Measuring and Explaining the Inter-Cluster Reliability of Multidimensional Projections

Hyeon Jeon, Hyung-Kwon Ko, Jaemin Jo, Youngtaek Kim, and Jinwook Seo

Abstract— We propose Steadiness and Cohesiveness, two novel metrics to measure the inter-cluster reliability of multidimensional projection (MDP), specifically how well the inter-cluster structures are preserved between the original high-dimensional space and the low-dimensional projection space. Measuring inter-cluster reliability is crucial as it directly affects how well inter-cluster tasks (e.g., identifying cluster relationships in the original space from a projected view) can be conducted; however, despite the importance of inter-cluster tasks, we found that previous metrics, such as Trustworthiness and Continuity, fail to measure inter-cluster reliability. Our metrics consider two aspects of the inter-cluster reliability: Steadiness measures the extent to which clusters in the projected space form clusters in the original space, and Cohesiveness measures the opposite. They extract random clusters with arbitrary shapes and positions in one space and evaluate how much the clusters are stretched or dispersed in the other space. Furthermore, our metrics can quantify pointwise distortions, allowing for the visualization of inter-cluster reliability in a projection, which we call a reliability map. Through quantitative experiments, we verify that our metrics precisely capture the distortions that harm inter-cluster reliability while previous metrics have difficulty capturing the distortions. A case study also demonstrates that our metrics and the reliability map 1) support users in selecting the proper projection techniques or hyperparameters and 2) prevent misinterpretation while performing inter-cluster tasks, thus allow an adequate identification of inter-cluster structure.

Index Terms—Multidimensional projections, MDP distortions, Inter-cluster tasks, Inter-cluster reliability, Distortion metrics

---

1 INTRODUCTION

Can we truly trust the clusters revealed by multidimensional projections (MDP)? One way to understand high-dimensional data in various domains is to reduce its dimensionality by MDP and thoroughly check the projection in a lower-dimensional space. However, distortions inherently occur when reducing dimensionality. Such distortions can make meaningful patterns in projections less trustworthy and can disturb users' accurate comprehension of the original data, leading to interpretation bias [49]. Therefore, it is important to measure the overall distortions using quantitative metrics [22, 49], or to visualize where and how the distortions occurred in the projection [31, 49], which are as important as generating a good projection in the MDP analysis.

This work was motivated by the following issue: although inter-cluster tasks, which investigate the inter-cluster structures of a given dataset (i.e., how clusters are located and related) through its 2D projection, have been regarded as the core tasks [37, 59] for using MDP, only a few previous metrics have tried to explain the distortions of inter-cluster structure represented in MDP thus far. Through a literature review (Sect. 3.1), we organized three types of inter-cluster tasks: 1) identifying clusters, 2) seeking the relationship between clusters, and 3) comparing clusters of the original data based on the projected representation. When performing these inter-cluster tasks, users must be aware of the preservation of the inter-cluster structure in the MDP. We refer to this as the *inter-cluster reliability*: how well the inter-cluster structures are preserved in the MDP. Several previous approaches have tried to reduce the loss of inter-cluster reliability during projection [21, 43, 47, 50, 51] or to explain it through visualizations [12, 35, 42]. However, measuring reliability is a challenging problem, as real-world datasets' inter-cluster structures often have no ground truth, and thus the metrics that quantify ground truth reliability cannot exist. Still, this research introduces new metrics to measure the loss of inter-cluster reliability by quantifying the preservation of randomly extracted clusters and validated their effectiveness through experiments.

Previous local quality metrics for MDP focused on measuring point-point distortion [8, 67] or cluster-point distortion [44], or quantified

the preservation of predefined clusters (e.g., labels of points) [20, 26]. However, all three have difficulties measuring inter-cluster reliability. Point-point distortion describes a distortion that occurs in the distance between points. Therefore, point-point distortion metrics take an intra-cluster distortion into account as the points within the cluster are close together but largely dismiss inter-cluster distortion as the points from different clusters are far away. Cluster-point distortion metrics instead describe a distortion that occurs between a cluster and its nearby points in both the original and projected spaces, and thus cannot consider complex inter-cluster structures that consist of multiple clusters, as they consider each cluster individually. Quantifying the preservation of the predefined clusters is also not an adequate measure for inter-cluster reliability, as many real-world datasets have no ground truth clusters.

In this paper, we propose *Steadiness* and *Cohesiveness*, two metrics that quantitatively evaluate inter-cluster reliability. Steadiness measures the inter-cluster reliability in the projected space (i.e., to what degree the cluster in the projection is in a steady state that reflects the actual cluster in the original space). Cohesiveness measures the inter-cluster reliability in the original space (i.e., to what degree real clusters in the original space stand together cohesively in the projection). We first formulated three design considerations of Steadiness and Cohesiveness so that the metrics can adequately evaluate inter-cluster reliability by quantifying how well inter-cluster tasks can be performed accurately in MDPs (i.e., measure the potential accuracy of the inter-cluster tasks). To make our metrics fulfill the considerations, we defined new inter-cluster distortion types—False Groups and Missing Groups—and designed Steadiness and Cohesiveness to measure each, respectively. Our metrics measure inter-cluster reliability by repeatedly extracting random clusters from one space and quantifying how much the clusters have been stretched or dispersed in the opposite space.

We also developed a reliability map to visualize inter-cluster reliability quantified by Steadiness and Cohesiveness within projections. By aggregating the underlying distortion information of each data point summarized in our metrics, the reliability map shows where and how the inter-cluster distortion occurred.

Through quantitative and qualitative experiments, we confirmed the effectiveness and usefulness of our metrics. The quantitative experiments verified that our metrics accurately capture inter-cluster distortions and thus properly measure inter-cluster reliability, while baseline local metrics, such as Trustworthiness & Continuity, failed to identify even apparent distortions. Furthermore, a qualitative case study showed that our metrics and the reliability map support a precise identification of the inter-cluster structure of the original space by helping users choose the proper projections and perform the inter-cluster tasks while

---

• *Hyeon Jeon, Hyung-Kwon Ko, Youngtaek Kim, Jinwook seo are with Seoul National University. E-mail: {hj, hkko, ytaek.kim}@hcil.snu.ac.kr, jseo@snu.ac.kr.*
• *Jaemin Jo is with Sungkyunkwan University. E-mail: jmjo@skku.edu.*

being aware of misinterpretations.

2 BACKGROUND AND RELATED WORK

2.1 MDP Distortions

Many MDP techniques, such as t-SNE [36], Isomap [65], and UMAP [41], have been proposed to understand and visualize high-dimensional data[1]; however, every MDP produces distortions because information loss is inevitable when dimensionality is reduced.

2.1.1 Distortion Types

In his seminal work [2], Michaël Aupetit defined two types of MDP distortion: stretching and compression. Stretching occurs when pairwise distances in the projected space are expanded compared to the original pairwise distances, and compression does the opposite. Afterward, Missing Neighbors and False Neighbors [31, 32] distortion types were introduced to interpret the stretching and compression in the context of neighborhood preservation. Let $f : X \to Y$ be a smooth mapping where $X = \{x_i \in \mathscr{R}^D, i = 1, 2, \ldots, N\}$ and $Y = \{y_i \in \mathscr{R}^d, i = 1, 2, \ldots, N\}$ for some $D > d$. Each data point p_i has its high-dimensional coordinate, x_i, and the corresponding low-dimensional coordinate y_i. For any point p_i, its k neighbors in the projected and original spaces are denoted as C_{y_i} and C_{x_i}, respectively. Missing Neighbors are then defined as $C_{x_i} \setminus C_{y_i}$. Similarly, False Neighbors are defined as $C_{y_i} \setminus C_{x_i}$ (Fig. 1a); however, our literature review (Sect. 3.1) indicated that measuring Missing and False Neighbors distortion cannot reflect how well inter-cluster tasks can be performed, and thus cannot correctly evaluate inter-cluster reliability.

