Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät

Masterthesis Machine Learning

kNN-Graph coarsening from the perspective of Dimensionality reduction

Moritz Christ

14.04.2025

Reviewers

Dr. Dmitry Kobak
(Machine Learning)
Hertie Institute for AI in Brain Health
Universität Tübingen

Prof. Dr. Philipp Berens
(Machine Learning)
Hertie Institute for AI in Brain Health
Universität Tübingen

knn-Graph coarsening *from the perspective of* Dimensionality Reduction

Moritz Christ
5477647

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Machine Learning

from the
October 15, 2024
to the
August 20, 2025

submitted to the
Wilhelm-Schickard Institute
Faculty of Science
Eberhard Karls University of Tübingen

Declaration of authorship

I hereby declare that I have written this master's thesis independently and only with the specified resources and that all passages taken from other works, whether verbatim or in spirit, have been identified as quotations with appropriate source references.
This master's thesis has not been submitted in the same or a similar form as an examination performance in any other degree program.
I have used AI for spelling and grammar corrections without any relevant text generation or translations. In other words, I have had texts written by me corrected in the same language.
These are purely linguistic corrections, so the meaning I originally intended was not significantly changed or expanded.
I have used AI to support the writing of code in software development. This is merely support and not the automatic generation of larger program parts.
All programs used and their version numbers are listed in a table in the appendix of my thesis.
I am aware that a breach of this declaration may have consequences under examination law and, in particular, may result in the examination being assessed as "insufficient" or the coursework being assessed as "failed" and, in the event of multiple or serious attempts to cheat, may result in de-registration or the initiation of proceedings for the withdrawal of any academic title awarded.

Moritz Christ, Tübingen, August 20, 2025

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Dmitry Kobak, for his support, insightful discussions, reviewing my thesis, and the many hours he generously dedicated to guiding this thesis. Special thanks also go to Prof. Dr. Philipp Berens for the opportunity to carry out this work in his research group and for kindly reviewing the final thesis.

I am grateful to Rita González Márquez for introducing me to the CIN servers and to the entire embedding subgroup for their helpful feedback during our weekly meetings. I would like to thank the whole research group that made my stay a pleasant one.

To my fellow Master's students and friends: heartfelt thanks go to Eileen, whose careful revision of this thesis made its completion possible. Jonathan, thank you for your willingness to help and your well-timed distractions whenever I got too caught up in my own head. Max, despite your mysterious relocation to Rottenburg, you still managed to be somewhat part of the journey. And to Annika and Elisa, who wrote their theses alongside me: thank you for making the process feel a little less lonely and similar overwhelming.

Finally, I want to thank my family for their constant encouragement—especially my dad, Ansgar, who not only revised parts of the thesis but also provided ideas, feedback, and an outlet for frustration over bugs that only occurred once-in-a-million. And finally, a special thanks to myself, for getting my shit together.

Abstract

The coarse-graining of k-nearest neighbor (kNN) graphs presents a fundamental challenge in high-dimensional data analysis, requiring a balance between structural simplification and the preservation of meaningful relationships. This thesis presents a comparative study of prominent graph coarsening techniques, focusing on their capacity to retain global and local structure across multiple abstraction levels. The methods investigated include landmark-based strategies (State-to-Landmark/HSNE, Landmark-to-Landmark), community detection–based aggregation (Walktrap, Leiden, Label Propagation), and spectral approaches (Kron Reduction, Local Variation).

Evaluation is conducted across a range of datasets, including image, text, and single-cell transcriptomic data, using both embedding-oriented and graph-theoretic metrics. The analysis highlights method-specific trade-offs in scalability, spectral fidelity, cluster preservation, and embedding quality. Particular emphasis is placed on the effects of graph directionality in random-walk–based models and the limitations of existing evaluation frameworks.

The results indicate that no single method consistently outperforms others across all criteria. Landmark-based approaches demonstrate superior performance in preserving global geometry but incur higher computational cost. Community detection methods offer hierarchical interpretability and robustness at lower complexity. Existing metrics, including centrality-based divergences and embedding cluster scores, are found to exhibit notable inconsistencies. Consequently, the thesis identifies a need for more principled evaluation strategies, including manifold-aware metrics and comparative tools for graphs of differing scales.

The findings establish a framework for selecting graph coarsening techniques in manifold learning and dimensionality reduction, while outlining directions for methodological refinement and research.

Contents

List of Figures

List of Tables

List of Abbreviations

Acc	Accuracy
rel. eigenerr	Relative Eigenvalue Error
DBI	Davies-Bouldin Index
KL	Kullback-Leibler
k**NN**	k-Nearest Neighbor
LP	Label Propagation
LL	Landmark-to-Landmark
LV	Local Variation
PCA	Principal Component Analysis
PSD	Positive-Semidefinit
SL	State-to-Landmark
TSNE	t-Distributed Stochastic Neighborhood Embeeding

1. Introduction

In Germany, there is a saying: *"Du siehst den Wald vor lauter Bäumen nicht"*, which translates to *"You can't see the forest for all the trees"*. It captures a familiar challenge: when overwhelmed by detail, it becomes difficult to perceive the underlying relationship of objects. In response to such complexity, humans have long developed ways to organize, categorize, and abstract.

One of the most powerful abstractions for representing complex relationships are graphs [1]. Graphs are mathematical structures that encode connections between entities and are used extensively across a wide range of domains. For example, a city's public transportation system can be modeled as a graph, where stations are nodes and routes are edges [2]. A family tree is another example, illustrating genealogical relationships in graph form [3]. Even social interactions can be represented as a network, where individuals are nodes and social ties form the edges between them [4]. In this way, graphs offer a natural and flexible framework for describing the structure of interconnected systems.

However, the same flexibility that makes graphs so useful also makes them challenging to interpret when they grow large. A graph representing social interactions in a town may already involve thousands of individuals and tens of thousands of connections. As graphs grow, it becomes increasingly difficult to extract meaningful insights from them. What begins as an attempt to make structure visible can quickly devolve into *"trees"* of connections, where no pattern stands out.

This thesis explores strategies for simplifying such graphs through the process of graph coarsening. We begin by constructing k-Nearest Neighbor (kNN) graphs from high-dimensional datasets, a common way to impose structure on unordered data [5, 6]. We then examine several approaches for reducing the size and complexity of these graphs while aiming to preserve their relational properties. The methods used for graph coarsening can be broadly categorized into two groups. First, we explore node aggregation approaches, in which groups of nodes are fused into supernodes based on their connectivity [7]. Second, we investigate landmarking approaches, which select a set of representative landmarks and connect them via random walks, offering another perspective on structural simplification [8]. Both strategies aim to reduce graph complexity while maintaining key topological features of the original data.

After coarsening, we visualize the resulting graphs using t-distributed Stochastic Neighbor Embedding (t-SNE), a widely-used nonlinear dimensional-

ity reduction technique [9]. This visualization step allows us to assess whether the coarsened graphs retain the essential structure of the original data and to what extent different approaches succeed in capturing meaningful global and local patterns.

The thesis is structured as follows. The first chapter introduces the necessary theoretical background, including graph theory, kNN-graphs, spectral graph theory, Markov chains, and clustering techniques. In the methods chapter, we describe the graph coarsening strategies under investigation and introduce the datasets and evaluation metrics used throughout the work. The results chapter presents our findings, followed by a discussion that interprets these outcomes in light of the underlying theory and practical implications. We conclude with a summary of our observations and offer an outlook on potential future directions in the field of graph simplification and visualization.

2. Background

This chapter provides foundational concepts. Section 2.1 introduces fundamental graph theory concepts, including notation and terminology that will be used in Chapter 3 for a clear understanding of the methods. A formal definition of the k-Nearest Neighbor (kNN) graph follows in Section 2.1.2.
In Section 2.1.3 spectral graph theory links linear algebra to graphs. Section 2.2 revisits dimensionality reduction, beginning with an analysis of Principal Component Analysis (PCA) and its reinterpretation of the coordinate system. It is advantageous to use PCA as an initialization for t-distributed Stochastic Neighbor Embedding (t-SNE) in, which is introduced in Section 2.2.1.
Section 2.3 lists cluster algorithms which are part of the subdivision for graph coarsening methods in Section 2.4.

2.1. Graph theory

2.1.1. Basics

Graphs can represent identical relational structures while appearing markedly different in visual form [1]. For example, Figure 2.1 presents the same graph rendered in two distinct configurations: the node positions have been rearranged, and the color assignments altered. Despite these changes, the underlying graph topology remains unchanged. This visual variability highlights the potential for misinterpretation in the absence of a formal framework. Consequently, clear and consistent notation is essential to ensure that graphs are correctly understood, regardless of their visual representation.

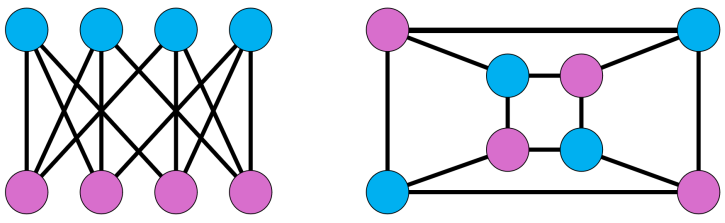

Figure 2.1.: An illustration of graph isomorphism, where two graphs share the same connectivity despite having different layouts

A **graph** G is defined as a tuple (V, E), where V is the set

of vertices, $V = \{v_1, \ldots, v_N\}$, and E is the set of edges, $E = \{(v_1, v_2, w_{1,2}), \ldots, (v_{N-1}, v_N, w_{(N-1,N)})\}$. This definition corresponds to a **directed graph** in which the triple $(v_i, v_j, w_{i,j})$ is the directed edge from v_i to v_j with weight $w_{i,j}$. If, for every edge $(v_i, v_j, w_{i,j})$, there exists an edge $(v_j, v_i, w_{j,i})$ and $w_{i,j} = w_{j,i}$, the graph is **undirected**. If $w_{i,j} \neq w_{i,j}$ but both are greater zero, it is called a bidirectional graph [1].

Graphs can be represented using an **weight matrix W** of size $N \times N$, where

$$W_{ij} = \begin{cases} w_{i,j}, & \text{if there is an edge between } v_i \text{ and } v_j, \\ 0, & \text{otherwise.} \end{cases}$$

If no edges have a weight, the graph is called unweighted and the presence of edges are indicated in a adjacency matrix **A**. The **degree** of a vertex quantifies its connectivity. The **indegree** d_i of a vertex v_i is defined as

$$d_i = \sum_j W_{ji}$$

and is sometimes represented as a **degree matrix**, a diagonal matrix with the degrees $d_1, \ldots, d_N$ on the diagonal. Conversely, the **out-degree** is the sum of all outgoing edges. A **stochastic graph** is one in which the out-degree of each vertex sums to one, allowing its interpretation as a Markov chain [10].

Graphs can be explored using various traversal strategies. A **random walk** follows transition probabilities to explore a graph, where each step moves to a neighboring vertex with a probability proportional to edge weights. However, random walks can be inefficient due to repeated visits to the same vertices.

A fundamental concept in graph analysis is **connectivity**. An graph is **connected** if there exists a path between every pair of vertices. A **component** is a maximal set of connected vertices that is not part of a larger connected subgraph. The **geodesic distance**, also known as the **number of hops**, refers to the shortest path between two vertices in terms of the number of edges traversed. The concept of distance and connectivity plays a crucial role in the kNN-graphs in the next section.

2.1.2. kNN-graphs

A **kNN-graph** is a graph in which each vertex v_i is connected to its k nearest neighbors [11]. These neighbors are determined according to a distance function, typically the Euclidean distance. Due to varying local densities, the neighborhood relation is generally not symmetric, i.e., v_i may be among the k nearest neighbors of v_j, but not vice versa.

We distinguish between three commonly used versions of kNN-graphs:

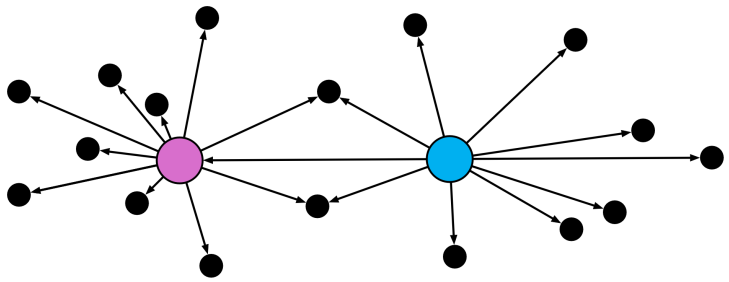

Figure 2.2.: Asymmetric relationships in a kNN-graph due to varying densities

1. **Directed kNN-graph**: Each vertex has exactly k outgoing edges but possibly varying in-degrees.
2. **Undirected kNN-graph**: The adjacency matrix is symmetrized, e.g., by setting $W_{ij} = W_{ji} = 1$ if either $(v_i, v_j) \in E$ or $(v_j, v_i) \in E$.
3. **Mutual kNN-graph**: Only edges for which both v_i is a neighbor of v_j and v_j is a neighbor of v_i are retained.

2.1.3. Spectral Graph Theory

The spectral analysis of graphs is a powerful tool for understanding their structure. Since spectral graph theory requires symmetric matrices, it naturally applies to undirected and mutual k-nearest neighbor graphs, while directed graphs are generally excluded [1, 5]. Spectral graph theory bridges graph-theoretical concepts with linear algebra by studying graphs through eigenvalues and eigenvectors of associated matrices.

Graph Laplacian:

$$\mathbf{L} := \mathbf{D} - \mathbf{W},$$

The fundamental object in spectral graph theory is the **graph Laplacian L**, defined for undirected graphs. **D** is the degree matrix and **W** the symmetric weight matrix of the graph. The Laplacian has two key features [12, 13]:

- **Symmetry**
  Since $\mathbf{D} = \mathbf{D}^\top$ and $\mathbf{W} = \mathbf{W}^\top$, it holds that $\mathbf{L} = \mathbf{L}^\top$.

$$\mathbf{L} = \mathbf{D} - \mathbf{W} = \mathbf{D}^\top - \mathbf{W}^\top = \mathbf{L}^\top. \tag{2.1}$$

- **Positive Semi-Definiteness (PSD)**

For any vector $\mathbf{f} \in \mathbb{R}^N$, the quadratic form $\mathbf{f}^\top \mathbf{L}\mathbf{f} \geq 0$.

$$\begin{aligned} \mathbf{f}^\top \mathbf{L}\mathbf{f} &= \mathbf{f}^\top \mathbf{D}\mathbf{f} - \mathbf{f}^\top \mathbf{W}\mathbf{f} \\ &= \sum_i d_i f_i^2 - \sum_{i,j} W_{ij} f_i f_j \\ &= \frac{1}{2} \sum_{i,j} W_{ij} (f_i - f_j)^2 \geq 0, \end{aligned} \tag{2.2}$$

Eigenvalues and connected components. The eigenvalues of $\mathbf{L}$ are real, non-negative, and can be ordered as:

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N.$$

The multiplicity[1] of the eigenvalue $\lambda = 0$ is directly related to the number of connected components of the graph [5].

Proof of the relation between eigenvalue multiplicity and connected components. First, consider a connected graph, i.e., a graph with a single connected component. Suppose $\mathbf{f}$ is an eigenvector satisfying $\mathbf{L}\mathbf{f} = 0$, which implies:

$$\mathbf{f}^\top \mathbf{L}\mathbf{f} = \frac{1}{2} \sum_{i,j} W_{ij} (f_i - f_j)^2 = 0.$$

Since all $W_{ij} \geq 0$, this is only possible if $f_i = f_j$ for all pairs (i, j) with $W_{ij} > 0$. The connectedness of the graph ensures that a path exists between any two vertices, so this constraint forces $\mathbf{f}$ to be constant on all vertices. Thus, the eigenvector associated with $\lambda_1 = 0$ is proportional to the constant vector $\mathbb{1}$.

In the case of a graph with k connected components, the Laplacian matrix can be permuted into a block-diagonal form:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{L}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{L}_k \end{bmatrix},$$

where each block $\mathbf{L}_\ell$ is the Laplacian of the ℓ-th connected component.

For each block $\mathbf{L}_\ell$, the constant vector on that component gives an eigenvector associated with the eigenvalue 0. Extending these to the full graph by padding with zeros outside the respective component yields k linearly independent eigenvectors. Hence, there are at least k eigenvalues equal to zero.

[1] In this context, the multiplicity is defined as the number of times zero appears as an eigenvalue

Finally, suppose there existed a $(k+1)$-th linearly independent eigenvector associated with $\lambda = 0$. This would require a non-trivial combination of the k indicator vectors of the components, which is impossible since their supports (the non-zero entries) are disjoint. Therefore, the multiplicity of $\lambda = 0$ is exactly k.

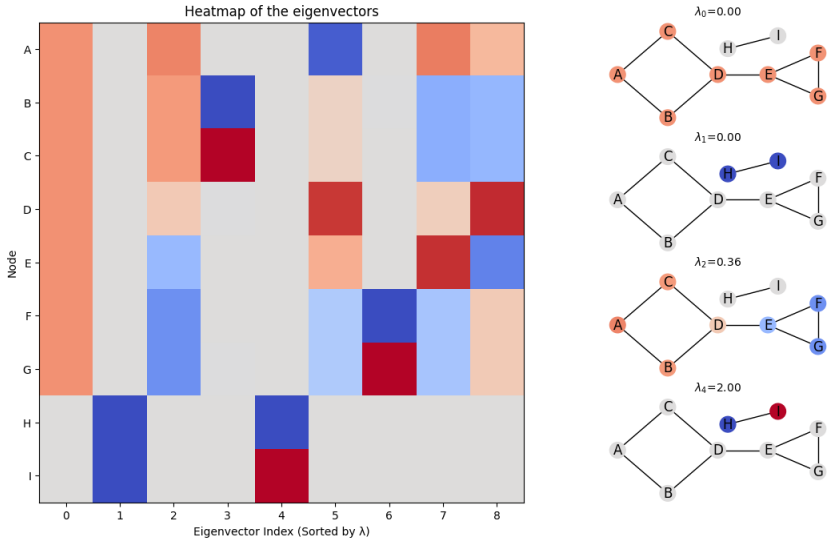

Figure 2.3.: Visualization of eigenvectors of a graph. Left: Heatmap of the eigenvectors. The first two eigenvectors are constant on the respective connected components, illustrating the multiplicity of $\lambda = 0$. Right: Graph with nodes colored according to eigenvector values, illustrating the global structure contained.

