/card/978-0-08-051573-1.

[10] Laurens van der Maaten and Geoffrey Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: https://www.jmlr.org/papers/v9/vandermaaten08a.html.

[11] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2018. DOI: 10.48550/arXiv.1802.03426. arXiv: 1802.03426.

[12] Damien François, Vincent Wertz, and Michel Verleysen. “The Concentration of Fractional Distances”. In: *IEEE Transactions on Knowledge and Data Engineering* 19.7 (2007), pp. 873–886. DOI: 10.1109/TKDE.2007.1037.

[13] Hyeon Jeon, Hyunwook Lee, Yun-Hsin Kuo, Taehyun Yang, Daniel Archambault, Sungahn Ko, Takanori Fujiwara, Kwan-Liu Ma, and Jinwook Seo. “Unveiling High-Dimensional Backstage: A Survey for Reliable Visual Analytics with Dimensionality Reduction”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2025, pp. 1–24. ISBN: 9798400713941. DOI: 10.1145/3706598.3713551.

[14] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A. Keim. “Knowledge Generation Model for Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1604–1613. DOI: 10.1109/TVCG.2014.2346481.

[15] 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.

[16] Laurens van der Maaten, Eric Postma, and Jaap van den Herik. *Dimensionality Reduction: A Comparative Review*. Tech. rep. TiCC-TR 2009-005. Tilburg University, 2009. URL: https://lvdmaaten.github.io/publications/papers/TR_Dimensionality_Reduction_Review_2009.pdf.

[17] John P. Cunningham and Zoubin Ghahramani. “Linear Dimensionality Reduction: Survey, Insights, and Generalizations”. In: *Journal of Machine Learning Research* 16.89 (2015), pp. 2859–2900. URL: https://www.jmlr.org/papers/v16/cunningham15a.html.

[18] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. “A Global Geometric Framework for Nonlinear Dimensionality Reduction”. In: *Science* 290.5500 (2000), pp. 2319–2323. DOI: 10.1126/science.290.5500.2319.

[19] Mikhail Belkin and Partha Niyogi. “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation”. In: *Neural Computation* 15.6 (2003), pp. 1373–1396. DOI: 10.1162/089976603321780317.

[20] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W. H. Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W. Newell. “Dimensionality Reduction for Visualizing Single-Cell Data using UMAP”. In: *Nature Biotechnology* 37.1 (2019), pp. 38–44. DOI: 10.1038/nbt.4314.

[21] Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. “Attraction-Repulsion Spectrum in Neighbor Embeddings”. In: *Journal of Machine Learning Research* 23.95 (2022), pp. 1–32. URL: https://www.jmlr.org/papers/v23/21-0055.html.

[22] Sebastian Damrich and Fred A. Hamprecht. “On UMAP’s True Loss Function”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 5798–5809. URL: https://proceedings.neurips.cc/paper_files/paper/2021/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

[23] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. “Assessing Single-Cell Transcriptomic Variability through Density-Preserving Data Visualization”. In: *Nature Biotechnology* 39.6 (2021), pp. 765–774. DOI: 10.1038/s41587-020-00801-7.

[24] Lucy Xia, Christy Lee, and Jingyi Jessica Li. “Statistical Method scDEED for Detecting Dubious 2D Single-Cell Embeddings and Optimizing t-SNE and UMAP Hyperparameters”. In: *Nature Communications* 15.1 (2024), p. 1753. DOI: 10.1038/s41467-024-45891-y.

[25] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. “Self-Supervised Learning: Generative or Contrastive”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.1 (2023), pp. 857–876. DOI: 10.1109/TKDE.2021.3090866.

[26] Thomas Yerxa, Yilun Kuang, Eero Simoncelli, and SueYeon Chung. “Learning Efficient Coding of Natural Images with Maximum Manifold Capacity Representations”. In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 24103–24128. DOI: 10.52202/075280-1047.

[27] SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. “Classification and Geometry of General Perceptual Manifolds”. In: *Physical Review X* 8.3 (2018), p. 031003. DOI: 10.1103/PhysRevX.8.031003.