To alleviate the mismatch between the inter-cluster reliability and point-point distortions, Martins et al. [37] defined distortion types relevant to the cluster-point relationship: Missing Members and False Members with regard to a group of data points. For a group $\Gamma_X \subset X$ of similar points (e.g., within the same category of a dataset or clustered by a clustering algorithm) in the original space, $\Gamma_Y \subset Y$ is used to denote the "projected group" that corresponds to Γ_X. Here, False Members are the points in $\Gamma_Y \setminus \Gamma_X$, and Missing Members are in $\Gamma_X \setminus \Gamma_Y$ (Fig. 1b); however, the literature review also revealed that the generalization was insufficient to reflect the degree to which users can perform inter-cluster tasks precisely. We further generalize the distortion types by proposing new inter-cluster distortion types that directly harm inter-cluster reliability.

2.1.2 Distortion Metrics

According to a survey conducted by Nonato and Aupetit [49], most distortion metrics aim to measure point-point distortion. Among them, a few metrics evaluate how much Missing and False Neighbors distortion has occurred. For instance, Trustworthiness and Continuity (T&C) [67] locally measure how Missing and False Neighbors distort the ranks of each point's neighbors. Mean Relative Rank Errors (MRREs) [29] are similar to T&C; however, they consider not only the rank variance of the Missing and False Neighbors but also of True Neighbors—the points that are judged as neighbors in both spaces. Local Continuity Meta-Criteria (LCMC) [8] is another variant of T&C; it only considers True Neighbors. Still, measuring point-point distortion cannot adequately measure inter-cluster reliability, since it needs to quantify the relationship between clusters.

Motta et al. [44] proposed graph-based group validation, which is the only metric measuring cluster-point distortion we could find as a relevant work. The metric first extracts clusters from both the original and projected spaces using graph-based clustering. The metric then calculates each cluster's structural persistence in the opposite space by measuring how much Missing and False Members distorted the cluster. Given that the metric examines each cluster independently, it is inappropriate to use it to measure inter-cluster reliability, which refers to multiple clusters at once.

Measuring the distortion of predefined clusters with a clustering quality metric has been widely used to evaluate MDP. For example, Joia et al. [26] and Fadel et al. [20] used the silhouette coefficient [57] to quantify cluster preservation in MDP. However, one limitation is that

the inter-cluster structures of real-world datasets are usually unknown. Graph-based group validation also suffers from the same problem, as it performs clustering once for all data and uses it as predefined clusters. By contrast, our metrics consider the complex inter-cluster structure by examining repeatedly extracted random clusters, thus much accurately quantify inter-cluster reliability.

2.1.3 Distortion Visualizations

To overcome the inherent limitation of metrics that describe only the overall distortions with one or two representative numerical values, complementary visualizations are proposed [49]. The visualizations aim to reveal the submerged distortion information summarized by a single or two values, thus helping users identify trustworthy areas of the projection or detect distortion patterns.

Distortion visualizations commonly highlight regions with local point-point distortions by decomposing the area into grids, where each grid cell corresponds to the data point and encodes the corresponding point's distortion to the cells. The decomposition is usually done using a heatmap [61], Voronoi diagram [2, 24, 31], or 2D point-cloud [37, 38]. By contrast, MING [10] explains False and Missing Neighbors by visualizing the shared amount of the nearest neighbor graphs constructed in the original and projected space. In this work, we quantified pointwise distortions by aggregating the inter-cluster distortion of the clusters and visualized them.

2.2 Inter-Cluster Reliability

As many MDP techniques intentionally focus on local neighbors, they have trouble reflecting the original high-dimensional space's global inter-cluster structure. For example, Barnes-Hut t-SNE [66] and LargeVis [63] concentrate on local structures by interpreting data based on k-Nearest Neighbor (kNN) graphs. Using a kNN with a small k ($k \ll N$) also allows them to reduce computation. However, as kNN graphs with small k only maintains the relations between the point and its local neighbors, they can only reflect limited local structures [21].

Recently proposed MDP techniques have tried to preserve both the local and global inter-cluster structures. For example, Narayan et al. [47] introduced den-SNE and densMAP, which modify t-SNE and UMAP, respectively, to better preserve clusters' density. Another common strategy is to first construct a global skeletal layout using representative points (i.e., landmarks) and formulate local structure around each landmark [20, 21, 26, 52, 55]. However, even for these approaches, completely retaining the original space's inter-cluster structure during the projection is inherently impossible. Therefore, it is vital to measure the extent to which these techniques preserve inter-cluster reliability for a proper evaluation and analysis.

Previous studies have attempted to explain the inter-cluster reliability of MDP through visualizations. For instance, the Compadre system [12] enables an inter-cluster structure analysis based on matrix visualization, and ClustVis [42] does so with a heatmap. Visual analytics systems [7, 35] with similar goals have also been proposed. Unlike these previous works, which utilized separate visual idioms to show the distortion, we adopted a strategy of visualizing the distortion within the projection [31, 37] to explain inter-cluster reliability. Therefore, users can directly identify where and how the inter-cluster distortions occurred in the projection.

3 DESIGN CONSIDERATIONS FOR STEADINESS AND COHESIVENESS

In this section, we first survey inter-cluster tasks, which are essential in data analysis using MDP [37, 59], through a literature review. We then establish the design considerations based on the survey that our metrics (Steadiness and Cohesiveness) should satisfy to adequately measure how much inter-cluster tasks can be held precisely in MDP.

3.1 Inter-Cluster Tasks Analysis

To identify the importance of inter-cluster structure preservation and to elicit the design considerations for our metrics, we inspected previous papers that addressed tasks related to clusters. We first investigated 31 papers introduced in a systematic review conducted by Sacha et al. [58],

---

[1]This paper denotes both linear and nonlinear embedding of multidimensional data as MDP, following previous research [18, 19, 49].
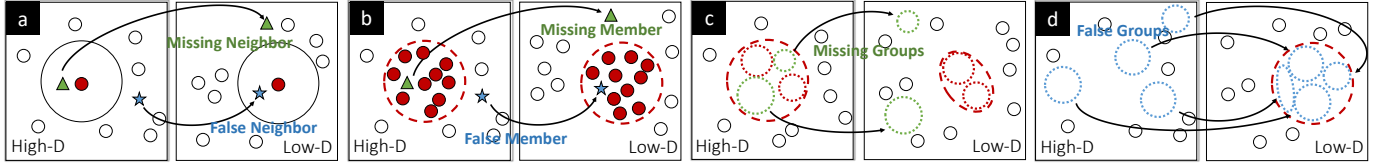
Fig. 1. Illustration of the concepts of a) Missing Neighbors and False Neighbors, b) a group and its Missing / False Members, c) Missing Groups, and d) False Groups. Data points are represented as black-bordered circles, triangles, and stars. Groups (clusters) are depicted as the objects with dashed or dotted outlines.

which surveyed how analysts interact with MDP. We also investigated 155 articles citing Sacha et al. using Google Scholar to expand the search space. As a result, we identified 26 papers concerning inter-cluster tasks: tasks that investigate the inter-cluster structure of original data through its 2D projections. Regarding the task taxonomy for MDP proposed by Etemadpour et al. [18, 19], we classified the tasks into three categories in terms of inter-cluster distortions. We then organized them into individual tasks, as listed in the following:

T1: **Identify separate clusters in the original space by exploring clusters in the projected space**. Recognize the separation between clusters [9, 14, 69] or distinguish a cluster from the others [46, 56].