Interpretation of eigenvectors. The eigenvectors associated with the smallest eigenvalues (low-frequency eigenvectors) reveal global structures of the graph. In particular, the first k eigenvectors serve as indicators for the connected components. Eigenvectors associated with larger eigenvalues (high-frequency eigenvectors) tend to capture local variations and finer graph structures. An example can be seen in Figure 2.3 in which the zero has a multiplicity of two because of the two connected components. The corresponding two eigenvectors capture the affiliation to the corresponding connected component. The eigenvectors associated with bigger eigenvalues correspond to graph partitions, in particular the first eigenvector of a connected component is called the fiedler vector and often used for graph partitions.[14][15]

Normalized Laplacians. Besides the unnormalized Laplacian $\mathbf{L}$, two commonly used variants are the **normalized Laplacians**:

$$\mathbf{L}_{sym} := \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}, \quad \mathbf{L}_{rw} := \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}.$$

$\mathbf{L}_{sym}$ is also called the normalized symmetric Laplacian, while the $\mathbf{L}_{rw}$ is called the random walk Laplacian. These matrices frequently arise in applications and share crucial spectral properties of L, such as positive semidefiniteness.

2.1.4. Markov Chains and Random Walks

Random walks naturally link graph theory to Markov chain theory, especially when considering the random walk Laplacian. Any graph can be interpreted as a discrete-time Markov chain, provided that its weight matrix is positive. This perspective allows to study random walks on graphs using the theory of Markov chains.

From graphs to Markov chains. In the context of random walks, vertices are interpreted as *states*, and edges represent possible transitions. A probabilistic view is obtained by normalizing the weight matrix $\mathbf{W}$ row-wise (i.e., performing an ℓ_1-normalization) to ensure that the transition probabilities sum to one. For vertices without outgoing edges (dangling nodes), a self-loop is added to maintain stochasticity. This process leads to the **transition matrix** $\mathbf{P}$ defined as:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W},$$

where $\mathbf{D}$ is the degree matrix. This construction corresponds to the **random walk Laplacian** [5]. It is important to note that after normalization, the resulting Markov chain is generally *directed*, even if the original graph was undirected.

Simulating random walks. A random walk starting from vertex i and lasting n steps can be simulated by iteratively applying $\mathbf{P}$ to the initial distribution $\mathbf{e}_i$, the canonical basis vector corresponding to vertex i:

$$\pi^{(n)} = \mathbf{P}^n\mathbf{e}_i.$$

The resulting vector $\pi^{(n)}$ describes the distribution over states after n steps. Sampling from this distribution is equivalent to simulating the random walk.

If the distribution $\pi^{(n)}$ converges as $n \to \infty$, the resulting vector is called a **stationary distribution** π satisfying:

$$\pi^\top\mathbf{P} = \pi^\top.$$

However, a stationary distribution does not always exist. Certain Markov chains lack convergence due to their structure, as can be seen in Figure 2.4. This happens rarely when absorbing states are present.

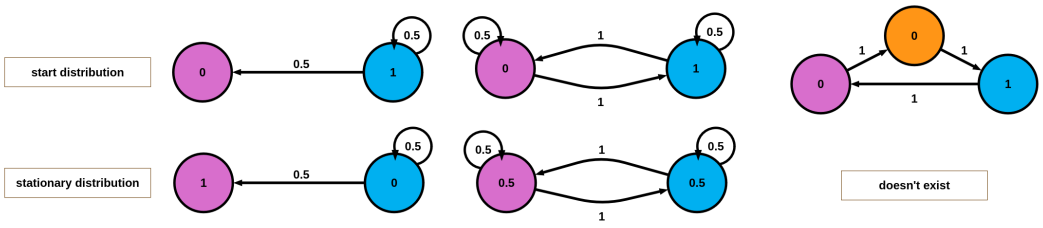

Figure 2.4.: Example of Markov chains with and without stationary distribution

Absorbing Markov chains. A notable special case is the **absorbing Markov chain** [16, 17]. In such chains, some states are designated as *absorbing* states, meaning that once entered, they cannot be left. All other states are classified as *transient*, from every transient state, it is possible to reach at least one absorbing state. The simplest example of such a graph is one with a starting state with a self-loop and an outgoing edge leading to an absorbing state with its own self-loop. Over time, the probability mass in the starting state diminishes, eventually converging to zero, while the absorbing state captures all probability mass.

Fundamental matrix for absorption probabilities. Absorbing Markov chains admit a convenient block structure by reordering the transition matrix $\mathbf{P}$ so that absorbing states come first:

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix}, \tag{2.3}$$

where:

- $\mathbf{Q} \in \mathbb{R}^{n_t \times n_t}$ contains the transition probabilities between transient states.
- $\mathbf{R} \in \mathbb{R}^{n_t \times n_a}$ contains the transition probabilities from transient states to absorbing states.
- $\mathbf{I} \in \mathbb{R}^{n_a \times n_a}$ is the identity matrix, representing the self-absorption for each absorbing state.

A key object for analyzing such chains is the **fundamental matrix B**:

$$\mathbf{B} := (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R},$$

$\mathbf{B}_{ij}$ contains the probability that a random walk starting from transient state i is eventually absorbed by absorbing state j. Another interpretation is that it provides the probabilities of absorption from each transient state to each absorbing state. This matrix plays a central role in various classical problems, such as the gambler's ruin [18], and will be used as a key component for Landmark connecting for directed graphs in Section 3.2.1.

Invertibility of $\mathbf{I} - \mathbf{Q}$. The matrix $\mathbf{I} - \mathbf{Q}$ is invertible, and this fact can be explicitly justified. Since all states governed by $\mathbf{Q}$ are transient, there is always a non-zero probability of eventually leaving the transient states. This implies that the powers of $\mathbf{Q}$ vanish as $k \to \infty$:

$$\lim_{k\to\infty} \mathbf{Q}^k = \mathbf{0}.$$

As a consequence, the *Neumann series* can be applied:

$$(\mathbf{I} - \mathbf{Q})^{-1} = \sum_{k=0}^{\infty} \mathbf{Q}^k,$$

which converges because the spectral radius $\rho(\mathbf{Q})$ is strictly less than 1. The spectral radius is defined as the largest absolute value of the eigenvalues of $\mathbf{Q}$. Since $\rho(\mathbf{Q}) < 1$, all eigenvalues satisfy $|\lambda| < 1$, so $\mathbf{I} - \mathbf{Q}$ has eigenvalues strictly larger than zero and is therefore invertible.

2.2. Dimensionality Reduction

Many tasks in data analysis involve high-dimensional datasets, which are often difficult to interpret, visualize, or process directly. A common approach to mitigate these challenges is to project the data into a lower-dimensional space while preserving relevant structural information.

In the context of graph-based data, random walks and connectivity patterns reveal meaningful relationships between data points. Such information often motivates dimensionality reduction techniques designed to preserve either global or local structures.

Dimensionality reduction aims to find a mapping

$$g : \mathbb{R}^N \to \mathbb{R}^\ell, \quad \text{with} \quad N < \ell,$$

that reduces the dimensionality of the data while retaining as much relevant information as possible.

This concept appears naturally in various applications. For example, in urban photography, the goal may be to extract a static background from a sequence of images containing moving objects. By taking the temporal median over many images, transient objects like pedestrians are effectively removed, and the background is preserved. Figure 2.5 illustrates such an example. This technique, known as *temporal median filtering* [19], acts as a simple yet effective dimensionality reduction.

This example illustrates a key challenge of dimensionality reduction: complete information preservation is not achievable. Moreover, the user might not

Figure 2.5.: Dimensionality reduction on a video sequence of Big Ben via temporal median filtering.

even desire certain information, such as the pedestrians and vehicles in the image sequence. The choice of method depends on which aspects of the data are intended to be preserved.

In the following, two fundamental techniques will be introduced:

- Principal Component Analysis (PCA), a linear method aiming to preserve the variance of the data.
- t-distributed Stochastic Neighbor Embedding (t-SNE), a non-linear method focusing on preserving local neighborhood relations.

Principal Component Analysis (PCA) is one of the most widely used linear dimensionality reduction techniques. It aims to project data from a high-dimensional space $\mathbb{R}^N$ onto a lower-dimensional subspace $\mathbb{R}^d$, where typically $N < d$. The projection can be written as:

$$\pi : \mathbb{R}^N \to \mathbb{R}^d, \quad x \mapsto \mathbf{V}_d^\top x,$$

where $\mathbf{V}_d \in \mathbb{R}^{N \times d}$ contains the principal directions [20].

To determine $\mathbf{V}_d$, the data is first centered by subtracting its mean [21]. Subsequently, the sample covariance matrix

$$\mathbf{C} = \mathbf{X}^\top \mathbf{X}$$

is computed, where $\mathbf{X}$ denotes the centered data matrix. This covariance matrix captures the variance structure of the data and is symmetric and positive semi-definite. Its eigendecomposition, which coincides with the singular value decomposition (SVD) in this case, yields:

$$\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}^\top.$$

Since $\mathbf{C}$ is of size $N \times N$, the decomposition is computationally efficient, especially when N is moderate.

The matrix $\mathbf{V}$ contains the eigenvectors, also referred to as **principal components**, ordered by decreasing eigenvalues. Selecting the first d eigenvectors corresponding to the largest eigenvalues maximizes the variance captured in the reduced space and results in matrix $\mathbf{V}_d$ [21].

For large datasets, iterative algorithms such as the Lanczos method [22] can compute the leading eigenvectors efficiently.

Due to its efficiency and its focus on preserving global variance structure, PCA is frequently used as an initialization method for non-linear dimensionality reduction techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) [23].

2.2.1. t-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a non-linear dimensionality reduction technique designed to preserve local neighborhood structures [9]. Due to its local focus, it benefits from initialization using PCA [24], although random initialization is also commonly used.

The method proceeds by defining two probability distributions:

- A distribution P measuring similarities between points in the high-dimensional space.
- A distribution Q measuring similarities between points in the low-dimensional embedding.

High-dimensional similarities. The matrix $\mathbf{P}$ encodes conditional probabilities:

$$p_{j|i} = \begin{cases} \dfrac{\exp\left(-\|x_i - x_j\|^2/(2\sigma_i^2)\right)}{\sum_{k\neq i}\exp\left(-\|x_i - x_k\|^2/(2\sigma_i^2)\right)} & \text{if } i \neq j, \\ 0 & \text{otherwise,} \end{cases}$$

where σ_i is chosen such that the perplexity of the distribution matches a user-defined value, typically set to 30:

$$\text{Perplexity} = 2^{\mathcal{H}}, \quad \mathcal{H} = -\sum_{j\neq i} p_{j|i}\log p_{j|i}.$$

The matrix $\mathbf{P}$ is then symmetrized and normalized:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N},$$

where N is the number of data points.

Low-dimensional similarities. In the low-dimensional embedding, the matrix $\mathbf{Q}$ defines the pairwise similarities using a Student-t distribution with one degree of freedom:

$$q_{ij} = \begin{cases} \dfrac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k\neq l}\left(1 + \|y_k - y_l\|^2\right)^{-1}} & \text{if } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

This choice of distribution addresses the *crowding problem* [9], which arises because, in lower-dimensional spaces, points tend to have insufficient "space" to accurately reflect the distances they had in high dimensions. The heavier tails of the Student-t distribution increase the probability of moderate to large distances, allowing better representation of distant points without forcing all points into tight clusters.

Optimization. The discrepancy between $\mathbf{P}$ and $\mathbf{Q}$ is minimized using the Kullback-Leibler (KL) divergence:

$$C = KL(\mathbf{P}||\mathbf{Q}) = \sum_i \sum_j p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right).$$

Gradient descent with momentum is employed to update the positions of the low-dimensional points:

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij}) w_{ij}, \quad \text{where} \quad w_{ij} = \frac{y_i - y_j}{1 + \|y_i - y_j\|^2}.$$

The gradient decomposes naturally into an *attractive* force when $p_{ij} > q_{ij}$, pulling points closer, and a *repulsive* force when $p_{ij} < q_{ij}$, pushing points apart.

An **early exaggeration** technique is typically employed during the initial iterations, where the attractive force is multiplied by a constant factor (usually 12) [9]. This promotes the formation of well-separated clusters early in the optimization, preventing cluster fragmentation and accelerating convergence.

2.3. Clustering

Dimensionality reduction techniques, especially t-SNE, are frequently used in combination with clustering to identify groups of similar data points. The resulting clusters often reveal underlying structures or functions within the data.

Clustering methods are commonly classified into three main categories [25]:

- **Fuzzy clustering**: Each data point can belong to multiple clusters with varying degrees of membership.
- **Overlapping clustering**: A data point can belong to multiple clusters without specifying degrees of membership.
- **Non-overlapping clustering**: Each data point is assigned to exactly one cluster.

An example of fuzzy clustering was already encountered in Section 2.1.4, where the fundamental matrix $\mathbf{B}$ provides probabilistic cluster memberships via absorption probabilities.

The methods introduced below focus on non-overlapping clustering, producing a partition of the data into disjoint subsets. Each algorithm outputs a cluster assignment vector labeling for each data point.

2.3.1. Label Propagation

The label propagation algorithm [26] is a simple and intuitive graph-based clustering method. The input consists of n data points and pairwise similarity information between them, which is typically encoded as a graph.

Initially, each point is assigned to its own cluster, forming n singleton clusters. In each iteration, clusters grow by assigning the most similar unassigned neighboring datapoint to them. This process continues until no unassigned points remain. If a data point is most similar to two clusters at the same time its association is random. In this work, label propagation is implemented using a graph structure, where edges represent pairwise similarities. For simplicity, uniform similarities are used, treating all connections equally.

2.3.2. Leiden Clustering

The Leiden algorithm [27] is an an extension of the Louvain algorithm [28]. Both are community detection methods aiming to maximize the **modularity** of a clustering [29]. Modularity measures the quality of a partition by comparing the observed number of intra-cluster edges to the expected number under a random graph model. As maximizing modularity is known to be NP-hard [30], the algorithm employs a heuristic approach.

The method starts by assigning each vertex to its own cluster. Iteratively, each vertex is considered for relocation to a neighboring cluster if such a move results in a modularity gain [28]. Once no further improvement is possible, a new graph is constructed, where the previously found clusters become nodes. This process is repeated until convergence.

After these steps the Leiden extension enhances the Louvain algorithm by reducing its tendency to produce poorly connected communities [27]. After the local modularity optimization step, the Leiden algorithm includes a refinement phase, during which clusters may be split to enhance intra-cluster connectivity. Furthermore, the algorithm only updates nodes whose neighborhoods have changed, resulting in improved efficiency.

The time complexity of the algorithm is $O(n^2)$ [27]. The implementation used in this thesis is based on the `leidenalg` package [2].

2.3.3. Walktrap

The Walktrap algorithm [31] detects communities using random walks. The underlying intuition is that random walks tend to get "trapped" inside densely connected communities, thus making them suitable for community detection.

Walktrap follows a hierarchical, agglomerative (bottom-up) strategy. A distance between nodes is defined based on their transition probabilities under the random walk:

$$d(i,j) = \sqrt{\sum_k \frac{\left(P_{ik}^t - P_{jk}^t\right)^2}{d_k}},$$

where t is the random walk length and d_k is the degree of node k. Initially, each node forms its own cluster. Iteratively, the two closest clusters are merged.

The algorithm has time complexity $O(n^2 log(n))$ and space complexity $O(n^2)$, where n is the number of nodes and m the number of edges. However, as noted by its authors [31], the worst-case scenario rarely occurs in practice. In this thesis, the `igraph` implementation is used.

[2] The modularity gain for the Leiden algorithm is computed as:

$$\Delta Q = \frac{k_{i,\text{in}} - 2\sum_{\text{tot}} k_i/m}{2m},$$

where m is the total weight of all edges, $k_{i,\text{in}}$ is the sum of weights of edges from node i to nodes in the target cluster, and $\sum_{\text{tot}}$ is the sum of weights of links incident to the nodes in the target cluster.

2.4. Graph Coarsening

Graph coarsening refers to the process of reducing the number of nodes in a graph while preserving its essential structural and functional properties, such as diffusion characteristics or topological features [32]. This process contrasts with graph sparsification, which reduces the number of edges while retaining the vertex set [33].

Coarsening can be applied iteratively, generating a hierarchy of graphs:

$$G = G_0 \succ G_1 \succ \cdots \succ G_n,$$

where $|V(G_0)| > |V(G_1)| > \cdots > |V(G_n)| = d$. The subscript denotes the coarsening level, where G_0 is the original graph and G_n is the coarsened graph. The reduction factor, denoted by $\gamma \in (0, 1)$, controls the shrinkage rate, defining the number of nodes in the next level as:

$$N_\ell = \lfloor N_{\ell-1} \cdot \gamma \rfloor .$$

The coarsening operation can be seen as a surjective projection of $\mathbf{W}$:

$$g : \mathbb{R}^{N \times N} \to \mathbb{R}^{d \times d}.$$

In this thesis, two families of graph coarsening approaches are distinguished: *node aggregation* and *landmark-based* methods (see Figure 2.6).

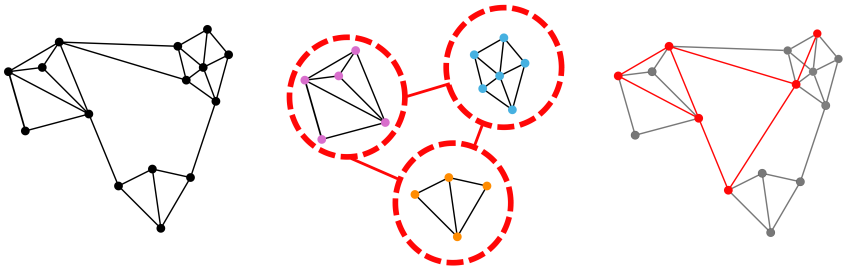

Figure 2.6.: Two graph coarsening paradigms: node aggregation and landmark sampling.

Node aggregation methods identify groups of similar vertices and merge them into new nodes, often called *supernodes*, although terms like *blocks*, *clusters*, or *metacells* are also used. In these methods, each node in the coarsened graph corresponds to a subset of nodes from the original graph.

In contrast, landmark-based methods select a subset of nodes, called *landmarks*, to constitute the vertex set of the coarsened graph directly. Instead of aggregating groups of nodes into supernodes, these methods retain selected nodes and adjust their connectivity to account for the influence of omitted nodes.

3. Material and Methods

This chapter presents the methodological framework employed in this thesis. The general workflow consists of three main stages. First, a k-nearest neighbor (kNN) graph is constructed from the input data. Second, the graph is coarsened using various various node aggregation-based methods and landmark-based approaches. Finally, the reduced graphs are embedded into two dimensions using t-SNE.