[28] Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. *Unsupervised Visualization of Image Datasets using Contrastive Learning*. 2022. DOI: 10.48550/arXiv.2210.09879. arXiv: 2210.09879.

[29] Yannis Kalantidis, Carlos Eduardo Rosar Kos Lassance, Jon Almazán, and Diane Larlus. “TLDR: Twin Learning for Dimensionality Reduction”. In: *Transactions on Machine Learning Research* (2022). URL: https://openreview.net/forum?id=86fhqdBUbx.

[30] Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. “Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization”. In: *Journal of Machine Learning Research* 22.201 (2021), pp. 1–73. URL: https://www.jmlr.org/papers/v22/20-1061.html.

[31] Dmitry Kobak and George C. Linderman. “Initialization Is Critical for Preserving Global Data Structure in Both t-SNE and UMAP”. In: *Nature Biotechnology* 39.2 (2021), pp. 156–157. DOI: 10.1038/s41587-020-00809-z.

[32] Vincent van Unen, Thomas Höllt, Nicola Pezzotti, Na Li, Marcel J. T. Reinders, Elmar Eisemann, Frits Koning, Anna Vilanova, and Boudewijn P. F. Lelieveldt. “Visual Analysis of Mass Cytometry Data by Hierarchical Stochastic Neighbour Embedding Reveals Rare Cell Types”. In: *Nature Communications* 8.1 (2017), p. 1740. DOI: 10.1038/s41467-017-01689-9.

[33] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. “The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains”. In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 83–98. DOI: 10.1109/MSP.2012.2235192.

[34] Ronald R. Coifman and Stéphane Lafon. “Diffusion Maps”. In: *Applied and Computational Harmonic Analysis* 21.1 (2006), pp. 5–30. DOI: 10.1016/j.acha.2006.04.006.

[35] Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. “Visualizing Structure and Transitions in High-Dimensional Biological Data”. In: *Nature Biotechnology* 37.12 (2019), pp. 1482–1492. DOI: 10.1038/s41587-019-0336-3.

[36] T. Tony Cai and Rong Ma. “Theoretical Foundations of t-SNE for Visualizing High-Dimensional Clustered Data”. In: *Journal of Machine Learning Research* 23.301 (2022), pp. 1–54. URL: https://www.jmlr.org/papers/v23/21-0524.html.

[37] Luis Gustavo Nonato and Michael Aupetit. “Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.8 (2019), pp. 2650–2673. DOI: 10.1109/TVCG.2018.2846735.

[38] Rebecca Faust, David Glickenstein, and Carlos Scheidegger. “DimReader: Axis Lines That Explain Non-Linear Projections”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 481–490. DOI: 10.1109/TVCG.2018.2865194.

[39] Brian Montambault, Gabriel Appleby, Jen Rogers, Camelia D. Brumar, Mingwei Li, and Remco Chang. “DimBridge: Interactive Explanation of Visual Patterns in Dimensionality Reductions with Predicate Logic”. In: *IEEE Transactions on Visualization and Computer Graphics* 31.1 (2025), pp. 207–217. DOI: 10.1109/TVCG.2024.3456391.

[40] Xiaofei He and Partha Niyogi. “Locality Preserving Projections”. In: *Advances in Neural Information Processing Systems*. Vol. 16. 2003. URL: https://proceedings.neurips.cc/paper_files/paper/2003/hash/d69116f8b0140cdeb1f99a4d5096ffe4-Abstract.html.

[41] Kerstin Bunte, Michael Biehl, and Barbara Hammer. “A General Framework for Dimensionality-Reducing Data Visualization Mapping”. In: *Neural Computation* 24.3 (2012), pp. 771–804. DOI: 10.1162/neco_a_00250.

[42] Edouard Couplet, Pierre Lambert, Michel Verleysen, Dounia Mulders, John A. Lee, and Cyril de Bodt. “Natively Interpretable t-SNE”. In: *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Vol. 2133. Communications in Computer and Information Science. Springer Nature Switzerland, 2025, pp. 107–123. DOI: 10.1007/978-3-031-74630-7_8.