T2: **Seek the relationships between clusters of the original space based on those of the projected space**. (**1**) Investigate the hierarchical or inclusion relation between clusters (i.e., check whether clusters can again be divided into smaller parts with higher density, which we call "subclusters") [34, 72]. (**2**) Estimate the clusters' similarities based on their distances in the projected space [46,71].

T3: **Compare clusters in the original space based on their features in the projected space**. Estimate and compare the clusters' original sizes or densities based on their sizes or densities in the projected space [1, 7].

The tasks were verified through semi-structured interviews with four machine learning (ML) engineers (E1-E4) with more than three years of experience. Three engineers confirmed that they practically perform the tasks for the real-world problem. Only E1 said that he does not perform the tasks. This is because he usually works with data with well-distributed vector representations processed by a deep neural network, where no inter-cluster structure exists.

Previous surveys of high-dimensional data analysis tasks based on MDPs further confirm our task analysis results, as those works show similar results to ours, despite using different methodologies. T1 is covered by Brehmer et al.'s task taxonomy based on interviews with 10 data analysts [5], and T2 and T3 are covered by the taxonomy of cluster separations in MDPs discussed by Sedlmair et al. [60].

Our survey indicated that point-point and cluster-point distortion metrics cannot correctly quantify how well inter-cluster tasks can be performed. Point-point distortion metrics focus on each point's neighborhood instead of the inter-cluster structure. Therefore, the metrics can only measure the potential accuracy of relation-seeking tasks relevant to point-point relations, such as finding kNN of the given point [18]; they cannot measure the extent to which inter-cluster tasks can be performed accurately as those tasks focus on the cluster level.

Cluster-point distortion metrics can estimate the potential accuracy of T3, as the size and density of each cluster are related to the cluster itself. More precisely, if an MDP generates outliers for a cluster, the cluster's size is reduced (if the density is maintained), or its density is reduced (if the size is maintained). Both distortions directly affect the comparison task. By contrast, cluster-point distortion metrics still fails for T1 and T2. As the metrics consider each cluster independently, they can only work for cluster identification tasks related to a single cluster (e.g., distinguishing the outliers of the cluster [18]) or relation-seeking tasks about a single cluster (e.g., finding k closest points of the given cluster [18]); however, they cannot provide sufficient information required to support T1 and T2 that consider multiple clusters at once.

3.2 Design Considerations

Based on the task analysis, we formulated three design considerations (C1, C2, C3) that Steadiness and Cohesiveness should satisfy to ad-

equately quantify how accurately the three inter-cluster tasks can be performed and thus able to precisely measure inter-cluster reliability.

C1: **Capture the inter-cluster structure in detail**. The inter-cluster structure in MDP is complex and intertwined [72], and often has no ground truth. Furthermore, each cluster's characteristics (e.g., shape, density, or size) vary widely [23]. Therefore, to quantify how precisely users can identify clusters (T1) or seek relationships between them (T2), we should thoroughly consider the inter-cluster structure in detail.

C2: **Consider stretching and compression individually**. The distances between clusters may be affected by two aspects of geometric distortions: stretching and compression [2]. If stretching occurs, users can misunderstand nearby clusters as distinct clusters. The opposite can happen if compression occurs (i.e., nearby groups can be identified as a single cluster). Furthermore, clusters' size and density can be overestimated due to stretching or can be underestimated by compression. As the two aspects of distortion result in different types of misperceptions about the clusters' size and density (T3) or their distance (T2-2), we should consider both aspects individually.

C3: **Measure how accurately the clusters identified in the projection reflect their original density and size**. Users can have misconceptions when comparing clusters (T3) if the projected clusters' size and density do not reflect those in the original space. To correctly quantify how much such misunderstandings can happen, we need to measure how accurately the clusters in the projection reflect their original density and size.

4 STEADINESS AND COHESIVENESS

We propose Steadiness and Cohesiveness to measure inter-cluster reliability by evaluating inter-cluster distortion, satisfying our four design considerations. Steadiness measures inter-cluster reliability in the projected space (e.g., separated clusters in the original high-dimensional space are still separated in the projected space), while Cohesiveness does the same for the original space (e.g., each cluster in the original space is not dispersed in the projected space).

4.1 Defining Inter-Cluster Distortion Types

To design Steadiness and Cohesiveness, we first defined two inter-cluster distortions types—False Groups and Missing Groups—by generalizing False and Missing Neighbors to the cluster level. False Groups distortion denotes the cases in which a low-dimensional group in a single cluster (red dashed circle in Fig. 1d) consists of separated groups in the original space (blue dotted circles in Fig. 1d), and Missing Groups distortion occurs when the original group (red dashed circle in Fig. 1c) misses its subgroups (green dotted circles in Fig. 1c) and therefore is divided into multiple separated subgroups in the projected space. Steadiness and Cohesiveness evaluate how well projections avoid False and Missing Groups, respectively (C2).

4.2 Computing Steadiness and Cohesiveness

We compute inter-cluster reliability through the following procedure: (Step 1) Constructing dissimilarity matrices. (Step 2) Iteratively computing partial distortions. (Step 3) Aggregating partial distortions into Steadiness and Cohesiveness. Based on the definitions of the two measures, Steadiness increases as clusters extracted from the projected space stay closer consistently together in the original space. In contrast, Cohesiveness increases when clusters in the original space are maintained more consistently in the projected space.

Each step is designed to satisfy all the design considerations (Sect. 3.2). First, we split the workflow to handle Steadiness and Cohesiveness independently after step 1 (C2). Step 2 exploits randomness to cover the complex inter-cluster structures (C1) and inherently quantifies how well the original density and size of clusters are retained (C3).

The workflow requires four functions as hyperparameters:

- Distance function for points, dist
  - Input: two points p and q
  - Output: the distance (or dissimilarity) between p and q
- Distance function for clusters, dist_cluster
  - Input: two clusters A and B
  - Output: the distance (or dissimilarity) between A and B
- Cluster extraction function, extract_cluster
  - Input: a seed point p
  - Output: a cluster in the projected space (for Steadiness) or the original space (for Cohesiveness) centered on p
- Clustering function, clustering
  - Input: a set of points P
  - Output: a clustering result of the input points P where the clustering takes place in the original space (for Steadiness) or the projected space (for Cohesiveness)

Two distance functions are used to compute the amount of inconsistency, while the other two functions are used for the iterative computation of partial distortions. These functions are explained in detail in Sect. 4.3.