3.1. Node Aggregation Methods

3.1.1. Clustering-based Coarsening

As introduced in Section 2.3, algorithms such as Leiden, Walktrap, and Label Propagation partition the nodes of a graph into clusters. These clusters can now serve as **supernodes** for constructing a coarsened graph.

The simplest approach to define edges between supernodes is to sum the connections between all nodes belonging to different clusters:

$$T_s(i,j) = \sum_{k \in C_i} \sum_{l \in C_j} T_{s-1}(k,l),$$

where C_i and C_j denote the sets of nodes belonging to clusters i and j, respectively. This approach is referred to as the **connectivity method**.

An alternative method, called **cluster walks**, connects supernodes based on random walk statistics. For this method, multiple random walks are initiated from each vertex in the original graph, terminating when the walk reaches a node outside the current cluster. A matrix is constructed by counting how often walks starting from cluster C_i reach cluster C_j, resulting in a matrix of size $n_{\text{cluster}} \times n_{\text{cluster}}$. After row-wise normalization, this matrix serves as the transition matrix for the coarsened graph.

It is important to note that both methods treat all supernodes equally, regardless of the size of the underlying clusters. In particular, larger clusters do not receive higher weights solely due to their cardinality.

3.1.2. MetaCell

The MetaCell approach relies on bootstrapping the original graph to obtain more stable cluster structures. Specifically, N_b bootstrap samples are generated by repeatedly sampling subgraphs, where each node is included independently with probability $p = 0.75$. Each subgraph is then clustered using the Walktrap algorithm.

From these multiple clusterings, the **co-occurrence matrix $\mathbf{S}^{\text{boot}}$** is computed, where each entry S_{ij}^{boot} counts how often nodes i and j were clustered together while appearing in the same bootstrap sample.

To sparsify the co-occurrence matrix, an adaptive thresholding is applied: for each node i, the K_{core} (default: 30) most frequently co-occurring neighbors define a threshold T_i. The sparsified weight matrix is defined as:

$$W_{ij}^{\text{boot}} = \begin{cases} S_{ij}^{\text{boot}}, & \text{if } S_{ij}^{\text{boot}} > \frac{\max(T_i, T_j)}{2}, \\ 0, & \text{otherwise.} \end{cases}$$

The Walktrap algorithm is then applied to the sparsified graph, yielding the final clusters $C_1, C_2, \ldots, C_n$.

The weight matrix of the resulting reduced graph is constructed similarly to the connectivity method but incorporates additional normalization:

$$B_{ab} = \frac{K_{core}^2}{|C_a| \cdot |C_b|} \sum_{i \in C_a} \sum_{j \in C_b} \left\lceil \frac{W_{ij}}{\text{median}_k(|C_k|)} \right\rceil .$$

This adjustment regularizes the weights by accounting for cluster sizes relative to the median cluster size, thereby reducing the influence of small clusters, which could otherwise be overrepresented in the coarsened graph.

3.1.3. Local Variation Methods

Spectral graph theory, as introduced in Section 2.1.3, provides a rigorous mathematical framework for analyzing graphs. While spectral theory has been successfully applied in *spectral sparsification*, i.e., reducing the number of edges in a graph, the problem of *graph coarsening*—reducing the number of nodes—lacks a comparable theoretical foundation. Loukas et al. [32] noted that existing theory supporting efficient graph reduction is largely circumstantial and identified the difficulty of predicting how different reduction strategies affect spectral properties as a core challenge.

One algorithm addressing this challenge is the *Local Variation* method proposed by Loukas et al., which forms the basis of the following approach. Unlike the clustering-based methods introduced earlier, Local Variation does not employ a traditional clustering algorithm. Instead, it constructs *candidate sets* of nodes based on local connectivity patterns.

Two modes for generating candidate sets are proposed:

- **Edge-based mode**: Each candidate set consists of all pairs of adjacent vertices.
- **Neighbor-based mode**: Each candidate set consists of all neighbors of a given vertex.

These candidate sets are then evaluated based on a **local spectral variation** cost function:

$$\text{cost}^k(\mathcal{C}) = \sum_{i=1}^{k} \frac{\sum_{u,v \in C} w_{uv} \left(\phi_i(u) - \phi_i(v)\right)^2}{\lambda_i},$$

where $\mathcal{C}$ is the candidate set and k the dimension of the eigenspace that should get preserved. $\phi_i(u)$ denotes the value of the i-th eigenvector of the Laplacian at vertex u, and w_{uv} is the weight of the edge (u, v) and then normalized by its respective eigenvalue λ_i. As discussed in Section 2.1.3, eigenvectors and eigenvalues encode global and local structures of the graph. This cost function favors groups of nodes with similar spectral properties and penalizes groups that are strongly connected but spectrally dissimilar.

Algorithm Overview. The algorithm receives as input:

- The Laplacian matrix $\mathbf{L}_{\ell-1}$ of the graph at level $\ell - 1$.
- A maximum local standard deviation threshold σ'.
- A target number of nodes.

The eigenvectors used in the cost function are expensive to compute; thus, the default value is $k = 4$. The threshold σ' is determined as the average local spectral variation cost multiplied by a tunable constant c. All vertices are initially unmarked, and the current variance σ_ℓ^2 is set to zero. An empty set is introduced $\mathcal{P}_\ell$, that will contain all accepted candidate sets. Candidate sets are generated according to the selected mode, and their local variation costs are computed and stored in a list $\mathcal{F}_\ell$, sorted by increasing cost.

Greedy Selection. The algorithm proceeds iteratively until one of the following stopping criteria is met:

- The list $\mathcal{F}_\ell$ is empty.
- The standard deviation σ_ℓ exceeds the threshold σ'.
- The desired number of supernodes is reached.

In each iteration, the candidate set $\mathcal{C}$ with the smallest cost is removed from $\mathcal{F}_\ell$. If none of its nodes are marked, the current variance is updated:

$$\sigma_\ell^2 \leftarrow \sigma_\ell^2 + (|\mathcal{C}| - 1) \cdot s,$$

where s is the cost of $\mathcal{C}$. If σ_ℓ remains below σ', the candidate set is accepted:

- All nodes in $\mathcal{C}$ are marked.
- $\mathcal{P}_\ell$ is updated to include $\mathcal{C}$.
- The current variance is updated to $\sigma_\ell = \sqrt{\sigma_\ell^2}$.

If, instead, some nodes in $\mathcal{C}$ are already marked, the unmarked subset $\mathcal{C}' \subset \mathcal{C}$ is extracted. If $|\mathcal{C}'| > 1$, its cost is recalculated and $\mathcal{C}'$ is reinserted into $\mathcal{F}_\ell$.

Graph Projection. After termination, the **restriction operator** matrix $\mathbf{P}_\ell$ is extracted from the $\mathcal{P}_\ell$. The restriction operator $\mathbf{P}_\ell$ is constructed as a binary matrix where each row corresponds to a candidate set in $\mathcal{P}_\ell$. For each such set, all columns (nodes) belonging to the set are assigned a value of 1 in the corresponding row, and 0 elsewhere. When used in matrix multiplication, this effectively fuses the nodes of each set by aggregating their contributions into a single coarse node. The coarsened Laplacian is computed as:

$$\mathbf{L}_\ell = \mathbf{P}_\ell^{\mp} \mathbf{L}_{\ell-1} \mathbf{P}_\ell^{+},$$

where $\mathbf{P}_\ell^{+}$ denotes the pseudoinverse of $\mathbf{P}_\ell$ [1]. Due to the construction of $\mathbf{P}_\ell$, it holds that $\mathbf{P}_\ell^{+} = \mathbf{P}_\ell^{\top}$, making the computation efficient [2]. Multiple rounds of graph coarsening may be required to achieve a desired reduction factor. Notably, in the *edge-based* mode, the graph size can be reduced by at most a factor of two in a single iteration, as candidate sets are restricted to edge pairs.

3.2. Landmark Approach

In contrast to aggregation-based methods, the landmark approach is typically not restricted by the cluster sizes given by the node aggregation algorithm. They can be applied iteratively, yielding a hierarchical structure indexed by the level ℓ. One iteration of graph coarse-graining can be naturally divided into two independent stages: **landmark sampling** and **landmark connecting**.

In the landmark sampling step, a subset of vertices is selected as landmarks, directly determining the size of the coarsened graph. The landmark connecting step establishes the connections between these landmarks, resulting in a reduced graph. The following subsections describe several strategies for both stages.

3.2.1. Landmark Sampling

Five different landmark selection strategies are considered. All methods operate on the input graph and reduce it to a user-specified number n of landmarks.

[1] $\mathbf{P}_\ell^{\mp}$ is the transpose of $\mathbf{P}_\ell^{+}$
[2] Proof in appendix at A.1

Random Sampling. The simplest strategy selects n landmarks uniformly at random. This method can result in poor coverage if selected landmarks are concentrated in a small region of the graph.

Plum Pudding Sampling (PPS). Plum Pudding search improves upon random sampling by preventing nearby points from being selected as landmarks [34]. After sampling a landmark, all its k-nearest neighbors are excluded from further selection, where k is the same parameter used to construct the initial kNN-graph. This process repeats until no more points can be selected. Since this approach depends on the graph's connectivity, it may occasionally yield fewer landmarks than desired. Moreover, PPS is not entirely immune to poor coverage, as illustrated in Figure 3.1. While PPS offers improved coverage compared to random sampling, other methods also exhibit limitations, and achieving consistently well-distributed landmark sets remains an open challenge.

Hub Sampling (Hubs) and Hub Excluding Neighbors (HBN). High-degree nodes, or *hubs*, often play structurally significant roles in graphs [35, 36]. HUBS selects the nodes with the highest degrees as landmarks, which can often yield representative samples. In the variant HBN, hubs are selected while excluding their immediate neighbors from consideration, analogous to PPS. Both methods, however, may still suffer from uneven coverage depending on the graph structure.

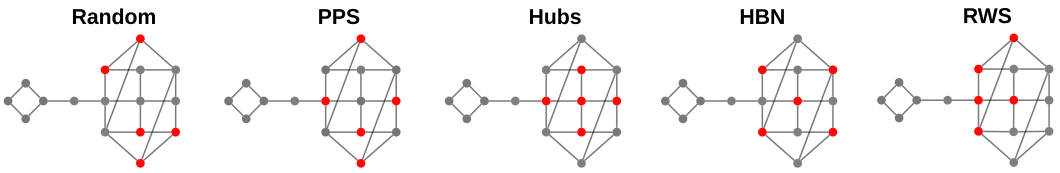

Figure 3.1.: Failure cases of landmark sampling strategies showing insufficient coverage of the left part of the graph.

Random Walk Sampling (RWS). RWS selects landmarks based on the distribution of random walk endpoints. A fixed number of random walks (default: 100) is initiated from each node, with each walk having a predefined length (default: 50 steps). The vertices most frequently visited at the end of these walks are selected as landmarks. This approach adapts from Pezzotti et al. [8] and can also be parameterized by a threshold instead of a fixed number of landmarks. In this work, the method is used by selecting the top-n vertices with the highest number of endpoint counts.

3.2.2. Landmark Connecting

Once landmarks have been sampled, they must be connected to form the reduced graph. Two main paradigms are considered:

- **Landmark-to-Landmark (L-L) connections**
- **State-to-Landmark (S-L) connections**

Landmark-to-Landmark (L-L) Connection

The L-L method connects landmarks directly by simulating how easily landmarks can reach each other in the original graph [9].

Multiple walks (parameter: n_{walks}) are initiated from each landmark, terminating upon encountering another landmark (excluding itself). The weight assigned to the edge between two landmarks is proportional to the fraction of walks that terminate at the target landmark:

$$W_{ij} = \frac{\text{number of walks from } i \text{ ending at } j}{n_{\text{walks}}}.$$

The resulting matrix $\mathbf{W}_\ell$ is row-stochastic but not necessarily symmetric. It can be interpreted as a transition matrix indicating the likelihood of moving between landmarks.

State-to-Landmark (SL) Connection

The SL method estimates transition probabilities indirectly by first computing the **influence matrix** $\mathbf{I}_\ell \in \mathbb{R}^{N_{\ell-1} \times N_\ell}$, where $I_\ell(s, i)$ represents the probability of reaching landmark i from node s via random walks on current level ℓ. This can be interpreted as a fuzzy clustering of all vertices in the graph.

The transition matrix $\mathbf{W}$ between landmarks is then computed as:

$$W'_\ell(i,j) = \sum_{k=1}^{N_{\ell-1}} I_\ell(k,i) I_\ell(k,j) w_{\ell-1}(k), \quad W_\ell(i,j) = \frac{W'_\ell(i,j)}{\sum_{m=1}^{|\mathcal{N}_\ell|} W'_\ell(i,m)},$$

where $\mathbf{m}_{\ell-1}$ is the weight vector describing the sizes or importance of nodes at the previous level. Initially, $\mathbf{m}_0$ is simply the vector of all ones. This formulation naturally incorporates the accumulated influence of all nodes when constructing inter-landmark connections. The mass vector is recursively updated, progressivly shrinking to the size of the size of the Influence matrix at each iteration:

$$\mathbf{m}_\ell = \mathbf{m}_{\ell-1}\mathbf{I}_\ell, \quad \mathbf{W}'_\ell = \mathbf{I}_\ell \cdot \mathbf{m}_{\ell-1} \cdot \mathbf{I}_\ell^\top.$$

$$m_\ell = m_{\ell-1} I_\ell, \quad W'_\ell = I_\ell \cdot m_{\ell-1} \cdot I_\ell^\top.$$

Despite the appearance of a quadratic complexity, Pezzotti et al. [8] observed that the sparsity of $\mathbf{I}_\ell$ makes the computation scale linearly in practice, due to localized regions of influence. Unconnected landmarks are filtered out before embedding. This method was originally developed with random walk–based landmark selection by Pezotti et al [8]. However, given our use of hubs, a fixed number of nodes, and more, we will refer to it as the State-to-Landmark (SL) approach to avoid potential confusion.

3.2.3. Kron Reduction

Kron reduction [35] is a well-established method for reducing graphs and is closely related to the SL approach described previously. In fact, it can be interpreted as the limiting case of the SL connection strategy, where random walks are replaced by solving for the exact absorption probabilities under a continuous diffusion model.

In its classical formulation, Kron reduction originated from electrical circuit theory, where the Laplacian matrix corresponds to the *admittance matrix* of a resistive network [35]. Vertices represent generators, and edges represent transmission lines. The Kron reduction aims to remove a subset of nodes while preserving quantities related to current flow, resistance distances, and diffusion behavior [5]. Given a selection of landmarks (often hubs [35, 36]), the Laplacian matrix is partitioned as:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{aa} & \mathbf{L}_{at} \\ \mathbf{L}_{ta} & \mathbf{L}_{tt} \end{pmatrix},$$

where the subscript a refers to landmark nodes and t to the remaining non-landmark nodes [3]. The reduced Laplacian is then computed using the **Schur complement** (rear section)[4]:

$$\mathbf{L}' = \mathbf{L}_{aa} - \mathbf{L}_{at}\mathbf{L}_{tt}^{-1}\mathbf{L}_{ta}.$$

This formula is equivalent to computing the exact influence of all possible diffusion (or random walks) from the non-landmark nodes into the landmark nodes. Therefore, Kron reduction can be seen as the exact version of the SL approach when random walks are allowed to fully relax to their stationary distribution.

In this interpretation, random walks or diffusions starting from any non-landmark node are fully propagated until they are absorbed by a landmark. This provides an alternative perspective on the *influence matrix* I used in

[3] Note the similarities to the canonical form introduced in 2.1.4

[4] A formal derivation of the Kron reduction from the Kirchhoff node potential equation is provided in Appendix A.1

Section 3.2.2, with Kron reduction providing the limiting case where the influence is computed exactly via linear system solving rather than estimated via sampling.

The reduced Laplacian $\mathbf{L}'$ maintains several important properties:

- Positive semi-definiteness.
- Preservation of resistance distances (interpretable as diffusion distances).
- Approximate preservation of small eigenvalues [7].

Kron reduction inherently assumes a Laplacian matrix, typically derived from either a standard or random walk formulation, and thus presupposes a bidirectional graph structure. If the if the underlying process can be modeled as an absorbing Markov chain, as discussed in Section 2.1.4, a similar reduction is also feasible for unidirectional graphs using the fundamental matrix $\mathbf{B}$. However, due to the dense nature of the Schur complement, the resulting graph typically becomes denser, even if the original graph is sparse [37]. We note that the exact solution via Kron reduction was also derived by Grady et al. and referenced in the original t-SNE paper [9, 38].

3.3. Metrics

3.3.1. Graph Metrics

Betweenness Centrality

Betweenness centrality quantifies the importance of a node in terms of its participation in the shortest paths between other pairs of nodes [39]. Specifically, it measures the fraction of all shortest paths that pass through a given node. Since computing betweenness centrality exactly requires considering all shortest paths in the graph, which can be computationally expensive, it is common practice to approximate it by selecting a subset of k nodes and estimating the metric based on this sample [40].

Harmonic Centrality

In a connected graph, harmonic centrality evaluate how central a node is based on its average distance to all other nodes [41]. It is defined as the reciprocal of the sum of the lengths of the shortest paths from a node to all other nodes:

where $d(i, j)$ denotes the shortest path distance between nodes i and j. This metric is often interpreted as the inverse of the average shortest path length [42, 5]. This is the improved version of closeness centrality, which accounts for disconnected components [43].

Kullback-Leibler Divergence

The Kullback-Leibler (KL) divergence [44], introduced earlier in Section 2.2.1, serves as a measure of similarity between probability distributions. In this work, it is used to compare the distributions of betweenness and closeness centrality between the original and reduced graphs (see Section 4). Direct comparison of the raw centrality values is not feasible due to the difference in graph sizes. However, both centrality profiles are assumed to be characteristic of the graph structure. To enable comparison:

- Histograms with the same number of bins are constructed for both the original and reduced graphs.
- The KL divergence is computed between these histograms.

A KL divergence value close to zero indicates a high similarity between the two distributions, suggesting that the respective metric is well preserved in the reduced graph.

Spectral Graph Distance

While betweenness and closeness centrality capture node-level information, the global structure of the graph must also be considered. To this end, the spectral norm distance [45] is employed, which compares the spectra (i.e., eigenvalues) of the Laplacian matrices of two graphs. Specifically, it is defined as:

$$D_{\text{spec}}(G, G') = \sum_{i=1}^{n_{\text{eig}}} \left| \lambda_i^{(G)} - \lambda_i^{(G')} \right|, \tag{3.1}$$

where $\lambda_i^{(G)}$ and $\lambda_i^{(G')}$ denote the i-th smallest eigenvalues of the original and reduced graphs, respectively, and $n_{\text{eig}} = \min(t_{\text{eigvals}}, \min(m, n))$ is the number of eigenvalues considered, with a default value of $t_{\text{eigvals}} = 1000$. I a graph is smaller as this threshold then its size is taken. This metric quantifies the global structural similarity between the graphs; smaller values indicate a higher similarity.