[43] Yang Yang, Jialei Gong, Hongjian Sun, Amos Choo, Jessica Cara Mar, Yunbo Wei, Yu Zhang, Wenjie Zhang, Minglei Shu, Zewen Kelvin Tuong, and Di Yu. “Feature-Preserving Manifold Approximation and Projection to Analyze Single-Cell Data”. In: *Nature Computational Science* 6.5 (2026), pp. 478–496. DOI: 10.1038/s43588-026-00970-6.

[44] Jarkko Venna and Samuel Kaski. “Local Multidimensional Scaling”. In: *Neural Networks* 19.6-7 (2006), pp. 889–899. DOI: 10.1016/j.neunet.2006.05.014.

[45] John A. Lee and Michel Verleysen. “Quality Assessment of Dimensionality Reduction: Rank-Based Criteria”. In: *Neurocomputing* 72.7–9 (2009), pp. 1431–1443. DOI: 10.1016/j.neucom.2008.12.017.

[46] Joseph B. Kruskal. “Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis”. In: *Psychometrika* 29.1 (1964), pp. 1–27. DOI: 10.1007/BF02289565.

[47] Fernando V. Paulovich, Luis Gustavo Nonato, Rosane Minghim, and Haim Levkowitz. “Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and Its Application to Document Mapping”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.3 (2008), pp. 564–575. DOI: 10.1109/TVCG.2007.70443.

[48] Ji Soo Yi, Youn-ah Kang, John T. Stasko, and Julie A. Jacko. “Toward a Deeper Understanding of the Role of Interaction in Information Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1224–1231. DOI: 10.1109/TVCG.2007.70515.

[49] Yixuan Zhang, Sara Di Bartolomeo, Fangfang Sheng, Holly Jimison, and Cody Dunne. “Evaluating Alignment Approaches in Superimposed Time-Series and Temporal Event-Sequence Visualizations”. In: *2019 IEEE Visualization Conference (VIS)*. 2019, pp. 1–5. DOI: 10.1109/VISUAL.2019.8933584.

[50] Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, and Anastasia Bezerianos. “Comparing Similarity Perception in Time Series Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 523–533. DOI: 10.1109/TVCG.2018.2865077.

[51] Meinard Müller. “Dynamic Time Warping”. In: *Information Retrieval for Music and Motion*. Springer, 2007, pp. 69–84. DOI: 10.1007/978-3-540-74048-3_4.

[52] Marco Cuturi and Mathieu Blondel. “Soft-DTW: A Differentiable Loss Function for Time-Series”. In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 894–903. URL: https://proceedings.mlr.press/v70/cuturi17a.html.

[53] Marco Cuturi. “Sinkhorn Distances: Lightspeed Computation of Optimal Transport”. In: *Advances in Neural Information Processing Systems*. Vol. 26. 2013. URL: https://proceedings.neurips.cc/paper_files/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.

[54] Bing Su and Gang Hua. “Order-Preserving Wasserstein Distance for Sequence Matching”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2906–2914. DOI: 10.1109/CVPR.2017.310.

[55] Jessica Hullman, Xiaoli Qiao, Michael Correll, Alex Kale, and Matthew Kay. “In Pursuit of Error: A Survey of Uncertainty Visualization Evaluation”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 903–913. DOI: 10.1109/TVCG.2018.2864889.

[56] Theresia Gschwandtner, Markus Bögl, Paolo Federico, and Silvia Miksch. “Visual Encodings of Temporal Uncertainty: A Comparative User Study”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 539–548. DOI: 10.1109/TVCG.2015.2467752.

[57] Ben Shneiderman. “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations”. In: *Proceedings of the IEEE Symposium on Visual Languages*. 1996, pp. 336–343. DOI: 10.1109/VL.1996.545307.

[58] Jonathan C. Roberts. “State of the Art: Coordinated & Multiple Views in Exploratory Visualization”. In: *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV)*. 2007, pp. 61–71. DOI: 10.1109/CMV.2007.20.

[59] 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. DOI: 10.2312/eurovisstar.20151110.

[60] 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*. ACM, 2016, pp. 855–864. DOI: 10.1145/2939672.2939754.

[61] Pili Hu and Wing Cheong Lau. *A Survey and Taxonomy of Graph Sampling*. 2013. DOI: 10.48550/arXiv.1308.5865. arXiv: 1308.5865.