4.2.1 Step 1: Constructing Dissimilarity Matrices

We begin the measurement by constructing dissimilarity matrices D^+ and D^-. We first construct distance matrices H and L satisfying $H_{ij} = \text{dist}(h_i, h_j)$ and $L_{ij} = \text{dist}(l_i, l_j)$, where h_i and l_i denote the original and projected coordinates of input data point p_i, respectively. For dist, we used Shared-Nearest Neighbor (SNN) based dissimilarity [16] as a default (Sect. 4.3). H and L are then normalized by dividing all elements by their max elements H_{max} and L_{max}. Raw dissimilarity matrix D is obtained by subtracting L from H. The positive elements in D denote that the distance between the corresponding points pair is compressed, and the opposite denotes that the distance is stretched. We then construct D^+ and D^-, where $D_{ij}^+ = (D_{ij}$ if $D_{ij} > 0$, else 0) and $D_{ij}^- = (-D_{ij}$ if $D_{ij} < 0$, else 0).

4.2.2 Step 2: Iteratively Computing Partial Distortions

The next step is to iteratively compute partial distortions by randomly extracting clusters from one space and evaluating their dispersion in the opposite space. In this section, we describe how to compute partial distortions in a single iteration.

Extracting random clusters   For each iteration, we first select a seed point randomly in the projected space (Steadiness) or the original space (Cohesiveness). Then, the extract_cluster function takes the random seed point as input and extracts a cluster centered on the point as output. The random selection of the seed point leads to the extraction of clusters from diverse locations and therefore it is possible to cover the entire inter-cluster structure after sufficient iterations (e.g., 200 iterations for the data consists of 10,000 points) (C1). By default, we use the SNN similarity (Sect. 4.3) for the extract_cluster function to gather points near the seed point.

Revealing the cluster's dispersion in the opposite space   Next, we reveal how the randomly extracted cluster is dispersed in the opposite space. To do this, the clustering function takes the points of the extracted cluster generated by extract_cluster as input. Afterward, the clustering function clusters the input points in the opposite space and returns the set of separated clusters $\mathscr{C} = \{C_1, C_2, \ldots, C_n\}$ as output. Hierarchical DBSCAN (HDBSCAN) [6, 40] utilizing an SNN-based distance function is used as the default clustering function (Sect. 4.3).

This step also allows the metrics to measure how well the clusters reflect their original density and size (C3). If a cluster's original outliers are merged into a single cluster during MDP (False Groups distortion), either the cluster's size or density will be increased. This situation can be captured while checking the projected cluster's dispersion in the original space. For the opposite case (Cohesiveness), if an original space's cluster loses some of its points during MDP, either its size or density in the projected space will be reduced. Revealing Missing Groups distortion captures this issue (Sect. 7.1).

Computing distortions between dispersed groups   In this step, we take $\mathscr{C}$ as input and generate distortion m_{ij} and its weight w_{ij} for each pair of clusters (C_i, C_j), based on point-stretching and point-compression metrics proposed by Michaël Aupetit [2]. We generalized the point-stretching and point-compression to the cluster-stretching $m_{ij}^{stretch}$ (Steadiness) and cluster-compression $m_{ij}^{compress}$ (Cohesiveness) by substituting the distance between points to the distance between clusters. For each cluster pair (C_i, C_j), we compute their distance δ_h and δ_l in the projected space and the original space, respectively, utilizing dist_cluster. The default dist_cluster is designed by expanding the SNN-based distance function for points (Sect. 4.3). Then, we check whether the distance is compressed or stretched and consecutively compute the distortion as follows:

$$m_{ij}^{\text{stretch}} = \frac{\mu_{C_i,C_j}^{stretch} - \min D^-}{\max D^- - \min D^-}, \; m_{ij}^{\text{compress}} = \frac{\mu_{C_i,C_j}^{compress} - \min D^+}{\max D^+ - \min D^+},$$

where

$$\mu_{C_i,C_j}^{stretch} = -(\delta_h - \delta_l) \text{ if } -(\delta_h - \delta_l) > 0, \text{ otherwise } 0,$$

$$\mu_{C_i,C_j}^{compress} = \delta_h - \delta_l \text{ if } \delta_h - \delta_l > 0, \text{ otherwise } 0.$$

The weight w_{ij} of a pair (C_i, C_j) is determined as $|C_i| \cdot |C_j|$. The weights penalize the distortion of larger clusters more than smaller ones; thus, we can deal with the inter-cluster structure consisting of the clusters of various sizes (C1).

4.2.3 Step 3: Aggregating Partial Distortions

This step aggregates the iteratively computed partial distortions to Steadiness and Cohesiveness. The iterative partial distortion measurement generates a set of distortions and their corresponding weights. Let's denote the set as follows:

- $(m_1^{compress}, w_1), \cdots, (m_{n_{St}}^{compress}, w_{n_{St}})$ where n_{St} denotes the number of total cluster pairs generated throughout the entire partial distortion measurement of Steadiness.

- $(m_1^{stretch}, w_1), \cdots, (m_{n_{Co}}^{stretch}, w_{n_{Co}})$ where n_{Co} denotes the number of total cluster pairs generated throughout the entire partial distortion measurement of Cohesiveness.

We then calculate the final scores as follows:

$$\text{STEADINESS} = 1 - \sum_{i=1}^{n_{St}} w_i \cdot m_i^{compress} / \sum_{i=1}^{n_{St}} w_i.$$

$$\text{COHESIVENESS} = 1 - \sum_{i=1}^{n_{Co}} w_i \cdot m_i^{stretch} / \sum_{i=1}^{n_{Co}} w_i.$$

The final scores lie in the range of [0, 1]. The weighted average is subtracted from 1 to assign lower scores to lower-quality projections.

4.3 Designing Hyperparameter Functions

4.3.1 Parameterizing Hyperparameter Functions

The workflow of computing Steadiness and Cohesiveness requires four hyperparameter functions: dist, dist_cluster, clustering, and extract_cluster. We parameterized these functions because both the definition of distance and the definition of clusters vary depending on the analysis goals. There are various ways to define the distance between two data points (e.g., Euclidean distance, geodesic distance, cosine similarity). The definition of clusters also varies, and thus many different clustering algorithms (e.g., K-Means [13], Density-based clustering [17], Mean shift [11]) exist. Therefore, it is unreasonable to use a fixed definition for both. This is in line with the fact that, as there are various ways to define similarity between each point and its local neighbors, there are diverse local metrics that utilize different similarity definitions. However, parameterization could reduce metrics' interpretability. Thus, we designed the default hyperparameter functions that align with our design considerations to allow users to easily understand and use our metrics.

4.3.2 Default Hyperparameter Functions

To design the default functions, we first set the definitions of distance and cluster. We defined distance as the dissimilarity of the points based on the Shared-Nearest Neighbors (SNN) [16] similarity, which assigns a high similarity to point pairs sharing more kNNs. SNN-based dissimilarity was selected because Steadiness and Cohesiveness should reflect the inter-cluster structure of the original high-dimensional space. Although it is common to use a kNN graph to reflect a high-dimensional space [63, 66], kNN's ability to describe the structure of data decreases as dimensionality grows [3, 25]. SNN-based dissimilarity tackles this issue as the similarity of two points is robustly confirmed by their shared neighbors, thus better representing the structure of high-dimensional spaces compared to kNN [15, 33]. We also defined a cluster as the contiguous data region, or manifold with an arbitrary shape, where the density of the region is higher than its surroundings. This definition is followed by density-based clustering algorithms. We used this definition because the metrics should capture the complex and intertwined inter-cluster structure consisting of clusters of various sizes and shapes (C1), and therefore should be able to define clusters more flexibly. We designed the default hyperparameter procedures to satisfy both the definitions and the original design considerations (Sect. 3.2).