Average Relative Eigenvalue Error

Since smaller eigenvalues of the Laplacian correspond to large-scale structures, it is desirable to weigh their deviations more heavily. The average relative eigenvalue error [7] addresses this by scaling each eigenvalue difference by the corresponding eigenvalue from the original graph:

$$\text{aRelEErr} = \frac{1}{n_{\text{eig}}} \sum_{i=1}^{n_{\text{eig}}} \frac{|\lambda_i^{(G)} - \lambda_i^{(G')}|}{\lambda_i^{(G)}}. \tag{3.2}$$

This adjustment ensures that deviations in the lower part of the spectrum, which capture the essential global structure, have a stronger influence on the metric.

3.3.2. Embedding Metrics

k-Nearest Neighbor Accuracy

The k-Nearest Neighbor Accuracy is a widely used metric for assessing neighborhood preservation [8, 9]. It measures the extent to which the k nearest neighbors of each point are preserved under the embedding:

$$\text{kNN-Acc}(X^{\text{ori}}, X^{\text{emb}}) = \frac{1}{k|X|} \sum_{i \in X} |\text{kNN}(X_i^{\text{ori}}) \cap \text{kNN}(X_i^{\text{emb}})|, \tag{3.3}$$

where X^{ori} and X^{emb} denote the data points in the original and embedded space, respectively. For the landmarking approach, this subset selection is straightforward, whereas for clustering-based methods, a representative node is sampled randomly from each cluster.

Trustworthiness

Trustworthiness [46] assesses whether points that are neighbors in the embedding space were also close in the original space. It is defined as:

$$T(k) = 1 - 2\frac{\sum_i \sum_{j \in U(i)} (\text{rank}^{\text{ori}}(i,j) - k)}{nk(2n - 3k - 1)}, \tag{3.4}$$

where $U(i)$ is the set of points that are among the k-nearest neighbors of i in the embedding space but not in the original space, and $\text{rank}^{\text{ori}}(i,j)$ is the rank of j relative to i in the original space. This metric is computed similarly to kNN-Accuracy by considering a relevant subset.

Davies-Bouldin Index

Since categorical datasets are used (see Section 3.4), cluster quality is assessed using the Davies-Bouldin Index (DBI) [47]. DBI balances intra-cluster compactness and inter-cluster separation:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} R_{ij}, \quad \text{where} \quad R_{ij} = \frac{a(i) + a(j)}{d_{\text{inter}}(i,j)},$$

with $a(i)$ and $a(j)$ denoting the average intra-cluster distances for clusters i and j, and $d_{\text{inter}}(i,j)$ the inter-cluster distance. Lower DBI values indicate better cluster separation and compactness.

Silhouette Value

The silhouette value [48] is another clustering metric. Compared to DBI, it is less sensitive to outliers [49] and accommodates non-spherical cluster shapes. For a data point i, the silhouette value is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{3.5}$$

where $a(i)$ is the average intra-cluster distance, and $b(i)$ is the smallest average distance between i and the points of any other cluster. By construction, $s(i) \in [-1, 1]$, where values close to 1 indicate good cluster assignment, values around 0 suggest ambiguous assignment, and negative values imply misclassification. The silhouette coefficient is defined as the average of $s(i)$ over all points, either globally or per cluster.

3.4. Datasets

To evaluate the proposed methods on graphs, multiple kNN-graphs are constructed from datasets belonging to two distinct categories: **categorical datasets** and **manifold datasets**. The categorical datasets consist of widely-used real-world datasets with available class labels (ground truth). The manifold datasets are synthetically generated and exhibit an underlying two-dimensional structure embedded in three dimensions.

3.4.1. Categorical Datasets

Digits

The **Digits** dataset [50] is a classical dataset from which the test set is used. It contains 1,797 grayscale images of handwritten digits. Each image is of size 8×8, resulting in a 64-dimensional feature space. The dataset includes ten equally distributed classes, corresponding to the digits 0 through 9. Pixel values range between 0 and 255.

MNIST

The **MNIST** dataset [51] is a larger benchmark dataset also consisting of handwritten digits. It comprises 70,000 grayscale images of size 28×28, leading to a 784-dimensional feature space. As in the Digits dataset, the classes represent digits 0 to 9 and are approximately equally distributed.

FMNIST

The **FMNIST** dataset [52] contains 70,000 color images of size 28×28 with one channel, resulting in a 748-dimensional feature space. Unlike MNIST, FMNIST focuses on natural images, consisting of ten classes representing clothing, each containing exactly 7,000 samples.

TASIC

The **Transcriptional Atlas of the Mouse Central Nervous System (TASIC)** dataset [53] consists of transcriptomic data from 23,822 mouse brain cells, categorized into 133 cell types. The classes are organized hierarchically, distinguishing inhibitory neurons, excitatory neurons, and non-neuronal cells. The dataset exhibits an imbalanced distribution of classes and is known to possess a hierarchical structure, where some classes represent broad cell types and others indicate specific subtypes [53]. A preprocessed version of this dataset is

used, containing the first 50 principal components, as provided by the Berens Lab.

3.4.2. Manifold Datasets

Swiss Roll

The **Swiss Roll** dataset is a classical synthetic dataset generated based on the implementation of [54] and extended in this thesis as can be seen in Figure 3.2. The dataset consists of 5,000 points sampled from a two-and-a-half turn spiral embedded in three dimensions. The color gradient indicates the position along the roll. Compared to the original version, the number of points and the extent of the spiral are increased. The known two-dimensional parametrization of the manifold allows for ground-truth comparison during dimensionality reduction.

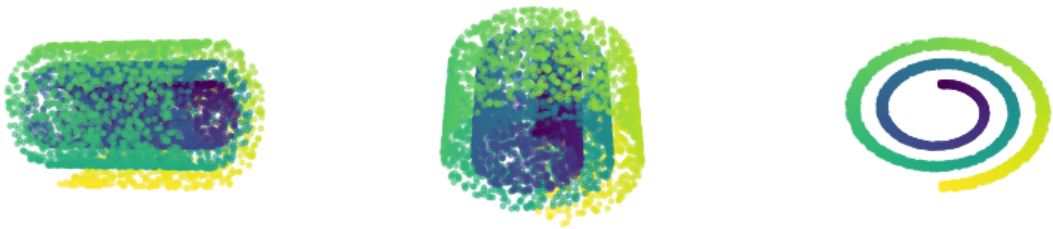

Figure 3.2.: Swissroll dataset from different points of view

DNA

The **DNA** dataset is synthetically generated and consists of 12,000 points forming two intertwined helical structures connected by cross-edges, resembling the structure of a DNA molecule. The structure is coiled twice. After appropriate unfolding, the dataset exhibits a ladder-like structure in two dimensions. All coordinates range between -1 and 1. With the corresponding y value increasing along its coils from zero to one. The dataset was created specifically for this thesis.

4. Results

4.1. Establishing a Baseline

To better understand what constitutes a strong performance according to the evaluated metrics, we establish a baseline using random subsampling. In this experiment, we randomly select a subset of points from the original dataset and construct a kNN-graph from it, followed by t-SNE embedding. To simulate a second level of coarse-graining, we apply the same procedure again to the subset, effectively mimicking two hierarchical levels without applying any coarse-graining algorithm.

This process is repeated five times per dataset to capture variance. Tables 4.1 and 4.2 report the average values and standard deviations for each metric at Level 1 and Level 2, respectively. This might be an inappropriate comparison instead of aggregating information, information is filtered random, therefore these metrics shouldn't be interpreted as an upper nor lower bound.

Metrics	MNIST	TASIC	FMNIST
kNN-Accuracy	0.453 ($\pm$0.003)	0.644 ($\pm$0.006)	0.440 ($\pm$0.002)
Trustworthiness	0.987 ($\pm$0.001)	0.997 ($\pm$0.000)	0.990 ($\pm$0.001)
Silhouette	0.218 ($\pm$0.015)	0.232 ($\pm$0.014)	0.149 ($\pm$0.009)
DBI	3.999 ($\pm$1.943)	2.161 ($\pm$0.288)	2.319 ($\pm$0.076)
harmonic	4.716 ($\pm$0.143)	5.732 ($\pm$1.158)	7.572 ($\pm$0.396)
betweenness	0.309 ($\pm$0.043)	0.412 ($\pm$0.088)	5.848 ($\pm$0.204)
dspectral	93.934 ($\pm$0.010)	98.670 ($\pm$0.010)	15.864 ($\pm$0.022)
rel. eigenerr	56.827 ($\pm$61.258)	177.356 ($\pm$32.57)	1.562 ($\pm$1.062)

Table 4.1.: Average metrics with standard deviation for the random subset (simulated Level 1) using undirected kNN-graphs.

Baseline Analysis

These baseline results demonstrate that metric values are not directly comparable across datasets. For example, the spectral distance for MNIST is approximately six times higher than for FMNIST, underscoring the influence

Metrics	MNIST	TASIC	FMNIST
kNN-Accuracy	0.541 (±0.007)	0.802 (±0.010)	0.563 (±0.012)
Trustworthiness	0.957 (±0.003)	0.992 (±0.002)	0.978 (±0.003)
Silhouette	0.141 (±0.023)	0.025 (±0.051)	0.142 (±0.050)
DBI	7.669 (±8.505)	1.325 (±0.221)	3.128 (±1.129)
harmonic	4.945 (±0.697)	11.774 (±2.054)	6.911 (±1.409)
betweenness	0.498 (±0.118)	0.457 (±0.110)	9.327 (±0.130)
dspectral	77.599 (±0.001)	66.314 (±0.001)	25.581 (±0.007)
rel. eigenerr	6564.774 (±0.027)	51678.266 (±19572.720)	104.381 (±0.002)

Table 4.2.: Average metrics with standard deviation for the random sub-subset (simulated Level 2) using undirected kNN-graphs.

of dataset-specific structure. Nevertheless, relative comparisons within each dataset remain meaningful.

Across datasets, the kNN-Accuracy for the random subset ranges around 0.45 for MNIST and 0.44 for FMNIST, while TASIC achieves higher values due to its inherent hierarchy. Trustworthiness remains high across all datasets and levels, exceeding 0.98 in all cases, suggesting that local neighborhood preservation is relatively stable under random subsampling.

Notably, the standard deviations for most metrics are low, indicating stability across the five repetitions. The only exception is the relative eigenvalue error, which shows considerable variance, particularly in the second-level subset. This suggests that while most structural and embedding-based metrics yield reliable baselines, spectral sensitivity may be more susceptible to sampling variability.

Interestingly, reducing the number of points further in the second-level simulation does not lead to higher variance but does result in systematically lower quality embeddings across nearly all metrics. This highlights the inherent difficulty of maintaining structure as data is aggressively downsampled, and emphasizes the need for thoughtful design in coarse-graining algorithms.

4.2. Node Aggregation Methods

Following the baseline analysis, we now evaluate a set of algorithmic approaches for graph coarsening. The datasets used in this thesis were selected for their structural diversity, allowing us to observe how different methods perform under varying conditions. To ensure comparability, all methods were applied across all datasets.

We begin by focusing on MNIST as a representative benchmark. A k-Nearest Neighbor (kNN) graph is constructed using $k = 10$, and two levels of coarsening are applied, each with a reduction factor of 0.1. This results in approximately

7,000 nodes at the first coarsened level and around 700 at the second level.

4.2.1. Local Variation Methods

Both local variation methods modes, i. e. neighborhood-based (LV Neigh.) and edge-based (LV Edge) were evaluated using an eigenspace preservation parameter of three, as suggested in the original work by Loukas et al.

Figure 4.1 shows the resulting embeddings of the coarsened MNIST kNN-graphs for both modes at two levels.

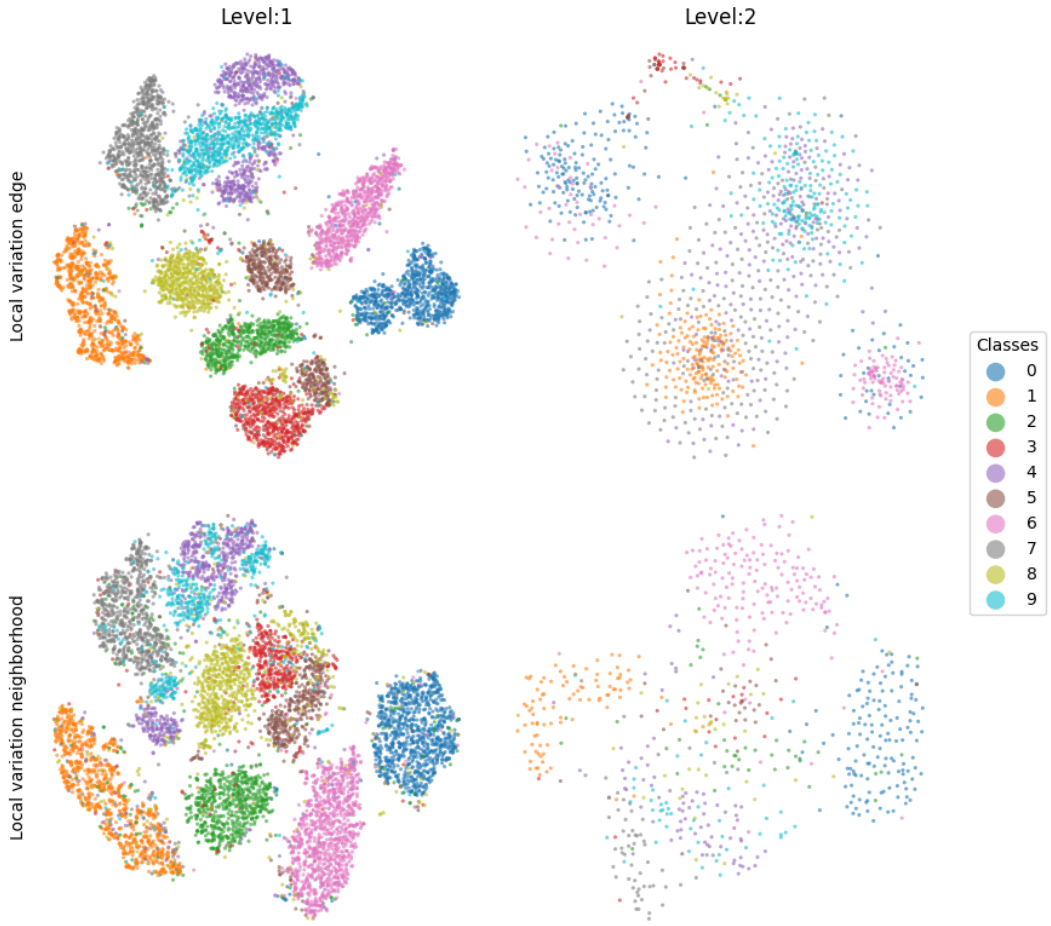

Figure 4.1.: Embeddings of kNN-graphs of MNIST coarsened with Local Variation methods with two levels. Left: Edge-based method. Right: Neighborhood-based method

At the first level of coarsening, the neighborhood-based method produces a clear clustering structure, though some digits remain entangled. Notably, the cluster representing digit 9 (light blue) overlaps with digit 4 (purple). They can look similar structure and we will find them therefore often intertwined.

Additionally, the yellow cluster (digit 8) is partially split, with a segment placed across the red and brown clusters (digits 5 and 3). While in the edge-based method digit 4 (purple) and digit 9 (lightblue) also overlap but are less fragmented, while cluster of digit 5 (brown) is separated. In the second coarsening level we can see that the embedding is a mixture of different classes, without clear class boundaries. For the neighborhood version, the embedding becomes also less distinct, compared to the first level. Only a few clusters remain clearly separated—such as the blue cluster corresponding to digit 0, the orange cluster representing digit one, and the pink cluster representing digit 6. Other digit classes are more entangled, with overlapping structures that reduce the interpretability of the coarsened graph.

4.2.2. Clustering Methods

We now compare the node aggregation methods with clustering-based approaches, using three algorithms: Leiden, Walktrap, and Label Propagation. For Walktrap and Label Propagation, the clustering process yields exactly 7,000 and 700 clusters, matching the desired reduction factors. In contrast, Leiden clustering requires a resolution parameter to control the number of clusters. We empirically set the resolution to 752 for the first level and 50 for the second, resulting in 7,034 clusters in the first level and 706 in the second.

The embeddings produced by these clustering methods are shown in Figure 4.2.

Compared to the Local Variation methods, these clustering approaches yield more distinct and compact clusters. In the Leiden clustering (left column), we observe some over-segmentation at the first level. For instance, the green cluster representing digit 3 and the brown cluster representing digit 5 are split into two separate subclusters. Similarly, the purple cluster for digit 4 is partially detached and appears near the light blue cluster corresponding to digit 9. At the second level, however, these effects are mitigated. The previously split clusters appear more cohesive, and the overall separation between digit classes is preserved. While the relative positions of the clusters remain similar, the overall layout of the embedding changes.

A similar trend is visible in the Walktrap results (middle column). The pink cluster, which is initially fragmented in the first level, becomes more coherent in the second. Notably, the orange cluster appears larger in the second level—both in absolute terms and compared to the other methods. Additionally, Walktrap results in fewer scattered or noisy points compared to the Leiden method across in the second level.