[62] Gaëlle Richer, Alexis Pister, Moataz Abdelaal, Jean-Daniel Fekete, Michael Sedlmair, and Daniel Weiskopf. “Scalability in Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 30.7 (2024), pp. 3314–3330. DOI: 10.1109/TVCG.2022.3231230.

[63] Bharat Kale, Maoyuan Sun, and Michael E. Papka. “The State of the Art in Visualizing Dynamic Multivariate Networks”. In: *Computer Graphics Forum* 42.3 (2023), pp. 471–490. DOI: 10.1111/cgf.14856.

[64] Manuel Freire, Catherine Plaisant, Ben Shneiderman, and Jen Golbeck. “ManyNets: An Interface for Multiple Network Analysis and Visualization”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 213–222. DOI: 10.1145/1753326.1753358.

[65] Adam Perer and Ben Shneiderman. “Balancing Systematic and Flexible Exploration of Social Networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006), pp. 693–700. DOI: 10.1109/TVCG.2006.122.

[66] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. “On the Readability of Graphs Using Node-Link and Matrix-Based Representations: A Controlled Experiment and Statistical Analysis”. In: *Information Visualization* 4.2 (2005), pp. 114–135. DOI: 10.1057/palgrave.ivs.9500092.

[67] 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* 35.3 (2016), pp. 693–716. DOI: 10.1111/cgf.12935.

[68] Michael Behrisch, Tobias Schreck, and Hanspeter Pfister. “GUIRO: User-Guided Matrix Reordering”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 184–194. DOI: 10.1109/TVCG.2019.2934300.

[69] Nathalie Henry and Jean-Daniel Fekete. “MatLink: Enhanced Matrix Visualization for Analyzing Social Networks”. In: *Human-Computer Interaction – INTERACT 2007*. Springer Berlin Heidelberg, 2007, pp. 288–302. ISBN: 978-3-540-74800-7. DOI: 10.1007/978-3-540-74800-7_24.

[70] Zeqian Shen and Kwan-Liu Ma. “Path Visualization for Adjacency Matrices”. In: *Eurographics/IEEE-VGTC Symposium on Visualization*. The Eurographics Association, 2007. DOI: 10.2312/VisSym/EuroVis07/083-090.

[71] Tomer Moscovich, Fanny Chevalier, Nathalie Henry, Emmanuel Pietriga, and Jean-Daniel Fekete. “Topology-Aware Navigation in Large Networks”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’09. Association for Computing Machinery, 2009, pp. 2319–2328. ISBN: 9781605582467. DOI: 10.1145/1518701.1519056.

[72] Eric D. Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. “Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 31–40. DOI: 10.1109/TVCG.2015.2467551.

[73] 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* 22.1 (2016), pp. 559–568. DOI: 10.1109/TVCG.2015.2467851.

[74] Sehi L’Yi, Jaemin Jo, and Jinwook Seo. “Comparative Layouts Revisited: Design Space, Guidelines, and Future Directions”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 1525–1535. DOI: 10.1109/TVCG.2020.3030419.

[75] Carolina Nobre, Miriah Meyer, Marc Streit, and Alexander Lex. “The State of the Art in Visualizing Multivariate Networks”. In: *Computer Graphics Forum* 38.3 (2019), pp. 807–832. DOI: 10.1111/cgf.13728.

[76] Jock Mackinlay. “Automating the Design of Graphical Presentations of Relational Information”. In: *ACM Transactions on Graphics* 5.2 (1986), pp. 110–141. DOI: 10.1145/22949.22950.

[77] William S. Cleveland and Robert McGill. “Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods”. In: *Journal of the American Statistical Association* 79.387 (1984), pp. 531–554. DOI: 10.1080/01621459.1984.10478080.

[78] Tamara Munzner. *Visualization Analysis and Design*. A K Peters/CRC Press, 2014. DOI: 10.1201/b17511.

[79] Heidi Lam. “A Framework of Interaction Costs in Information Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1149–1156. DOI: 10.1109/TVCG.2008.109.

[80] 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 (BELIV)*. 2006, pp. 1–5. DOI: 10.1145/1168149.1168168.