Distance function for points, dist As mentioned, the distance function is based on SNN similarity. Let us first denote k-nearest neighbors of a point p as $p_1, p_2, \cdots, p_k$, in order. The SNN similarity between two points p, q is defined as $sim(p,q) = \sum_{(m,n) \in N_{p,q}} (k + 1 - m) \cdot (k + 1 - n)$ where $N_{p,q}$ is a set of each pair (m,n) satisfying $p_m = q_n$. $sim(p,q)$ increases when more k-nearest neighbors with high ranks overlap. We consecutively normalized all SNN similarity values by dividing them by the max SNN similarity max_sim of the dataset. Finally, we defined distance function dist as $dist(p,q) = 1/(sim(p,q) + \alpha)$. We used reciprocal transformation [62] to further penalize low similarity, where α controls the amount of penalization. $\alpha = 0.1$ is used as the default.

Distance function for clusters, dist_cluster As for dist_cluster, we first defined the similarity between clusters and converted it to their distance. We used average linkage [45], as it is robust to outliers compared to competitors (e.g., simple linkage), thus defining the similarity of two clusters A and B as $sim(A,B) = \sum_{p_A \in A} \sum_{p_B \in B} sim(p_A, p_B)/|A| \cdot |B|$, where p_A, p_B denotes the points in A and B. We then defined the distance between A and B as $dist\_cluster(A,B) = 1/(sim(A,B) + \alpha)$.

Clustering function, clustering As our definition of cluster is the one used in conventional density-based clustering, designing clustering required a single decision: selecting the proper density-based clustering algorithm. We selected HDBSCAN, which is a state-of-the-art density-based clustering algorithms. As HDBSCAN can handle clusters with various shapes and densities and is robust to noises (outliers) [39], exploiting it helps to reveal the dispersion of clusters regardless of the clusters' characteristics (e.g., shape, size, or density). Therefore, it helps the metrics deal with complex inter-cluster structures (C1). HDBSCAN also tackles the curse of dimensionality [68], which suits our metrics that need to consider the higher dimensional space. To align clustering with our dissimilarity definition, our HDBSCAN utilized dist for the distance calculation.

Cluster extraction function, extract_cluster The design of extract_cluster mainly follows a density-based clustering process, aligned to clustering; it uses random seed point as a sole core point and assigns nearby points, which are treated as non-core points, successively to form a cluster. In detail, the function traverses seed point p's k-nearest neighbors and includes each neighbor point p_i as a cluster member with a probability of $sim(p, p_i)/max\_sim$. When the neighbor point is determined as a cluster member, it goes into a queue so that its neighbor can also be traversed later. Adding neighbors stochastically makes extracted clusters not span the entire kNN graph but form a dense structure.

To diversify the size of the extracted clusters, we limited the traversal number starting from the seed point and allowed repeated visits. Combined with the random starting seed point, this strategy enriches the range that our metrics cover, thus helping the metrics deal with complex

inter-cluster structures (C1). The strategy fundamentally relies on the fact that randomness can help analyze a complex, uncertain system [64]. We fixed the number of traversal to 40% of the total number of data points for our evaluations (Sect. 6, 7).

4.4 Visualizing Steadiness and Cohesiveness

To overcome the limitation of metrics in that they describe the overall distortion in a single or two numeric values, we developed a complementary visualization: a reliability map (Fig. 4, 5, 6). The reliability map reveals how and where inter-cluster distortion occurred by showing Steadiness and Cohesiveness at each point. The distortions at each point are quantified by aggregating partial distortion values computed throughout the measurement of our metrics. The map shows these pointwise distortions embedded within the projection.

The pointwise distortion is obtained by aggregating partial distortions computed throughout the iterative process (Sect. 4.2.2). Recall that the iterative computation results in a set of distortion $m_{ij}^{compress}$ or $m_{ij}^{stretch}$ and weight w_{ij} between a pair of clusters $\{C_i, C_j\}$. For all $\{C_i, C_j\}$, we register every $p \in C_j$ to every $q \in C_i$ with the distortion strength $m_{ij} \cdot w_{ij}$, and do the same in the opposite direction. Duplicated registration of a point are removed by averaging distortion strengths. We compute each point's approximated local distortion by summing up the registered distortion strengths.

The reliability map visualizes these pointwise distortions through edge-based distortion encoding. We constructed a kNN graph in the projection and made each edge (p, q) of the graph depict the sum of p and q's pointwise distortion. If the points within a narrow region have high distortion, the edges between the points will be intertwined in the region (e.g., red dotted contours in Fig. 5, 6); they will be recognized as clusters with distinguishable inter-cluster distortion. However, using a large k might generate visual clutter; we empirically found that k between 8 and 10 is an adequate choice for both expressing inter-cluster distortion and avoiding visual clutter. Martins et al.'s point cloud distortion visualization [37] is similar to ours, but it computes the distortion value at each pixel instead of encoding to edges.

To express False Groups and Missing Groups distortion types simultaneously, we used CheckViz's two-dimensional color scale [31] (lower right corner of Fig. 5). Following the color scheme of CheckViz, we assigned purple to the edges with False Groups distortion and green to the edges with Missing Groups distortion. Moreover, edges with no distortion are represented as white, while black edges indicate that both distortion types occurred together.

We also implemented a cluster selection interaction (e.g., lower right box in Fig. 4) to allow users to identify Missing Groups distortion more precisely. After users select a cluster $C = \{p_1, p_2, \cdots, p_{|C|}\}$ by making a lasso with a mouse interaction, the reliability map constructs $C' = \{A(p_1) \cup A(p_2) \cup \cdots \cup A(p_{|C|})\}$, where $A(p)$ denotes the set of registered points of a point p. Subsequently, the edges connected to the points in C' are highlighted in red. Each highlighted edge's opacity encodes the sum of distortion strength of its incident points toward C (i.e., how much distance between its incident points and C is stretched).

5 IMPLEMENTATION

Steadiness and Cohesiveness are written in Python with an interface for users or programmers to easily implement and use user-defined hyperparameter functions. This is to facilitate the development and verification of possible alternatives of Steadiness and Cohesiveness later. The partial distortion computation is parallelized with CUDA GPGPU [48] supported by Numba [27]. We implemented the reliability map in JavaScript using D3.js [4]. The source code of the metrics and the map is available at github.com/hj-n/steadiness-cohesiveness and github.com/hj-n/snc-reliability-map, respectively.