The Label Propagation method (right column) also produces a clean separation between clusters in the first level, with a layout similar to that of the Leiden clustering. However, the purple cluster corresponding to digit 4 is again

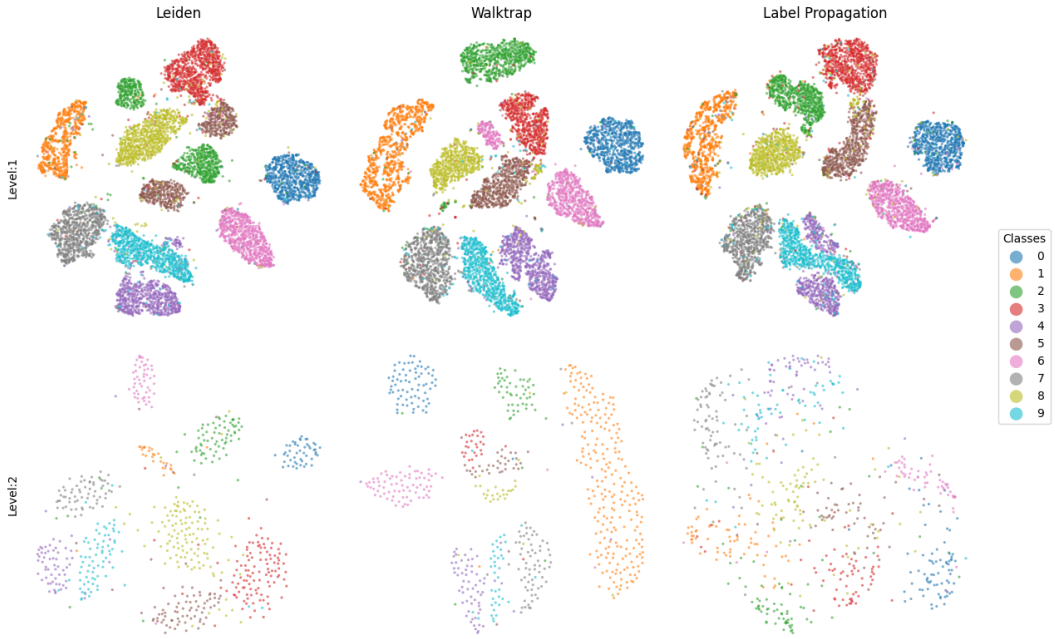

Figure 4.2.: Embeddings of coarsened MNIST kNN-graphs using clustering methods. Top row: First coarsening level. Bottom row: Second coarsening level. Columns correspond to the Leiden, Walktrap, and Label Propagation methods, respectively.

split by the light blue cluster representing digit 9. Interestingly, in the second level, the embedding quality deteriorates. While the overall layout remains comparable to the first level, the clusters are less compact and more noise is visible. Nevertheless, some digit classes—such as green (2), red (3), and pink (6)—remain relatively well-separated. This stands in contrast to the Local Variation Neighborhood method, where different clusters weren't preserved across levels.

4.2.3. Node Aggregation Methods: Quantitative Comparison

Table 4.3 presents a quantitative comparison of the node aggregation methods at the first coarsening level on the MNIST dataset. The metrics include both embedding quality measures and graph structural fidelity indicators.

The first four rows evaluate the embedding quality. Among all methods, the Walktrap algorithm achieves the highest kNN-Accuracy, Trustworthiness, and average Silhouette score, while also yielding the lowest Davies-Bouldin Index. This indicates not only well-separated and compact clusters but also highlights its effectiveness in preserving local neighborhoods during the dimensionality

Level 1	LV (Edge)	LV (Neigh.)	Leiden	Walktrap	LP
k**NN-Accuracy**	0.184	0.203	0.222	**0.245**	0.204
Trustworthiness	0.945	0.941	0.969	**0.970**	0.959
Silhouette	0.306	0.243	0.378	**0.389**	0.369
DBI	1.910	4.008	1.808	**1.054**	1.505
harmonic (KL-Div.)	9.184	5.716	6.318	**4.356**	5.281
betweenness (KL-Div.)	0.547	0.593	**0.450**	0.761	1.354
dspectral	3861.560	6821.510	91.470	93.410	**90.100**
rel. eigenerr	**1.039**	1.055	16.150	14.390	21.820

Table 4.3.: Metrics of node aggregation methods on MNIST, best value for each metric is bold

reduction process, reflecting good retrieval of nearest neighbors from high- to low-dimensional space. These results are consistent with the qualitative visualizations, which show Walktrap producing clean and distinct embeddings.

The next two metrics compare structural centrality profiles between the original and coarsened graphs using Kullback-Leibler divergence. Here, Walktrap again performs best for harmonic centrality, while Leiden yields the lowest divergence for betweenness centrality, suggesting it better preserves the relative importance of nodes in terms of shortest paths.

In terms of spectral fidelity, the Label Propagation method achieves the smallest spectral distance (90.10), closely followed by Leiden (91.47) and Walktrap (93.41). By contrast, the Local Variation methods exhibit significantly higher spectral distances, exceeding 3,000, indicating substantial distortion in spectral properties.

Interestingly, the relative eigenvalue error tells a different story. Despite their poor performance in spectral distance, the Local Variation methods achieve the smallest relative eigenvalue errors, with the edge-based variant yielding the lowest value (1.039). This suggests that while their overall spectral profiles differ from the original, they still preserve the leading eigenvalues relatively well. Among the clustering methods, Walktrap performs best in this category as well, with a relative error of 14.39.

We observe further that compared to our established baseline the kNN Accuracy and Trustworthiness went down. While the KL-Divergence of betweenness and harmonic centrality exhibit values above the random subset. While for Leiden, Walktrap and Label Propagation the Silhouette value and the DBI improved but this was expected as all of these methods employ clustering algorithms. Interestingly, both the spectral distance norm and the relative eigenvalue error are lower than those observed in the baseline.

4.3. Landmarking approaches

4.3.1. Landmark Selection

Since landmark-based methods consist of two distinct steps—selecting landmarks and computing connections, it is natural to assess both stages independently. We assume that insights drawn from evaluating the landmark selection step will be broadly representative for all subsequent connection methods.

To this end, we evaluate several landmark selection strategies using the SL connection. The resulting embeddings of the coarsened graphs are shown in Figure 4.3.

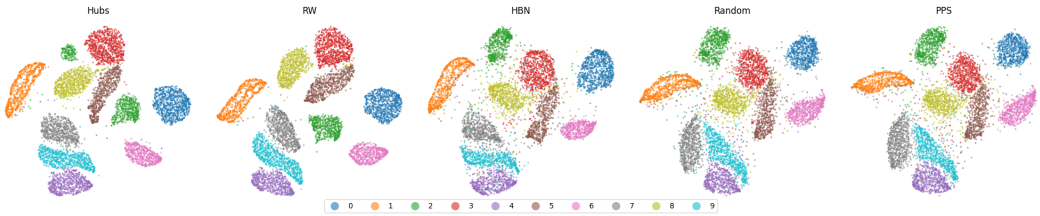

Figure 4.3.: Embeddings of coarsened MNIST kNN-graphs using different landmark sampling strategies.

We evaluate landmark selection strategies exclusively at the first coarsening level to avoid inconsistencies introduced by propagation effects in deeper hierarchies.

At the first coarsening level, the hub-based landmark sampling strategy visibly fragments the green cluster corresponding to digit 2. Additionally, we observe gaps within the orange cluster representing digit 1, a pattern previously seen in clustering-based methods. Both hubs and random walk sampling show discontinuities within the red cluster (digit 3), again consistent with earlier observations.

In contrast, the random, HBN, and PPS sampling strategies introduce more noise and reduce inter-cluster separation—particularly between digits 3 (red), 5 (brown), and 8 (yellow). These observations are not limited to visual inspection. The corresponding embedding metrics also confirm their comparatively lower quality as can be seen in Table A.1. Thus, these strategies yield both noisier and quantitatively less structured embeddings than hub- or random walk–based selection. Interestingly the graph metrics for the PPS sampling all perform better than those of the hubs and random walk sampling achieving lower KL-Divergence for Betweenneess (1.332) and Hamonic Centrality (8.552) compared to 3.296 and 10.747 respectively.

4.3.2. Effect of kNN-graph Directionality on Landmark Sampling

Before proceeding to the evaluation of different landmark connection strategies, we highlight an important observation regarding the behavior of the SL random walk method when applied to directed kNN-graphs compared to undirected.

In both the landmark selection and landmark connection stages, we observed notable differences depending on whether the underlying kNN-graph was directed. Specifically, when using an undirected graph, the set of landmarks identified through random walks showed a high degree of overlap—up to 78%, with those selected simply by choosing high-degree nodes (hubs). However, this overlap decreases substantially to 47% when using a directed graph.

More critically, the class distribution of the selected landmarks is affected by the graph directionality. In datasets with class imbalance or small sample sizes, using directed graphs for random walk–based selection can result in the underrepresentation or complete omission of certain classes in the selected landmarks. This issue is visualized in Figure 4.4, which shows how the landmark class distributions vary across directed and undirected graphs for different datasets.

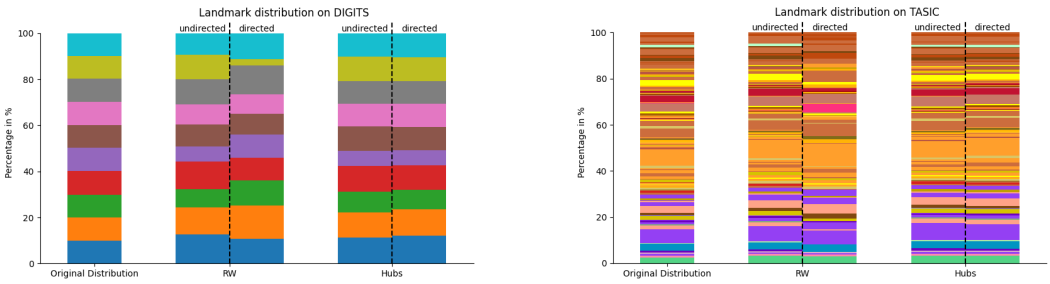

Figure 4.4.: Class distribution of landmarks for different datasets when using directed vs. undirected kNN-graphs.

Interestingly, hub-based landmark selection does not suffer from this underrepresentation, even when applied to directed graphs. Due to this robustness and its significantly lower computational cost, we choose to use hub-based selection rather than random walks in our subsequent evaluations. The impact of this decision on the embedding metrics is discussed in the following chapter. We should furthermore note that for the second level the landmark distribution became skewed for all sampling techniques. [1]

[1] Figure can be found in the Appendix at A.1

4.3.3. Landmark Connection Comparison

In this section, we compare different strategies for connecting landmarks after selection. Specifically, we evaluate three approaches: LL , SL approach and the Kron reduction. The Kron reduction is applied on undirected kNN-graphs, while SL and the LL operates on a directed kNN-graph.

The resulting embeddings at two coarsening levels are shown in Figure 4.5.

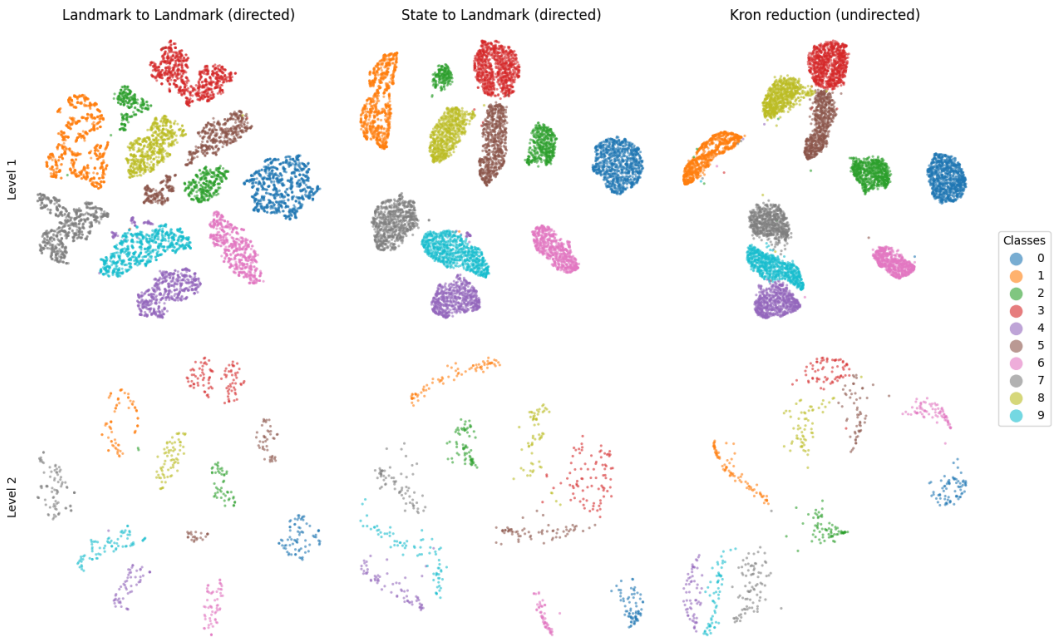

Figure 4.5.: Embeddings of MNIST kNN-graphs using different landmark connection strategies. Left to right: Hubs with Landmark-to-Landmark random walks (LL), State-to-Landmark approximation (SL) and Kron reduction. Top: First coarsening level; Bottom: Second coarsening level.

All three strategies produce well-separated clusters at both coarsening levels, indicating that landmark-based approaches can preserve global structure across reductions. At the first level, both random walk–based methods (LL and SL) show a fragmentation of the green cluster corresponding to digit 2. However, the purple cluster (digit 4) and the light blue cluster (digit 9) remain well-separated in both cases.

Kron reduction, by contrast, maintains the integrity of the green cluster representing digit 2 across both levels. It also preserves greater inter-cluster distances compared to the other methods, contributing to clearer spatial separation in the embedding.

In the second level, the SL embeddings are compact, with clusters increasing in density but also showing elongation. Interestingly, the SL and KR embed-

dings begin to resemble one another at this level, the SL embeddings now also recover the green cluster.

The LL method, on the other hand, produces an embedding that visually resembles those from the clustering methods. The clusters are less dense compared to SL, but the overall layout remains consistent across all approaches. Notably, the LL method also recovers the green cluster at the second level, despite its initial fragmentation.

Across all methods, the general layout and relative positions of the clusters remain stable, reinforcing the robustness of the landmarking framework in preserving both local and global structures.

4.3.4. Effect of kNN-graph Directionality on Landmark Connecting

An additional, and somewhat unexpected, finding arises when examining the effect of graph directionality in the SL connection strategy. Specifically, using a directed kNN-graph significantly improves the quality of the embeddings, particularly at the second level of coarsening. The corresponding embedding metrics for both coarsening levels are shown in Figure 4.6.

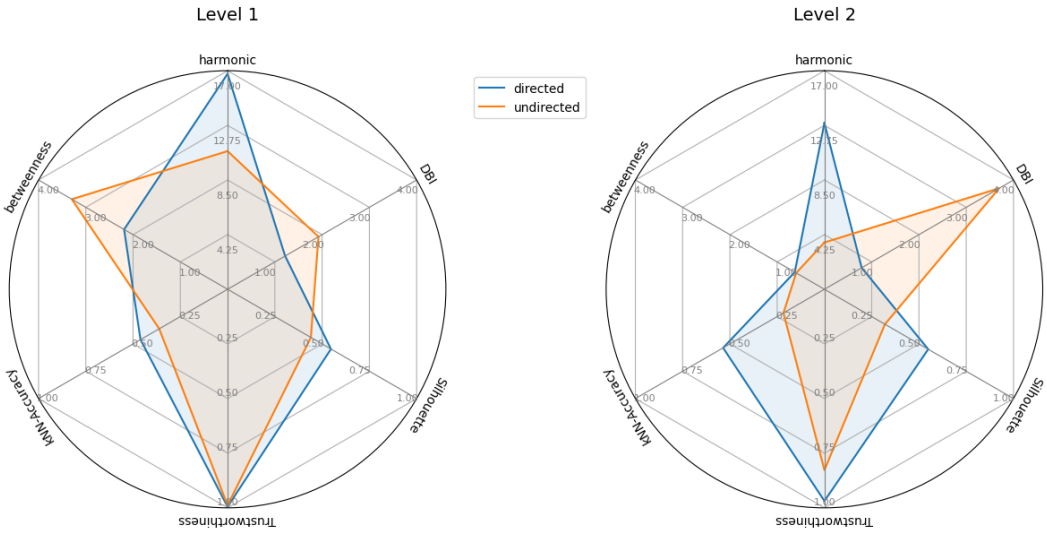

Figure 4.6.: Impact of graph directionality on the embedding quality of the State-to-Landmark approach for both levels. The metrics in the upper half should get minimized while the metrics in the lower half should get maximized.

The radar charts reveal that across all embedding metrics the directed graph outperforms the undirected one, especially at the second level. Note that, the

radar charts are not normalized; each axis maintains the same scale across both charts, enabling direct comparison between the levels.

Regarding structural preservation, the KL divergence values for harmonic centrality is slightly higher for the directed graphs. Nonetheless, we observe that cluster integrity, particularly at the second level, was better maintained when using directed graphs. In the undirected case, clusters became less distinguishable.

To further investigate whether this improvement generalizes beyond the SL method, we tested other Landmark techniques. [2] We found that only the LL approach also benefitted from using a directed kNN-graph. [3]

In summary, directed kNN-graphs tend to produce more compact and cohesive embeddings in SL and LL methods. However, they may introduce slightly more grainy structure compared to their undirected counterparts. This trade-off should be considered when selecting graph configurations for coarse-graining tasks.

4.4. Evaluating Coarse-Graining Approaches on a Complex Dataset

To assess the robustness and generalization of the coarse-graining methods, we apply them to the TASIC dataset—a challenging single-cell transcriptomic dataset. TASIC presents a higher level of difficulty than MNIST due to the presence of numerous cell types with highly similar expression profiles, which naturally introduces more biological noise.[4]

Figures 4.7 and 4.8 present visualizations of the embeddings with three different biological labels. The first row shows the major class, distinguishing between excitatory (Glutamatergic), inhibitory (GABAergic) neurons, and other cell types. The second row reflects finer subclass distinctions within these major categories, while the third row indicates the brain regions from which the cells were sampled.

In Figure 4.7, both SL approach and Kron reduction exhibit strong preservation of biological structure. At Level 1, major cell classes remain clearly separated, and subclasses form coherent subclusters. Even at Level 2, where substantial downsampling occurs, most clusters remain interpretable. Notably, the Kron method better preserves both global structure and local neighbor-

[2] Kron reduction was replaced by the Exact solution given by the absorbing markov chain solution

[3] Embeddings of the directed graph can be seen in Figure 4.5, and a figure of the embeddings on the undirected graphs is provided in the appendix, Figure A.2

[4] We also conducted control experiments on MNIST with increasing salt-and-pepper noise and observed similar degradation patterns.