[81] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983. URL: https://search.worldcat.org/title/1023872477.

[82] Madeleine Udell and Alex Townsend. “Why Are Big Data Matrices Approximately Low Rank?” In: *SIAM Journal on Mathematics of Data Science* 1.1 (2019), pp. 144–160. DOI: 10.1137/18M1183480.

[83] Vincent Thibeault, Antoine Allard, and Patrick Desrosiers. “The Low-Rank Hypothesis of Complex Systems”. In: *Nature Physics* 20.2 (2024), pp. 294–302. DOI: 10.1038/s41567-023-02303-0.

[84] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. “Design Study Methodology: Reflections from the Trenches and the Stacks”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2431–2440. DOI: 10.1109/TVCG.2012.213.

[85] Herbert A. Simon. *The Sciences of the Artificial*. 3rd ed. MIT Press, 1996. URL: https://mitpress.mit.edu/9780262264495/the-sciences-of-the-artificial/.

[86] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. “Design Science in Information Systems Research”. In: *MIS Quarterly* 28.1 (2004), pp. 75–105. DOI: 10.2307/25148625.

[87] John Dewey. *Logic: The Theory of Inquiry*. Henry Holt and Company, 1938. URL: https://books.google.com/books?id=3m0aAAAAIAAJ.

[88] David L. Morgan. “Pragmatism as a Paradigm for Social Research”. In: *Qualitative Inquiry* 20.8 (2014), pp. 1045–1053. DOI: 10.1177/1077800413513733.

[89] Tamara Munzner. “A Nested Model for Visualization Design and Validation”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 921–928. DOI: 10.1109/TVCG.2009.111.

[90] Guillaume Huguet, Alexander Tong, Edward De Brouwer, Yanlei Zhang, Guy Wolf, Ian Adelstein, and Smita Krishnaswamy. “A Heat Diffusion Perspective on Geodesic Preserving Dimensionality Reduction”. In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 6986–7016. DOI: 10.52202/075280-0306.

[91] Fintan McGee, Mohammad Ghoniem, Guy Melançon, Benoît Otjacques, and Bruno Pinaud. “The State of the Art in Multilayer Network Visualization”. In: *Computer Graphics Forum* 38.6 (2019), pp. 125–149. DOI: 10.1111/cgf.13610.

[92] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. “A Taxonomy and Survey of Dynamic Graph Visualization”. In: *Computer Graphics Forum* 36.1 (2017), pp. 133–159. DOI: 10.1111/cgf.12791.

[93] Robert Amar, James Eagan, and John Stasko. “Low-Level Components of Analytic Activity in Information Visualization”. In: *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS)*. 2005, pp. 111–117. DOI: 10.1109/INFVIS.2005.1532136.

[94] Parisa Salmanian, Angelos Chatzimparmpas, Ali Can Karaca, and Rafael M. Martins. “DimVis: Interpreting Visual Clusters in Dimensionality Reduction with Explainable Boosting Machine”. In: *Machine Learning Methods in Visualisation for Big Data*. 2024. ISBN: 978-3-03868-256-1. DOI: 10.2312/mlvis.20241125.

[95] Heidi Lam, Melanie Tory, and Tamara Munzner. “Bridging from Goals to Tasks with Design Study Analysis Reports”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 435–445. DOI: 10.1109/TVCG.2017.2744319.

[96] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1597–1607. URL: https://proceedings.mlr.press/v119/chen20j.html.

[97] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. “Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 21271–21284. URL: https://proceedings.neurips.cc/paper_files/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

[98] Ulrike von Luxburg. “A Tutorial on Spectral Clustering”. In: *Statistics and Computing* 17.4 (2007), pp. 395–416. DOI: 10.1007/s11222-007-9033-z.

Papers

The papers associated with this thesis have been removed for copyright reasons. For more details about these see:

https://doi.org/10.3384/9789181187601

FACULTY OF SCIENCE AND ENGINEERING

Linköping Studies in Science and Technology. Dissertation No. 2557, 2026
Department of Science and Technology

Linköping University
SE-581 83 Linköping, Sweden

www.liu.se

LINKÖPING UNIVERSITY