6 QUANTITATIVE EVALUATIONS AND DISCUSSIONS

We evaluated Steadiness and Cohesiveness in terms of quantifying inter-cluster reliability by comparing them with existing local distortion metrics. We verified that our metrics well capture inter-cluster reliability, while previous local metrics miss some cases even with the apparent
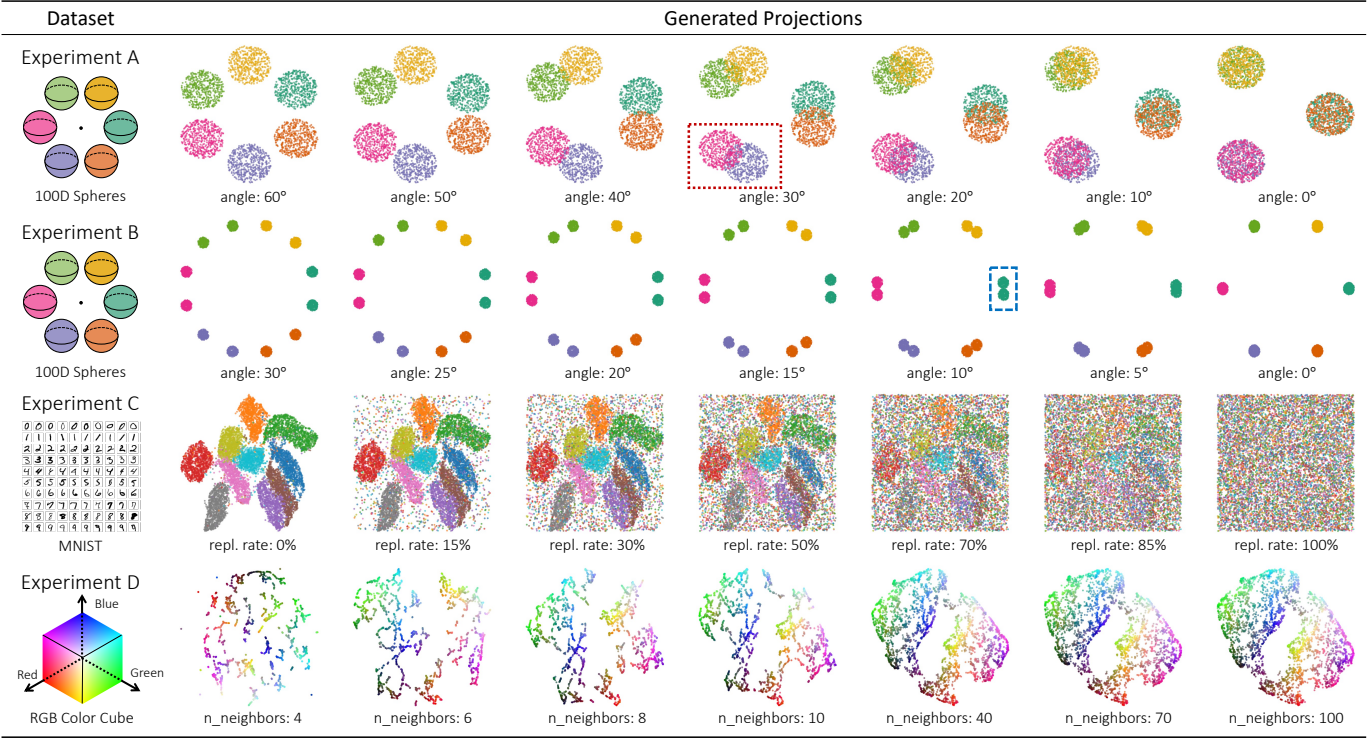
Fig. 2. The datasets and their projections used in Experiment A-D. A, B) The synthetic projection of the dataset with six 100-dimensional spheres, consists of six circles (A) and 12 circles (B) with an increasing amount of overlapping. C) MNIST dataset and their projections created by replacing a certain proportion of the *t*-SNE projection with random points with increasing replacement (repl.) ratios. D) The UMAP projection of the data randomly sampled from the RGB color cube dataset with an increasing `nearest neighbors` hyperparameter value.

distortions. The reliability map ascertained that our metrics accurately captured where and how the inter-cluster distortion occurred. Moreover, we evaluated our metrics' robustness by testing simpler hyperparameter functions (Sect. 4.3).

As baseline metrics, we chose T&C and MRREs (Sect. 2.1.2), the two representative local metrics that measure nearest neighbors preservation. We chose the two for comparison because 1) they were designed to measure Missing and False Neighbors, the point-wise version of Missing and False Groups and 2) nearest-neighbor preservation has been used as the core evaluation criteria for evaluating MDP techniques previously [21, 30, 43, 55, 66]. For MRREs, in this section we use "MRRE [Missing]" for the one that quantifies Missing Neighbors, and "MRRE [False]" for the other that quantifies False Neighbors.

6.1 Sensitivity Analysis

We conducted four experiments to check whether Steadiness and Cohesiveness can sensitively measure inter-cluster reliability. We designed the first two experiments (A, B) to evaluate our metrics' ability to quantify the inter-cluster distortion using the projections with synthetically generated False Groups (Experiment A) or Missing Groups (Experiment B) distortions respectively. The next two experiments (C, D) were conducted to investigate whether our metrics have the ability to properly assess the overall inter-cluster reliability difference of the projections.

6.1.1 Experimental Design

Experiment A: Identifying False Groups The goal of the first experiment was to evaluate whether and how Steadiness and previous local metrics (Continuity, MRRE [False]) identify False Groups. We first generated high-dimensional data consisting of six 100-dimensional spheres whose centers were equidistant from the origin. Each sphere consisted of 500 points. We then set the initial 2D projection of the dataset as six circles around the origin (the first projection on the first row of Fig. 2). Note that this projection is the most faithful view of the original data as we made each circle correspond to one high-dimensional sphere. To simulate False Groups distortion, we then distorted this ground-truth projec-

tion by overlapping the circles in pairs (the first row of Fig. 2). For each pair of circles (A, B) centered at C_A, C_B, respectively, we adjusted the degree of overlap by changing $\angle C_A O C_B$ from 60° to 0° with an interval of 2.5°, where O is the origin. For each projection, we measured Steadiness and Cohesiveness ($k = [80, 90, 100, 110, 120]$, 500 iterations), T&C ($k = [5, 10, 15, 20, 15]$), and MRREs ($k = [5, 10, 15, 20, 15]$). We used different k values and used the mean of their results as the final score for soundness.

Experiment B: Identifying Missing Groups To evaluate Cohesiveness and previous local metrics' (Trustworthiness, MRRE [Missing]) ability to measure Missing Groups distortion, we used the same high-dimensional dataset as Experiment A, but this time, we synthesized the initial projection consisting of 12 equally distant circles, where each consists of 250 points. We made a pair of nearby circles correspond to a single sphere in the original space (the second row of Fig. 2). We then overlapped each pair of circles (A, B) by adjusting $\angle C_A O C_B$ from 30° to 0° with an interval of 1.25° (the second row of Fig. 2). Note that unlike Experiment A, the initial projection is the least faithful projection but becomes more faithful as the circles in each pair overlap more. We used the same metrics setting as Experiment A.

Experiment C: Capturing quality degradation To test our metrics' ability to capture the quality degradation of the projection, we computed our and previous metrics for the projections with different levels of quality degradation. We first created a 2D *t*-SNE projection of the MNIST dataset [28] (the first projection on the third row of Fig. 2). We then replaced a certain proportion of the projected points with random points. We varied the replacement rate from 0 to 100% with an interval of 5% (the third row of Fig. 2). The inter-cluster reliability of the projections certainly degrade as the replacement rate increases. We checked whether the metrics can capture such quality degradation. We used the same metrics setting as Experiment A.