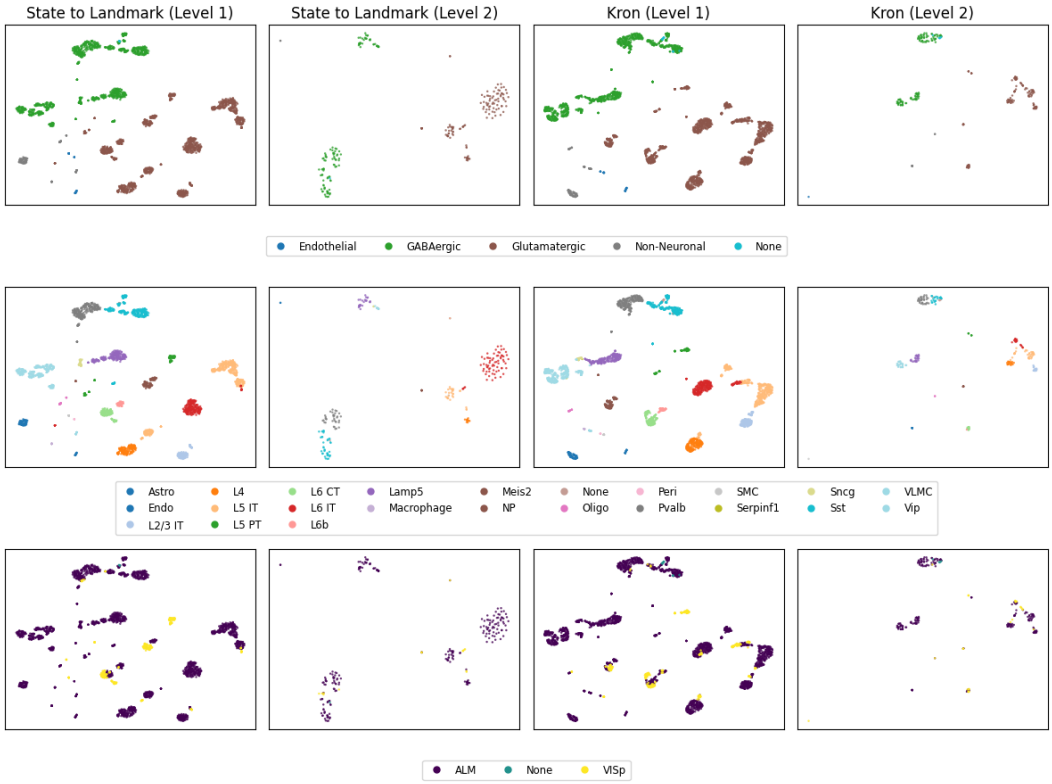

Figure 4.7.: Embeddings of the TASIC dataset using landmark-based methods: State-to-Landmark (left) and Kron reduction (right), across two hierarchy levels. First row visualizes the major classes of excitatory (GABAeric) and inhibatory (Glutameric) and other cell types. Each major class is characterized by a distinct set of subclasses which can be seen in the second row. In the last column the region where the sample was taken from is visualized, either the Anterior Lateral Motorcortex (ALM) or the primary visual cortex (VISp).

hood with minimal fragmentation. While the SL method emphasizes local neighborhood preservation, it can introduce distortions in the global geometry of the embedding. This is particularly visible in the subclass row, where the Kron reduction tends to produce more detailed internal structures, whereas the SL method favors forming more distinct clusters.

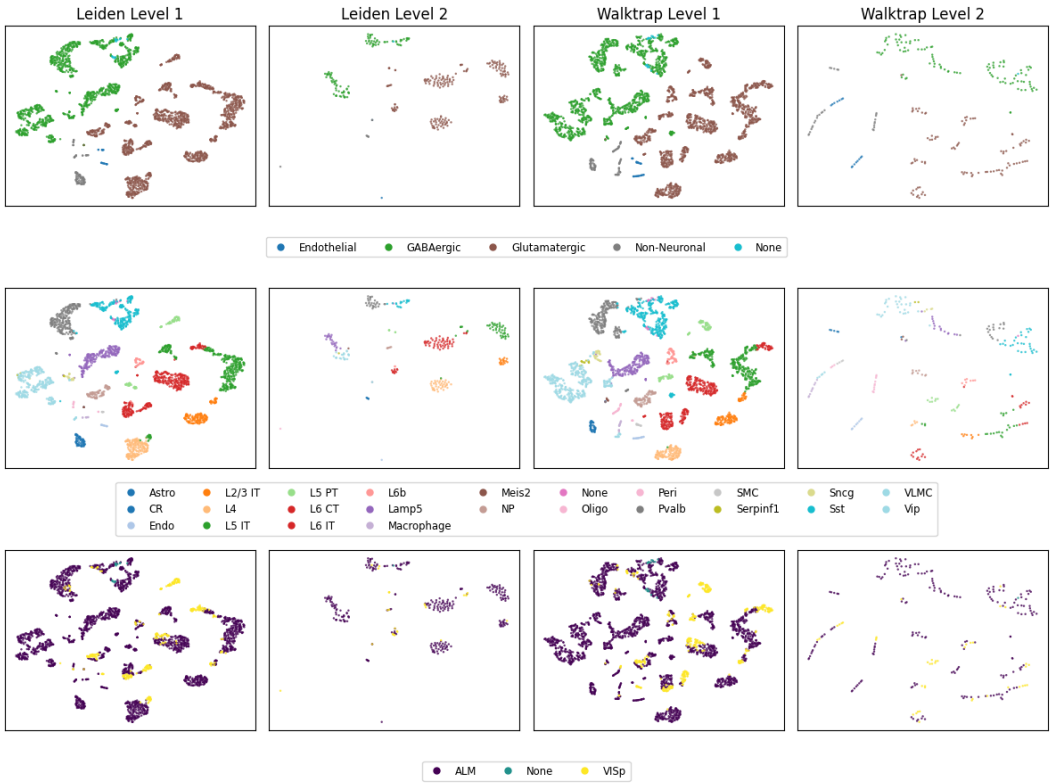

Figure 4.8.: Embeddings of the TASIC dataset using clustering-based methods: Leiden and Walktrap on undirected kNN-graphs, shown at both coarsening levels. Colouring according to major class, in first row, subclass in second row and brain region in the third row

Figure 4.8 presents the embeddings from clustering-based coarse graphs. Both Leiden and Walktrap yield well-structured clusterings at Level 1 that align well with known biological classes and subclasses. At Level 2, Leiden preserves the separation between major neuronal classes (e.g.,inhibatory (GABAergic) and excitatory (Glutamatergic)), while Walktrap focuses more on local consistency. Interestingly, Walktrap retains spatial information related to brain region more clearly, suggesting that the embedding structure might reflect anatomical organization.

Comparative Observations Across all methods and hierarchy levels, the visualizations confirm that hierarchical coarse-graining is capable of preserving

biologically meaningful structure. The SL and Leiden approaches offer the best trade-off between global abstraction and local resolution. SL maintains class and subclass integrity even after strong reduction, while Leiden yields compact clusters aligned with known types but occasionally loses subtype resolution at deeper levels. Kron reduction, tends to homogenize cell identities at Level 2.

In terms of quantitative metrics, the landmark-based methods outperform clustering-based approaches. SL achieves the highest kNN-Accuracy (0.78), followed closely by Kron (0.77), while Walktrap and Leiden reach 0.70 and 0.56, respectively. The kNN-Accuracies of the Landmarking approaches outperform those of the random subset which are at 0.64. Radar plots comparing additional metrics are provided in Figure A.3 in the appendix.

The landmark methods also show superior performance in trustworthiness and silhouette score. However, in terms of the KL divergence of betweenness, SL shows the highest deviation (1.23), whereas all other methods stay below 0.39. This highlights a trade-off: while SL excels in preserving embedding geometry, it may distort centrality-related properties more than clustering-based methods.

4.5. Learning the Manifold

Swissroll

In the SL approach, structural information from surrounding nodes is condensed into selected landmark nodes. We investigate whether this aggregation allows the embedding to preserve or even reveal manifold structure by applying both the SL and LL methods on classical manifold datasets, starting with the Swissroll. Furthermore, we investigate whether graph coarsening is more effective when performed in a single step or through a multi-step hierarchical process. Therefore we varied the reduction factor.
The results are shown in Figure 4.9, and additional visualization is included in the appendix for the LL approach (Figures A.4 and **??**).

Interestingly, the produced embeddings are similar independent of their hierarchy level.
When applying a mild reduction (e.g., factor of 0.25), the Swissroll retains its spiral structure, likely aided by PCA initialization. However, even without PCA, the structure persists, albeit less strongly.
At extreme reductions, the roll unfolds entirely into a straight line, evidence that SL retains global geometry. Notably, this effect was not observed in the LL approach. Although LL achieved partial unrolling at a reduction of 0.01, the structure remained curved.

Figure 4.10 shows quantitative embedding metrics over varying reduc-

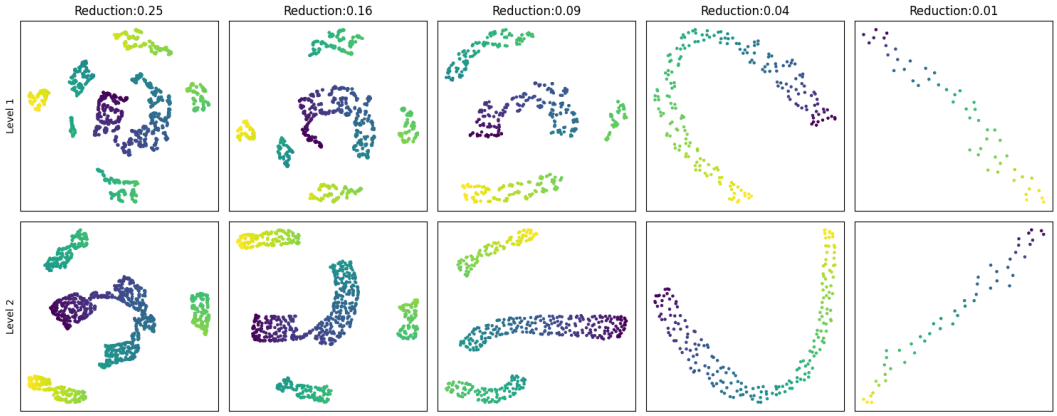

Figure 4.9.: Swissroll embedding with State-to-Landmark approach in one step (first row) and in two steps (second row) and varying reduction factors.

tion factors. Solid lines represent one-level reductions; dashed lines represent two-level reductions.

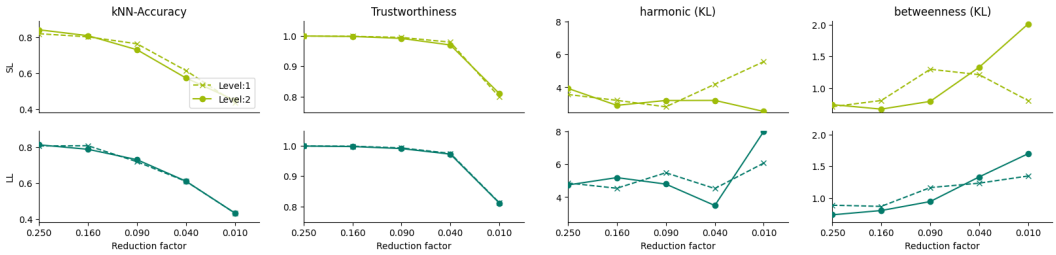

Figure 4.10.: Embedding metrics over increasing reduction for Swissroll. Dashed lines: two-level coarsening; solid lines: one-level.

Most metrics remained consistent between one and two levels. We visualized the distribution of landmarks, given by random walks along the Swissroll manifold. Figure 4.11 shows that landmarks tend to cluster in specific regions when using a directed graph. In directed graphs, random walks may become trapped in certain areas, leading to oversampling of local neighborhoods. In contrast, the undirected graph maintains a more even spread of landmarks.

We got similar placements for the landmarks selected via hubs, as given by the undirected random walks. This underlines the overlap of hubs sampling and random walks sampling.

DNA

The straight line found by the graph coarse-graining with the highest reduction factor may be shaped by other contributing factors. To verify that these

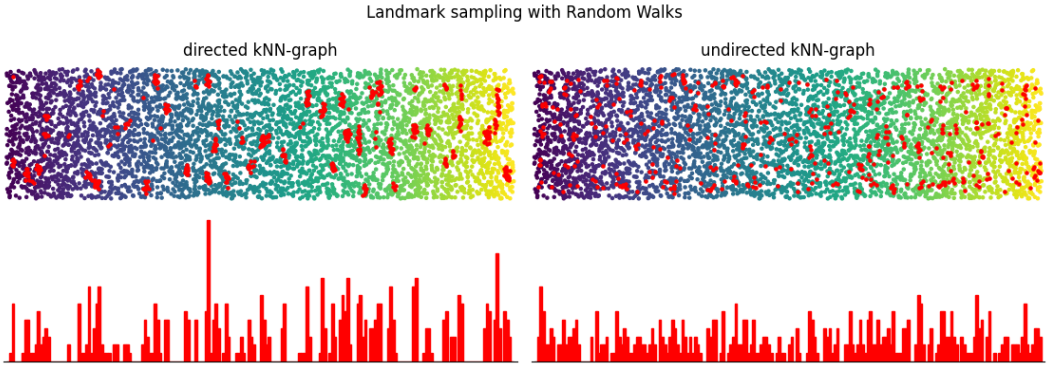

Figure 4.11.: Enrolled Swissroll with landmarks in red for the undirected and directed kNN-graph. Histograms show the distribution of landmark positions along the fold.

findings were not biased by the Swissroll's geometric regularity, we applied the same coarse-graining techniques in one step to the DNA manifold dataset. As shown in Figure 4.12, neither the full nor subsampled versions of the dataset produced an unwinded embedding. However, coarse-graining with landmark methods revealed repeating loop structures, which correspond to rungs in the DNA ladder structure. While complete unwinding was not achieved, the embedding reflect the meaningful periodicity.

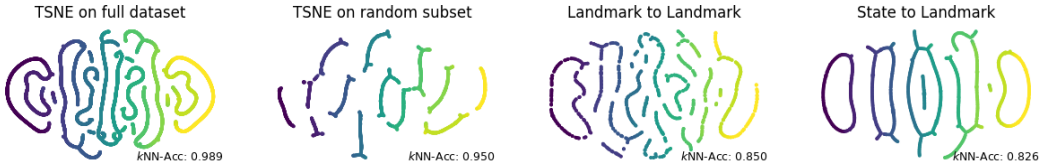

Figure 4.12.: Embeddings of the DNA dataset using Landmark-based coarse-graining approaches with one step and a reduction of 0.1.

Superimposing Non-Landmarks via Triangulation

Preliminary experiments explored how non-landmark nodes might be re-embedded after coarse-graining. We implemented a simple triangulation-based approach: each non-landmark was assigned to its three most probable landmarks, and its position inferred via weighted interpolation. The resulting embeddings, shown for the Swissroll in Figure A.3 and DNA in Figure A.3, demonstrate that fine-scale structure can be partially recovered in this way.

We also experimented starting t-SNE with the superimposed positions and restricting landmark position updates. However, this resulted in distorted embeddings. These approaches are exploratory and are not further evaluated in this thesis.

5. Discussion

5.1. Limitations of Local Variation Methods

In this section, we advise against the use of the Local Variation methods introduced by Loukas et al. [7] for the purpose of graph coarse-graining in embedding tasks. Across our experiments, these methods consistently produced inferior embeddings compared to simpler clustering-based approaches. With the sole exception of the relative eigenvalue error, all other evaluation metrics indicated that Local Variation underperforms significantly. This is particularly notable given that the alternative methods, such as Label Propagation, are algorithmically simpler but yield better preservation of structure and interpretability.

In our results, methods producing more visually and quantitatively consistent embeddings exhibited higher relative eigenvalue error, whereas Local Variation methods yielded a low eigenvalue error despite generating poor embeddings. This paradox may be explained by the objective of the Local Variation method itself: it explicitly attempts to preserve a predefined eigenspace. In our case, we followed the original paper's recommendation and used the first three eigenvectors. Consequently, the method minimizes deviation in only the first few eigenvalues, while ignoring the rest of the spectrum. This design choice is also reflected in the spectral distance, which remained high despite a low relative error in the preserved portion of the spectrum.

This limited eigenspace preservation becomes particularly problematic at the second level of coarsening. Since many similar nodes are already aggregated in the first level, subsequent fusions, guided only by the coarsest spectral structure, are likely to group boundary nodes from different classes. Without incorporating a broader eigenspace, the algorithm lacks the resolution needed to separate nuanced or overlapping structures. While increasing the number of preserved eigenvectors might mitigate this issue, it comes at a high computational cost, as noted by Loukas et al.

An additional explanation lies in the misalignment between the objective of preserving spectral properties and that of generating meaningful low-dimensional embeddings from kNN-graphs. As illustrated in Figure 2.3, certain eigenvectors may encode structural features that are not semantically relevant for the downstream task. For example, preserving a separation between nodes B and C may be less useful than maintaining a more salient boundary be-

tween H and I. This limitation is exacerbated by the greedy node pairing strategy, which can fuse semantically unrelated nodes if their spectral roles are indistinguishable in the limited eigenspace.

The Kron reduction method offers a useful point of comparison. Like Local Variation, it aims to preserve spectral structure, but does so in a way that results in both low spectral distortion and significantly better embedding quality. This further suggests that the shortcomings of Local Variation lie in its restricted eigenspace optimization and the one-shot fusion constraint per level. This constraint increases the likelihood of fusing dissimilar nodes when their optimal fusion partners are unavailable, a phenomenon especially visible in the edge-based variant of the algorithm. The neighborhood-based variant performed slightly better, likely due to its broader candidate set.

In conclusion, our results show no practical advantage of the Local Variation method over alternative approaches. Although it is theoretically grounded in spectral preservation, this focus does not translate into improved embedding quality in practice. Moreover, the implementation exhibits significant limitations: it constrains fusions to a single pass per level and relies on a narrow eigenspace that is insufficient for capturing finer structural distinctions. Additionally, we observed relatively high wall times even when preserving only three eigenvectors. Since the runtime scales cubically with the number of preserved eigenvectors—as stated by the authors—we expect substantially higher computational costs for deeper eigenspace preservation. This eliminates any potential computational advantage of Local Variation over alternative spectral methods such as Kron reduction, which achieved better embeddings at lower spectral distortion and similar or lower runtime.

5.2. Limitations of Label Propagation

Label Propagation performed notably better than Local Variation, particularly at the first level of coarse-graining. Despite its conceptual similarity to the edge-based Local Variation method—especially in the way nodes are merged based on connectivity—it achieved consistently superior cluster quality. Interestingly, it also yielded the lowest spectral distortion norm among all methods, even outperforming more computationally intensive approaches.

This behavior can be partially attributed to the way Label Propagation operates. The algorithm favors strong local affinities and tends to form equally sized clusters by assigning nodes to the most dominant neighboring label. As a result, it emphasizes local homogeneity, which is beneficial in datasets with balanced and well-separated classes. This likely explains its strong performance on MNIST and its comparability to Leiden and Walktrap in the first coarsening level.