Experiment D: Identifying the effect of projection hyperparameters The final experiment was conducted to evaluate the capability of our metrics to capture the inter-cluster reliability differences caused by the hyperparameter choices of an MDP technique. This experiment was
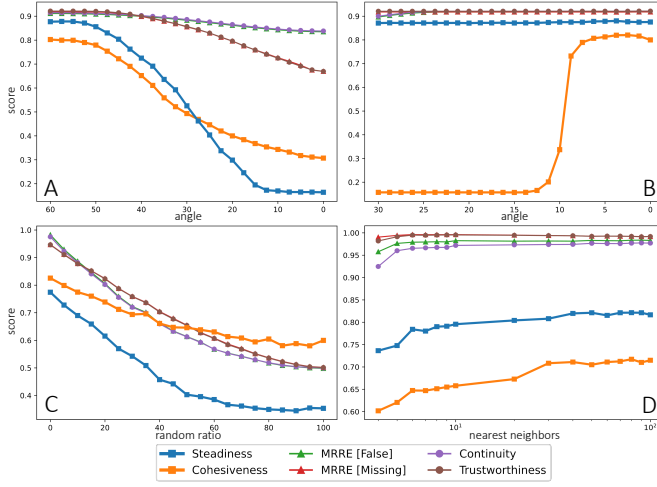
Fig. 3. The result of quantitative experiments A-D; the scores measured by our metrics (Steadiness and Cohesiveness) and baseline local distortion metrics (MRREs, T&C).

inspired by an analysis from the UMAP paper [41] where the authors assessed the impact of a hyperparameter, the number of `nearest neighbors` n, on the projection quality. Lower n values drive UMAP to more local structures, while higher values make the projection to preserve the global structures rather than the local details. In the original analysis, the authors qualitatively analyzed how n affects the UMAP projection of a randomly sampled 3-dimensional RGB cube data. The authors concluded that since randomly sampled data have no manifold structure, larger n values generate more appropriate projections than lower n values. Lower n values instead treat the noises from random sampling as fine-scale local manifold structures, generating an unreliable interpretation of the structure [41].

We tested whether our and previous metrics can quantitatively reproduce this conclusion. We first constructed a dataset of 4000 points randomly sampled from a 3-dimensional RGB cube. UMAP projections of the dataset with different n ($4-9$ with an interval of 1, $10-90$ with an interval of 10) were then generated (the fourth row of Fig. 2) and tested with the same metrics setting as Experiment A. We set another hyperparameter of UMAP, `min_dist`, to 0 because higher `min_dist` values tune projections to lose the local structure, reducing the difference between the projections generated with higher and lower n values. Setting it at 0.0 prevents such an effect from affecting the experiment.

6.1.2 Results

Experiment A As we decreased the angle between each circle pair (i.e., increasing the amount of false overlap), both Steadiness (slope $= 3.05 \cdot 10^{-2}$, $p < .001$) and Cohesiveness (slope $= 1.98 \cdot 10^{-2}$, $p < .001$) decreased. The baseline local metrics: Trustworthiness (slope $= 9.31 \cdot 10^{-3}$, $p < .001$), Continuity (slope $= 3.02 \cdot 10^{-3}$, $p < .001$), MRRE [Missing] (slope $= 9.13 \cdot 10^{-3}$, $p < .001$), and MRRE [False] (slope $= 3.01 \cdot 10^{-3}$, $p < .001$), also decreased, but the slope was statistically gentle compared the our metrics ($p < .001$ for all). (Fig. 3A)

Experiment B As we decreased the angles between each circle pair (i.e., making projections more faithful), Cohesiveness drastically increased around $10°$ (slope $= -0.169$ in range $[15, 5]$, $p < .001$), which is the point where the circle pair starts to overlap. Other measures such as Steadiness (slope $= -2,29 \cdot 10^{-4}$, $p < .001$), Trustworthiness (slope $= -1.81 \cdot 10^{-5}$, $p < .001$), Continuity (slope $= -7.17 \cdot 10^{-4}$, $p < .001$), MRRE [Missing] (slope $= -3.86 \cdot 10^{-6}$, $p < .001$) and MRRE [False] (slope $= -7.22 \cdot 10^{-4}$, $p < .001$) did not changed significantly (Fig. 3B).

Experiment C As the replacement rate increased, Steadiness (slope $= -4.29 \cdot 10^{-3}$, $p < .001$), Cohesiveness (slope $= -2.35 \cdot 10^{-3}$, $p < .001$), Trustworthiness (slope $= -4.60 \cdot 10^{-3}$, $p < .001$), Continuity (slope $= -4.70 \cdot 10^{-4}$, $p < .001$), MRRE [Missing] (slope $= -4.60 \cdot 10^{-3}$, $p < .001$), and MRRE [False] (slope $= -4.78 \cdot 10^{-4}$,
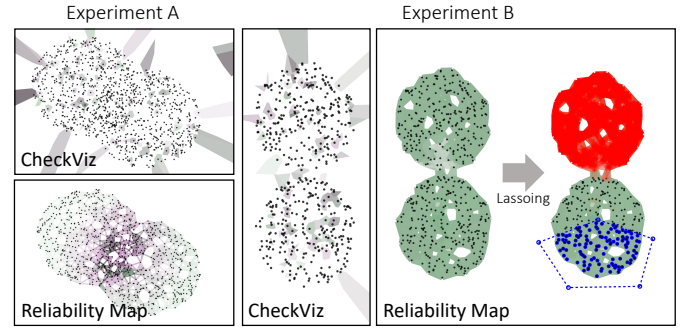

Fig. 4. The reliability map and CheckViz visualizing the distortion of the projections from Experiment A (red dotted square in Fig. 2) and Experiment B (blue dashed square in Fig. 2)). Unlike CheckViz, where no interesting pattern was shown, the reliability map demonstrated where and how False Groups distortion occurred (Exp. A). Even Missing Groups distortions was identified by the cluster selection interaction (Exp. B).

$p < .001$) all decreased. (Fig. 3C)

Experiment D As we increased n, both Steadiness (slope $= 5.78 \cdot 10^{-4}$, $p < .001$) and Cohesiveness (slope $= 9.44 \cdot 10^{-4}$, $p < .001$) increased, while Trustworthiness (slope $= -8.17 * 10^{-6}$, $p < .001$), MRRE [Missing] (slope $= -3.87 \cdot 10^{-5}$, $p < .001$) decreased. Continuity (slope $= 2.13 \cdot 10^{-4}$, $p < .001$) and MRRE [False] (slope $= 8.83 \cdot 10^{-5}$, $p < .001$) increased, though the slopes were statistically gentle compared to Steadiness ($p < .001$ for both). All baseline local metrics early saturated near the max score around $n = 5$. (Fig. 3D).

6.1.3 Discussion

The result of Experiment A suggests that our metrics could identify a loss of the inter-cluster reliability caused by False Groups distortion, as Steadiness decreased when the overlap of circle pairs increased. Cohesiveness also decreased, which means that not only False but also Missing Groups distortions had occurred. This is because for point p and q in a circle, although their Euclidean distance is maintained while the circle is overlapping with another circle, the SNN similarity decreases as more points intervene between p and q. Continuity and MRRE [False] also captured the decrease in the inter-cluster reliability due to False Groups distortion, but slower compared to our metrics.

For Experiment B, the result confirms that Cohesiveness correctly identifies Missing Groups distortion as the metric increased following the increasing overlap of circles that reduces Missing Groups distortion. Moreover, in Experiment B, the amount of Missing Groups distortion was captured only by Cohesiveness, which showed that our metrics have the ability to pinpoint the particular inter-cluster distortion type. In contrast, both Trustworthiness and MRRE [Missing] failed to capture this apparent Missing Groups distortion.

The Reliability map further confirms the results of Experiment A and B as it showed that Steadiness and Cohesiveness accurately identified the place where False Groups and Missing Groups occurred (Fig. 4). Reliability map located the False Groups distortion of Experiment A by highlighting the overlapped area in purple. For Experiment B, it was able to identify the Missing Groups relationship of two separated circles in a pair through the cluster selection interaction, as selecting the portion of one circle showed that the other circle was actually close to it; this result matches the ground truth. In contrast, CheckViz, which visualized the False and Missing Neighbors distortion of each point computed by T&C, did not show any pattern.