However, this behavior does not generalize well to datasets with more complex, imbalanced structures or deeper level. On the second level in the MNIST dataset, the benefits of Label Propagation diminish. In multiple embeddings of first-level coarsened graphs, intertwined clusters, such as the purple and light blue classes of digits nine and five on MNIST, were consistently present. These overlaps likely contribute to the observed discrepancies in embedding metrics and indicate that Label Propagation may not always preserve class boundaries effectively. This highlights a key limitation: while the algorithm is effective for early abstraction steps, it lacks the sophistication needed for deeper hierarchical reduction.

Despite these limitations, Label Propagation remains highly efficient in terms of runtime and memory. With a time complexity of $\mathcal{O}(T(|V|+|E|))$ and a space complexity of $\mathcal{O}(|V|+|E|)$, where T is the number of iterations, $|V|$ the number of vertices, and $|E|$ the number of edges, the method is particularly well-suited to sparse kNN-graphs. This makes it appealing in scenarios where scalability is more important than fine-grained structural preservation.

In summary, while Label Propagation offers an efficient and surprisingly effective first-level reduction, we do not recommend its use for deeper hierarchical coarsening. Its lack of control over merge semantics and reduced adaptability to complex datasets limit its usefulness beyond the initial level.

5.3. Comparing Leiden and Walktrap Clustering

Across all datasets and levels, Leiden and Walktrap clustering methods produced highly similar embeddings—both in qualitative structure and in most quantitative metrics. Their ability to form compact, well-separated clusters makes them attractive choices for unsupervised learning tasks, especially in large-scale data settings. On the complex TASIC dataset on the second level the Leiden clustering was more similar to the Landmarking approaches, retaining some global structure.

In terms of runtime complexity, Leiden with its runtime complexity of $\mathcal{O}(n \log n)$ is more efficient than Walktrap, which has a complexity of $\mathcal{O}(n^2 \log n)$. However, in practice the runtime of Walktrap is often much lower, as this worst-case complexity only occurs if the algorithm merges each node with a cluster one step at a time. Moreover, Walktrap offers a built-in hierarchical clustering structure, which is particularly useful when different levels of granularity or varying reduction factors are required.

A key drawback of both Leiden and Walktrap is the lack of precise control over the number of clusters. While these methods can approximate a desired reduction factor, they cannot guarantee exact compression ratios. In contrast, landmark-based methods and Label Propagation allow for explicit control over

the number of nodes in the coarsened graph, offering greater flexibility in structured hierarchical designs.

Most importantly, we observe that the SL approach consistently outperforms both Leiden and Walktrap in critical embedding metrics. It achieves higher kNN accuracy, better silhouette scores, and stronger trustworthiness, particularly in deeper hierarchy levels. This suggests that while Leiden and Walktrap are strong general-purpose clustering methods, they are ultimately surpassed by SL in preserving the structure and semantics of the original data during coarse-graining.

5.4. Landmarking

Landmark-based approaches consistently outperformed clustering-based methods in terms of Silhouette scores and DBI, particularly in deeper hierarchy levels. This can be attributed to their tendency to preserve global distances, increasing inter-cluster separation and, in turn, improving these metrics. While this could be considered a form of "gaming" the metrics, the consistently high kNN-Accuracy across datasets suggests that the embeddings remain meaningful, not merely artifacts of metric inflation.

Interestingly, even the LL approach, which does not preserve global structure as explicitly as SL approach or Kron reduction, still achieves competitive Silhouette and DBI values compared to the SL methods. This suggests that landmark-based aggregation alone introduces structural regularity beneficial for downstream learning tasks. On the TASIC dataset in particular, SL and Kron reduction clearly outperformed other methods by preserving interpretable biological structure across multiple coarsening levels. Unlike Walktrap, which produced distortions, landmark-based methods maintained compact and well-separated clusters.

However, one drawback of landmark methods, especially in SL, emerged in the second coarsening level: a visible shift in the class distribution of remaining landmarks. We hypothesize that this is always the case but exaggerated by the use of hub-based landmark selection. While hubs (nodes with high in-degree) tend to represent structurally important regions in the graph and enhance embedding quality, they are also inherently biased. Hubs are more likely to appear in densely populated areas and are typically representative of majority classes [55]. As a result, class imbalance can affect the spatial distribution of selected landmarks.

This interpretation aligns with our observation that the distribution of hubs across classes is often proportional to the overall class distribution in the dataset. Hubs are therefore well-suited to represent clusters. Prior work by Radovanović et al. [55] also supports the idea that hubs conserve the intrinsic

variance of high-dimensional data, i.e. high dimensional data clusters. We further validated this with empirical results on MNIST which found that about 50 percent of the connections of hubs are to other hubs.

Given this, a natural question arises: why do random walks in directed kNN-graphs tend to avoid these hubs? We hypothesize that this is due to the asymmetric structure of directed graphs. In such graphs, it is possible for a random walk to reach a landmark in one direction (e.g., from cluster A to cluster B), but not return. This asymmetry can cause some clusters to be underrepresented, particularly in small or imbalanced datasets. The problem is mitigated in larger or more balanced datasets.

In the Swissroll dataset, it was observed that random walk–based landmark selection tends to become trapped in the directed kNN-graph. This effect was more pronounced than in the undirected graph, as reflected by the more uniform landmark distribution in the corresponding histogram, thereby strengthening this interpretation.

This raises the question: are landmark embeddings merely "cherry-picking" easy-to-learn data points, i.e., those in the dense center of clusters? We revisit this question in the manifold learning experiments, but preliminary observations suggest that landmark selection can indeed bias the representation space toward high-density regions.

Another interesting observation is the degraded performance of the SL method in the second coarsening level when using an undirected graph. In this case, SL underperforms not only compared to its directed variant, but also compared to the LL method. We rule implementation issues out as a cause for this observation since the random walk module is shared across all methods. We hypothesize that this behavior may stem from the random walk returning to the starting landmark due to the increased connectivity in the coarsened graph, especially when self-loops are present. This behavior is more likely in the second level, where fewer nodes and denser connections make backtracking more probable. This may also explain why errors tend to occur at the boundaries between neighboring classes. Furthermore it is explicitly forbidden for the LL method, which otherwise should also suffer from that.

Practical Considerations

All landmarking approaches differ significantly in computational complexity. Kron reduction requires solving a linear system, which has cubic runtime in the number of nodes. In contrast, the SL approach is claimed to be linear, based on the assumption that random walks terminate quickly due to locality. However, this assumption is not always justified. In the worst case, a random walk may take up to $N - n$ steps, where N is the number of total nodes and n the number of landmarks. Additionally, in pathological cases (e.g., components

without any landmark), random walks may fail to terminate altogether.

As such, without an explicit cap on random walk length, both SL and LL methods could exhibit worst-case complexity up to $\mathcal{O}(n^2)$. This complexity is still favorable compared to Kron but should be considered when applying these methods to large datasets.

Additionally, we recommend keeping the reduction factor below 0.4 for both SL and LL methods, as higher compression often results in overly sparse graphs and poor-quality embeddings.[1] This introduces a practical drawback: while landmark-based approaches theoretically allow for multiple levels of hierarchy, in practice, the levels must be spaced carefully to avoid excessive information loss. This constraint stands in contrast to clustering-based methods, which can more naturally accommodate arbitrary reductions without requiring such careful tuning between levels.

5.5. Manifold Learning

Throughout this thesis, we have seen that landmark-based methods are highly effective at producing meaningful low-dimensional representations—particularly by amplifying the inherent cluster structure in categorical datasets. The selection of hubs as landmarks might be viewed as a form of "cherry-picking," where points that are already central or easy to embed are favored. However, our experiments on manifold datasets, such as Swissroll and DNA, suggest that these approaches are capable of more than just emphasizing cluster boundaries.

On the Swissroll dataset, we found that the SL approach was particularly effective at capturing the global structure of the fold. Even though each node is typically connected to only a few neighbors in the sparse kNN-graph, the coarse-graining process was able to unfold the manifold structure meaningfully. To better understand this, we analyzed the weight matrix used in the coarsened graph and observed that many nodes received contributions from distant landmarks, that weren't in their immediate neighborhood. This indicates that the SL method is capable of propagating global structure through relatively few local connections.

These findings were further supported by experiments on the DNA dataset. While the full helical structure could not be entirely recovered, the SL approach was able to recreate loop-like structures that resemble the steps of the DNA ladder, something not observed with other methods. This suggests that the SL method, particularly in its exact formulation, is uniquely suited to capturing the geometry of high-dimensional manifolds.

[1] Beyond this threshold, embeddings tended to collapse into disorganized point clouds due to insufficient structural information.

We hypothesize that this capability stems from two factors: (1) the reduced number of points after coarse-graining, and (2) the global propagation of structural information. The first factor simplifies the optimization landscape for dimensionality reduction. Without coarse-graining, dense point clouds can introduce local minima that prevent proper unfolding. This is especially evident in the full DNA embedding: although PCA initialization provides a rough stripe layout, the embedding fails to close the loops due to these local minima. A similar limitation occurs in the Swissroll dataset, where uncoarsened blocks struggle to align into a straightened manifold.

In summary, coarse-graining, particularly via the SL method, has a demonstrably positive effect on revealing manifold structure. However, its limitations also become apparent in complex datasets like DNA, where full topological reconstruction remains out of reach. Strategies for improving manifold recovery are discussed in the Outlook section.

5.6. Implementation Challenges

Issues with Random Walks

During implementation, we encountered two major challenges. The first involved the execution of random walks for the SL method. Although prior work suggests that thousands of walks can be executed per millisecond on modern hardware [8, 9], we were unable to achieve similar performance. We developed our own random walk routine using parallelization and vectorized NumPy operations. Despite this, random walks remained the dominant runtime component during coarse-graining.

For example, on a Lenovo ThinkPad E480 with 8GB RAM and an 8-core 1.8 GHz processor, performing 7 million random walks took approximately 32 minutes. We believe more efficient implementations are possible and note a clear gap in available Python libraries: existing implementations in NetworkX, iGraph, and others typically use fixed-length walks, rather than terminating upon reaching a landmark. This limitation points to the need for a dedicated, optimized Python package for landmark-targeted random walks.

Difficulties with the MetaCell Method

The second issue arose in our attempt to reproduce the MetaCell-based coarsening method. Despite multiple efforts, our implementation consistently yielded disconnected graphs that did not align with the performance reported in the original publications.

Several factors likely contributed to this discrepancy. First, the original authors construct a custom similarity graph from their single cell dataset, which

may not generalize to standard kNN-graphs on other datasets, like MNIST. Second, the method relies on numerous hyperparameters that are either not disclosed or only vaguely mentioned.

Moreover, the graph sampling process required by MetaCell was extremely slow. Sampling just 50 subgraphs took over 40 minutes, and evidence suggests that hundreds of samples (e.g., 500 in the original study) are necessary for a stable and connected coarse-grained graph. This makes hyperparameter tuning computationally prohibitive. As a result, we discontinued our evaluation of MetaCell due to impractical runtime requirements and lack of reproducibility.

Limits of Graph Metrics in Predicting Embedding Quality

We encountered numerous cases in which graphs exhibiting high KL divergence in centrality distributions nonetheless produced high-quality embeddings—while graphs with low divergence occasionally led to structurally incoherent projections. This inconsistency is, largely, a consequence of differing sparsity patterns and weighting schemes across coarsening methods. Each approach induces its own topology, with distinct edge weights and connectivity profiles, leading to direct comparison between centrality-based descriptors being unreliable.

This limitation is particularly pronounced in graphs constructed with probabilistic edge weights, such as those derived from random walk transition probabilities. In such cases, shortest path metrics, which form the basis of many centrality measures—become unintuitive: the shortest path corresponds to the least probable sequence of transitions. To resolve this, we propose a transformation of the edge weights prior to applying shortest-path–based metrics. Specifically, subtracting each weight from one and applying a logarithmic transformation converts high-probability paths into low-cost paths. Aligning the shortest path with the most probable route. This also better reflects the multiplicative structure of path probabilities in stochastic processes. Moreover, since centrality is intrinsically linked to spectral properties (as in PageRank [56]), this insight offers a plausible explanation for the poor alignment between spectral metrics and embedding quality in practice.

On the Use of Embedding Metrics

While more interpretable, embedding metrics are not without limitations. Metrics such as the Silhouette score and the DBI rely on ground truth labels to define cluster structure. However, these labels do not necessarily align with the latent organization discovered by unsupervised embedding techniques. For example, in the MNIST dataset, we observed that the digit 2 cluster was split based on subtle stylistic variations—specifically, the presence or absence

of a loop at the bottom of the digit. While this split reflects semantically valid structure, it is penalized by metrics that assume a single label-based cluster.

This problem becomes more pronounced in complex datasets such as FMNIST, where intra-class variation gives rise to distinct substructures. As shown in Figure 5.1, new clusters emerge in the embedding that are not captured by the original labels.

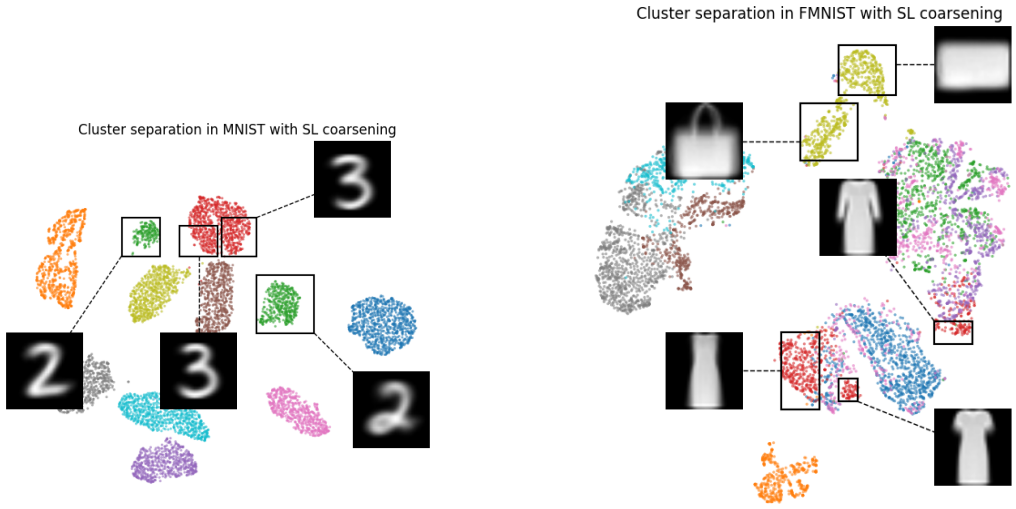

Figure 5.1.: Additional clusters found in embeddings, with a mean image of their respective datapoints.

These emergent patterns, despite being meaningful as seen by the visualisation of the mean image, are treated as fragmentation and receive poor scores from clustering-based metrics. A similar example arises in MNIST, where a "crack" in the red cluster (corresponding to digit 3) on the MNIST dataset reflects a structural refinement but is penalized nonetheless.

Another example is the trustworthiness metric, which measures the fidelity of local neighborhood preservation. Although theoretically appealing, we found it to be consistently high across all methods and datasets, offering little practical discrimination. As an alternative, we suggest adopting more sensitive metrics such as *mean relative rank error*, which may better capture the distortion of local geometry.

Among all embedding metrics, kNN-Accuracy emerged as the most robust and reliable indicator. By quantifying the overlap between neighborhoods in the high-dimensional and embedded spaces, it directly reflects the preservation of local structure. Nevertheless, it too has limitations. The metric is biased toward tightly packed clusters and may penalize embeddings that uncover more complex or hierarchical relationships. This was evident in the TASIC dataset, where kNN-Accuracy remained informative, yet failed to fully capture the advantages of methods that revealed latent hierarchical structure.

6. Outlook

This thesis has investigated a range of methods for coarse-graining kNN-graphs. Our findings show that no single approach is universally optimal; rather, the choice of algorithm must be guided by the user's specific objectives—whether to preserve global distances, ensure scalability, or support hierarchical interpretation.

Based on our results, we propose the following guidelines:

- For applications prioritizing the preservation of **global structure** and manifold geometry, the State-to-Landmark approach, with either exact solutions or directed random walks, is most effective.
- When **efficiency** is the primary concern and fine structural detail is of secondary importance, the Landmark-to-Landmark method offers a lightweight and scalable solution.
- For use cases that require **hierarchical organization** or intermediate representations, Walktrap clustering provides a natural multi-resolution decomposition, particularly effective for small to medium datasets.

While these methods offer complementary strengths, the evaluation of coarsened graphs and their embeddings remains a crucial challenge. No single metric fully captures both the preservation of graph structure and the quality of the resulting embedding. Graph-based metrics, though grounded in theory, often fail to align with perceptual or task-relevant notions of embedding quality. Conversely, label-based embedding metrics may penalize emerging structure not present in the original annotations. To improve evaluation, future work should explore more principled approaches,such as reweighted centrality-based KL divergence using logarithmic transformations, and manifold-aware metrics that assess geometric continuity in unsupervised settings.
Several research directions emerge from this work:

Hybrid Methods Combining graph coarse-graining strategies could yield improved trade-offs between efficiency and accuracy. For example, Label Propagation could be used to identify broad clusters, which are then refined using Kron reduction or SL methods. Such combinations may offer robustness while preserving global relationships.

Multi-Level Embeddings Our results, particularly on the TASIC dataset, suggest that different reduction levels capture complementary structural information. Future research could explore how to combine embeddings across coarse-graining levels into unified, interpretable representations. Initial steps in this direction were taken by superimposing non-landmark points, but more principled techniques are needed.

Graph Comparison Metrics A pressing need remains for metrics that compare graphs of differing sizes while remaining sensitive to both global and local structure. Our proposed transformation of centrality measures is one possible avenue, though alternative metrics should be explored and rigorously evaluated.

Manifold-Aware Metrics Especially in unsupervised contexts, metrics that assess the fidelity of manifold structure in the embedding space could provide a more meaningful basis for evaluation than label-based or purely spectral metrics.

Role of Graph Directionality Finally, the influence of directionality in kNN-graphs, particularly in SL and LL methods, warrants further investigation. Our observations suggest it plays a critical role in embedding behavior, especially at deeper levels of coarsening.

In summary, this thesis has illuminated the interplay between graph structure, coarse-graining, and low-dimensional embedding. It provides a comparative evaluation of key methods, highlights open challenges in metric design, and offers practical recommendations for method selection, by identifying concrete paths for further development.

A. Appendix

A.1. Further Proofs

Derivation pseudoinverse of P in local variation algorithm is its transpose.

$$
\begin{aligned}
\begin{pmatrix}
\underbrace{\frac{1}{\sqrt{a}}, \cdots, \frac{1}{\sqrt{a}}}_{a \text{ repeats}} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \underbrace{\frac{1}{\sqrt{z}}, \cdots, \frac{1}{\sqrt{z}}}_{z \text{ repeats}}
\end{pmatrix} &= \mathbf{P} \\
\mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1} &= \mathbf{P}^+ \\
\mathbf{P}^T\mathbf{I} &= \mathbf{P}^+ \\
\mathbf{P}^T &= \mathbf{P}^+ \\
(\mathbf{P}\mathbf{P}^T)^{-1} &= \mathbf{I} \\
\begin{pmatrix}
\underbrace{\frac{1}{\sqrt{a}} \cdot \frac{1}{\sqrt{a}} + \cdots + \frac{1}{\sqrt{a}} \cdot \frac{1}{\sqrt{a}}}_{a \text{ repeats}} & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \underbrace{\frac{1}{\sqrt{z}} \cdot \frac{1}{\sqrt{z}} + \cdots + \frac{1}{\sqrt{z}} \cdot \frac{1}{\sqrt{z}}}_{z \text{ repeats}}
\end{pmatrix} &= \mathbf{P}\mathbf{P}^T \\
\begin{pmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & 1
\end{pmatrix} &= \mathbf{P}\mathbf{P}^T
\end{aligned}
\tag{A.1}
$$

Derivation of the effective Laplacian in Kron reduction.

$$
\begin{aligned}
\mathbf{L}\boldsymbol{v} &= \mathbf{0} \\
\begin{pmatrix} \mathbf{L}_{ll} & \mathbf{L}_{ln} \\ \mathbf{L}_{nl} & \mathbf{L}_{nn} \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_l \\ \boldsymbol{v}_n \end{pmatrix} &= \mathbf{0} \\
\mathbf{L}_{ll}\boldsymbol{v}_l + \mathbf{L}_{ln}\boldsymbol{v}_n &= \mathbf{0} \\
\mathbf{L}_{nn}\boldsymbol{v}_n + \mathbf{L}_{nl}\boldsymbol{v}_l &= \mathbf{0} \\
-\mathbf{L}_{nn}^{-1}\mathbf{L}_{nl}\boldsymbol{v}_l &= \boldsymbol{v}_n \\
\mathbf{L}_{ll}\boldsymbol{v}_l + \mathbf{L}_{ln}(-\mathbf{L}_{nn}^{-1}\mathbf{L}_{nl}\boldsymbol{v}_l) &= \mathbf{0} \\
(\mathbf{L}_{ll} - \mathbf{L}_{ln}\mathbf{L}_{nn}^{-1}\mathbf{L}_{nl})\boldsymbol{v}_l &= \mathbf{0} \\
\mathbf{L}_{ll} - \mathbf{L}_{nl}\mathbf{L}_{nn}^{-1}\mathbf{L}_{nl} &= \mathbf{L}'
\end{aligned}
\tag{A.2}
$$

A.2. Further Tables

SLRW	kNN-Accuracy	Trustworthiness	Silhouette	DBI	harmonic	betweenness	rel. eigenerr	dspectral
Hubs	**0.362**	0.989	0.441	1.924	10.746	3.296	3.144	83.249
RW	0.358	**0.991**	**0.490**	**0.844**	11.683	2.823	3.105	83.104
HBN	0.174	0.949	0.407	1.053	**8.218**	1.389	2.810	81.892
Random	0.214	0.953	0.440	0.900	10.487	2.646	2.923	82.299
PPS	0.174	0.949	0.407	1.048	8.552	**1.332**	**2.795**	**81.8919**

Table A.1.: Table of metrics for landmark selection State to Landmark

Level 1	harmonic	betweenness	kNN-Accuracy	Trustworthiness	Silhouette	DBI
MNIST	19.494(±1.351)	0.729(±0.273)	0.479(±0.003)	0.986(±0.001)	0.232(±0.016)	2.898(±0.286)
TASIC	21.820(±0.251)	0.443(±0.019)	0.677(±0.005)	0.998(±0.000)	0.255(±0.019)	2.244(±0.263)
FMNIST	3.436(±0.108)	6.677(±0.129)	0.469(±0.003)	0.990(±0.00)	0.145(±0.008)	2.364(±0.058)
Level 2						
MNIST	13.915(±1.603)	2.296(±0.155)	0.575(±0.006)	0.96(±0.003)	0.160(±0.024)	12.571(±19.613)
TASIC	11.708(±0.544)	0.925(±0.136)	0.850(±0.005)	0.994(±0.001)	0.103(±0.057)	1.153(±0.355)
FMNIST	3.383(±0.271)	9.881(±0.197)	0.603(±0.014)	0.981(±0.003)	0.153(±0.042)	2.743(±0.540)

Table A.2.: Average metrics with standard deviation of the random subset with directed kNN graph

A.3. Further Figures

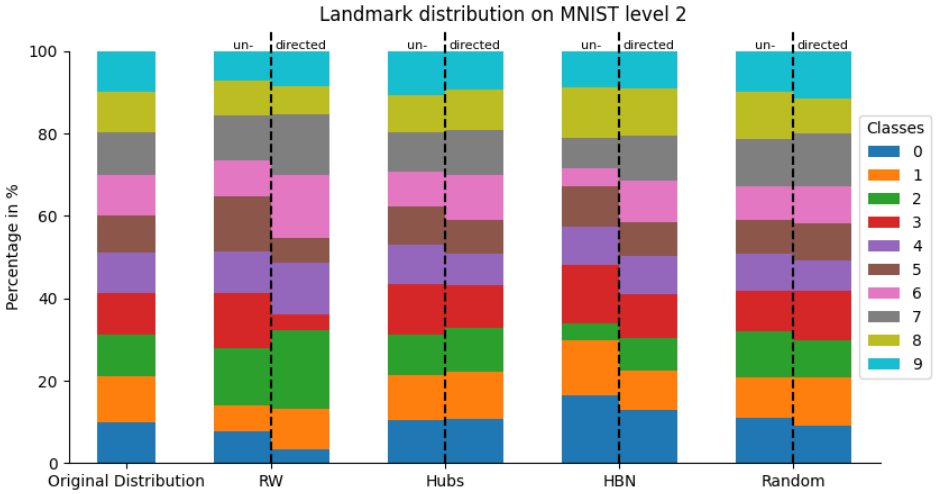

Figure A.1.: Distribution of landmarks in MNIST at the second coarse-graining level

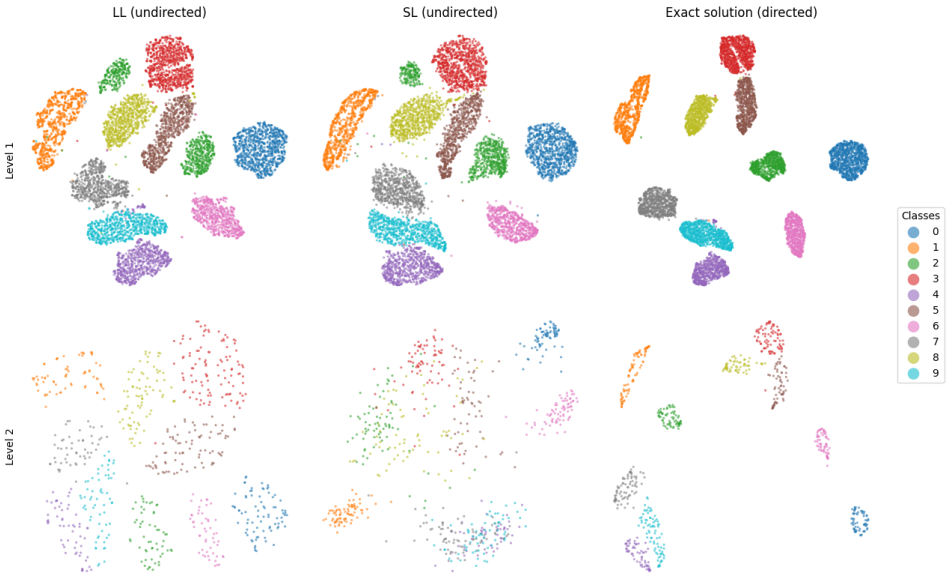

Figure A.2.: MNIST embeddings different directionality than figure 4.5

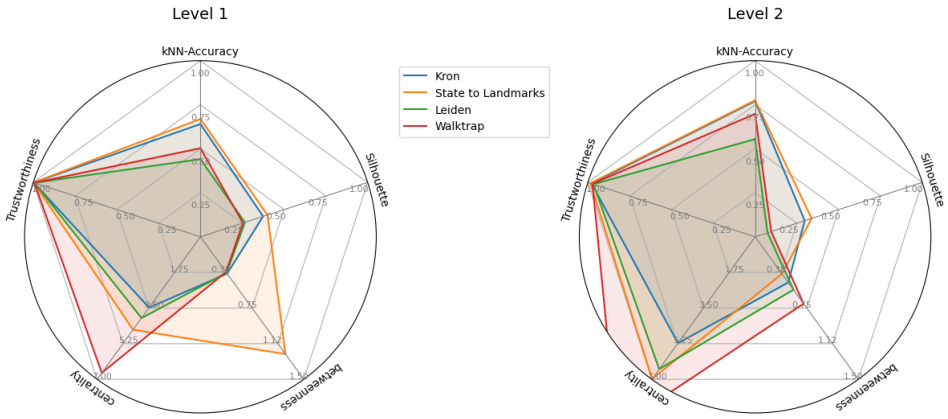

Figure A.3.: Radar chart of the metrics for the method comparison on the TASIC dataset

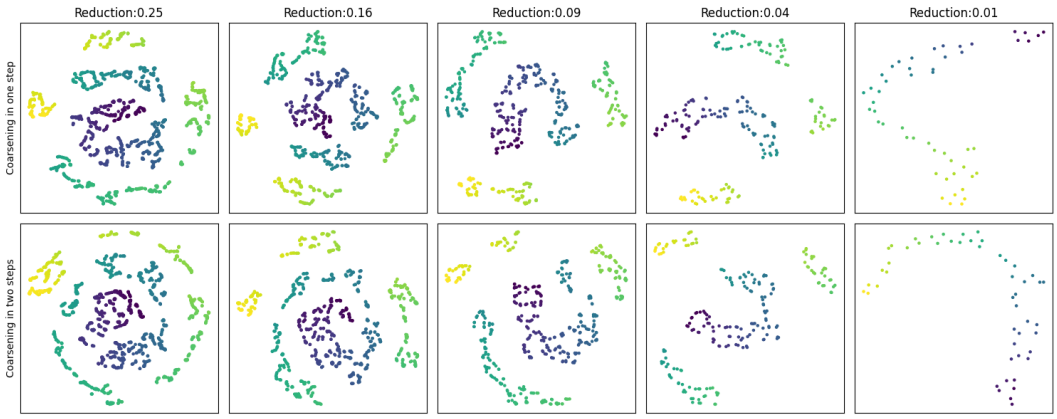

Figure A.4.: Swissroll embedding with Landmark to Landmark approach for both level with varying reduction

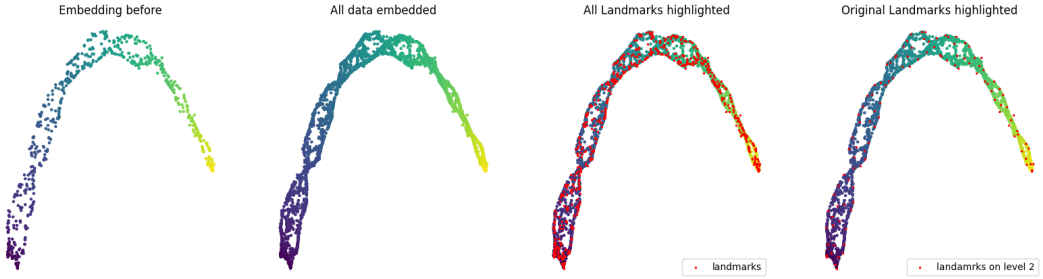

Figure A.5.: Embedding of Swissroll with superimposed positions for non-landmarks

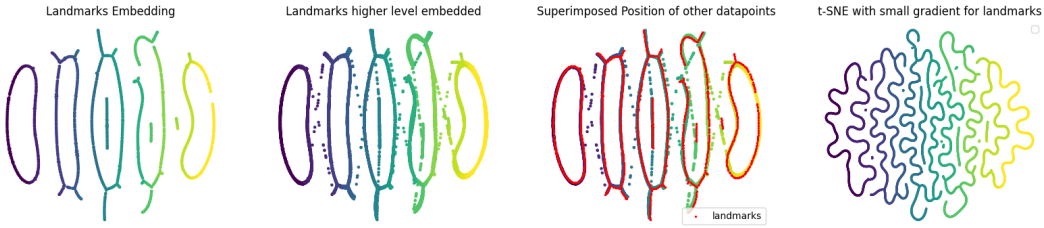

Figure A.6.: Embedding of DNA dataset with superimposed positions for non-landmarks

Bibliography

[1] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph theory*. 2008.

[2] Vukan R Vuchic. *Urban transit systems and technology*. 2007.

[3] Jean Anne Hayes-Williams. The earliest dated tree of jesse image: thematically reconsidered. *Athanor*, 2000.

[4] Stanley Wasserman and Katherine Faust. Social network analysis: Methods and applications. 1994.

[5] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, pages 395–416, 2007.

[6] Rachid Guerraoui and Anne-Marie et al. Kermarrec. Smaller, faster & lighter knn graph constructions. In *Proceedings of The Web Conference 2020*, pages 1060–1070, 2020.

[7] Andreas Loukas and Pierre Vandergheynst. Spectrally approximating large graphs with smaller graphs. In *International conference on machine learning*, pages 3237–3246, 2018.

[8] Nicola Pezzotti and Thomas et al. Höllt. Hierarchical stochastic neighbor embedding. In *Computer graphics forum*, pages 21–30, 2016.

[9] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

[10] James R Norris. *Markov chains*. Number 2. 1998.

[11] Jie Chen and Haw-ren et al. Fang. Fast approximate knn graph construction for high dimensional data via recursive lanczos bisection. *Journal of Machine Learning Research*, 2009.

[12] Bojan Mohar and Yousef et al. Alavi. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, page 12, 1991.

[13] Bojan Mohar. Some applications of laplace eigenvalues of graphs. In *Graph symmetry: Algebraic methods and applications*, pages 225–275. 1997.

[14] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, pages 298–305, 1973.

[15] Andrew Ng and Michael et al. Jordan. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2001.

[16] Pierre Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. 2013.

[17] John Kemeny and Laurie et al. Snell. *Finite markov chains*, volume 26. 1969.

[18] Julian Lowell Coolidge. The gambler's ruin. *Annals of Mathematics*, pages 181–192, 1909.

[19] Aggelos Katsaggelos and Richard et al. Kleihorst. Adaptive image sequence noise filtering methods. In *Visual Communications and Image Processing'91: Image Processing*, pages 716–727, 1991.

[20] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, page 417, 1933.

[21] Christopher Bishop and Nasser Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.

[22] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of research of the National Bureau of Standards*, pages 255–282, 1950.

[23] Martin Wattenberg and Fernanda et al. Viégas. How to use t-sne effectively. *Distill*, page e2, 2016.

[24] Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *Nature communications*, page 5416, 2019.

[25] Charu Aggarwal and Chandan Reddy. *Data Clustering: Algorithms and Applications*. CRC Press, 2013.

[26] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. *ProQuest number: information to all users*, 2002.

[27] Vincent Traag and Ludo et al. Waltman. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, pages 1–12, 2019.

[28] Vincent Blondel and Jean-Loup et al. Guillaume. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, page P10008, 2008.

[29] Mark Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, pages 8577–8582, 2006.

[30] Ulrik Brandes and Daniel et al. Delling. Maximizing modularity is hard. *arXiv preprint physics/0608255*, 2006.

[31] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *Journal of graph algorithms and applications*, pages 191–218, 2006.

[32] Andreas Loukas. Graph reduction with spectral and cut guarantees. *Journal of Machine Learning Research*, pages 1–42, 2019.

[33] Daniel Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 563–568, 2008.

[34] Dehua Peng and Zhipeng et al. Gui. Scalable manifold learning by uniform landmark sampling and constrained locally linear embedding. *arXiv preprint arXiv:2401.01100*, 2024.

[35] Florian Dorfler and Francesco Bullo. Kron reduction of graphs with applications to electrical networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 150–163, 2012.

[36] Bayu Jayawardhana and Shodhan et al. Rao. Handling biological complexity using kron reduction. In *Mathematical Control Theory I: Nonlinear and Hybrid Control Systems*, pages 73–93, 2015.

[37] Jie Chen and Yousef et al. Saad. Graph coarsening: from scientific computing to machine learning. *SeMA Journal*, pages 187–223, 2022.

[38] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.

[39] Linton Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[40] Ulrik Brandes and Christian Pich. Centrality estimation in large networks. *International Journal of Bifurcation and Chaos*, pages 2303–2318, 2007.

[41] Linton Freeman. Centrality in social networks: Conceptual clarification. *Social network: critical concepts in sociology. Londres: Routledge*, pages 238–263, 2002.

[42] Murray Beauchamp. An improved index of centrality. *Behavioral science*, pages 161–163, 1965.

[43] Stanley Wasserman. Social network analysis: Methods and applications. *The Press Syndicate of the University of Cambridge*, page 201, 1994.

[44] Solomon Kullback and Richard Leibler. On information and sufficiency. *The annals of mathematical statistics*, pages 79–86, 1951.

[45] Irena Jovanović and Zoran Stanić. Spectral distances of graphs. *Linear Algebra and its Applications*, pages 1425–1435, 2012.

[46] John Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, pages 1431–1443, 2009.

[47] David Davies and Donald Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, pages 224–227, 1979.

[48] Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, pages 53–65, 1987.

[49] John Pavlopoulos and Georgios et al. Vardakas. Revisiting silhouette aggregation. In *International Conference on Discovery Science*, pages 354–368, 2024.

[50] Ethem. Alpaydin and Cenk. Kaynak. Optical Recognition of Handwritten Digits. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C50P49.

[51] Yann LeCun and Corinna et al. Cortes. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[52] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[53] Bosiljka Tasic and Vilas et al. Menon. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature neuroscience*, pages 335–346, 2016.

[54] Stephen Marsland. *Machine learning: an algorithmic perspective*. 2011.

[55] Milos Radovanovic and Alexandros et al. Nanopoulos. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, pages 2487–2531, 2010.

[56] Lawrence Page and Sergey et al. Brin. The pagerank citation ranking: Bringing order to the web. Technical report, 1999.