In Experiments C and D, both Steadiness and Cohesiveness could capture the decrease (Experiment C) and the increase (Experiment D) in inter-cluster reliability. Moreover, Experiment D showed that our metrics also can be used to quantify the effect of a hyperparameter by reproducing the result of human observers' qualitative analysis [41]. In contrast, local metrics barely captured the certain increase of inter-cluster reliability in Experiment D.

Overall, the experiments proved that our metrics can properly measure inter-cluster reliability. On the contrary, local metrics failed for some cases even with the apparent inter-cluster distortion.
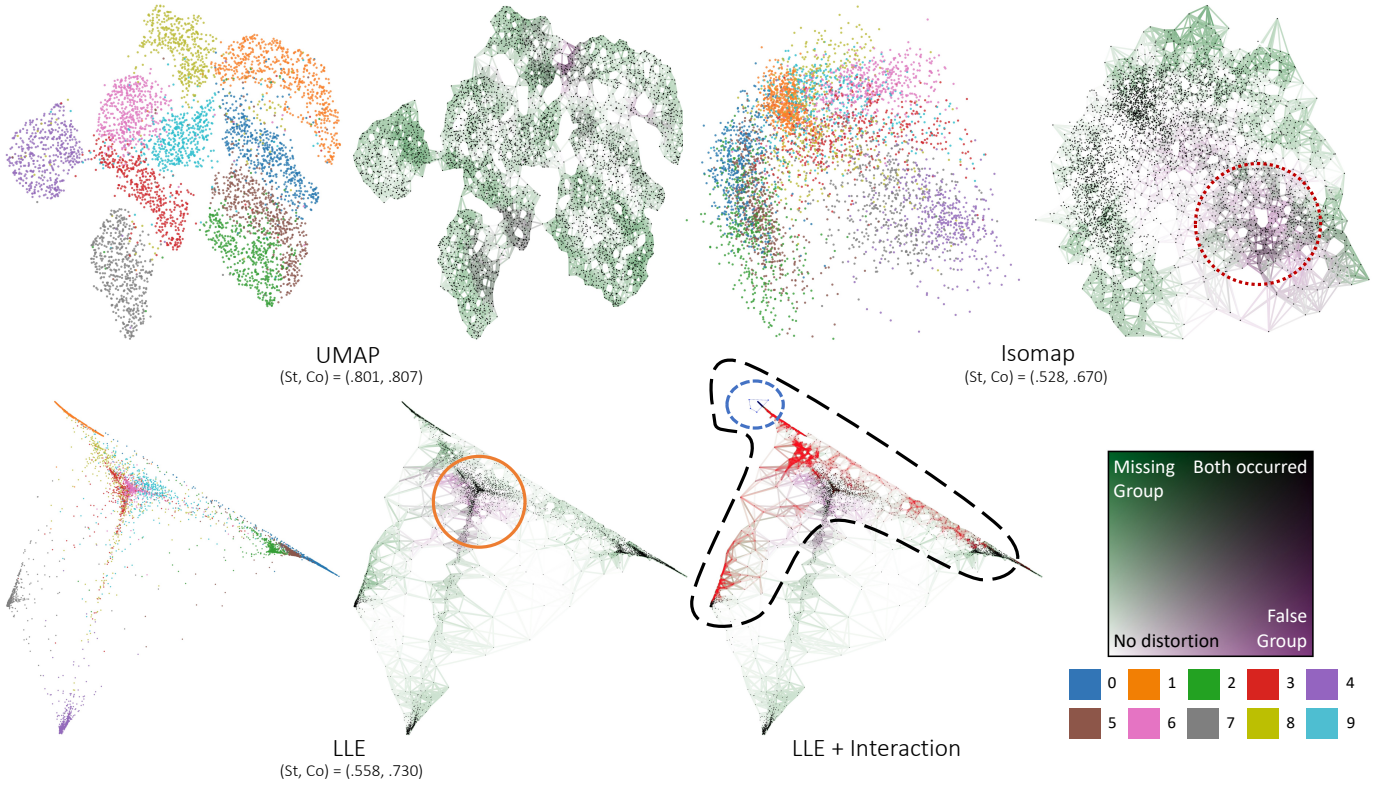
Fig. 5. The UMAP, Isomap, and LLE projections of MNIST test dataset and the reliability maps that visualize each projection's inter-cluster distortion. Overall Steadiness (St) and Cohesiveness (Co) scores are depicted under the name of each technique. For each MDP technique, the left pane shows the class identity, and the right pane shows the reliability map. Color-encoding of the inter-cluster distortion used in the reliability map is in the lower right corner. The projection and the corresponding reliability map of t-SNE and PCA are in Appendix B.

| clustering | Distance Measurement | | | |
| | SNN-based | | Euclidean | |
	St	Co	St	Co
HDBSCAN	$5.78 \cdot 10^{-4}$	$9.44 \cdot 10^{-4}$	$3.35 \cdot 10^{-4}$	$-1.37 \cdot 10^{-3}$
X-Means	$1.59 \cdot 10^{-4}$	$1.10 \cdot 10^{-3}$	$2.07 \cdot 10^{-4}$	$-4.98 \cdot 10^{-4}$
K-Means ($K = 5$)	$2.06 \cdot 10^{-4}$	$1.40 \cdot 10^{-3}$	$3.10 \cdot 10^{-4}$	$-7.01 \cdot 10^{-4}$
K-Means ($K = 10$)	$2.36 \cdot 10^{-4}$	$1.16 \cdot 10^{-3}$	$3.36 \cdot 10^{-4}$	$-7.29 \cdot 10^{-4}$
K-Means ($K = 20$)	$2.91 \cdot 10^{-4}$	$9.32 \cdot 10^{-4}$	$3.41 \cdot 10^{-4}$	$-6.12 \cdot 10^{-4}$

Table 1. The result of Experiment D conducted with diverse hyperparameter procedure settings (Sect. 6.2). Each cell depicts the slope of regression line which represents the relation between `nearest neighbor` value and the score of Steadiness (St) and Cohesiveness (Co). Every regression analysis result satisfied $p < .001$.

6.2 Robustness Analysis

We also investigated the robustness of Steadiness and Cohesiveness against hyperparameters by conducting Experiment D using Steadiness and Cohesiveness with different hyperparameters. We tested Steadiness and Cohesiveness with simpler hyperparameters functions as hyperparameter functions can considerably change the behavior of our metrics. For the goal, we tested simpler clustering algorithms, X-Means [54] and K-Means [13] (number of clusters $= 5, 10, 20$), instead of the default HDBSCAN algorithm for `clustering`. We also tested the Euclidean distance as `dist` instead of the default SNN-based distance. While using Euclidean distance for the distance measurement between points, we also defined the distance between two clusters `dist_cluster` as the Euclidean distance between their centroids instead of the default definition based on SNN similarity to align with `dist`. For `extract_cluster`, we treated every traversed points as cluster members instead of using probability to weight the points with high SNN similarity.

As a result (Table 1), Steadiness and Cohesiveness with simpler clustering hyperparameter functions both increased as `nearest neighbors`

values increased, which confirms the ability to properly quantify inter-cluster reliability. This result shows that our metrics' capability is not bound mainly by the selection of `clustering` but instead originates more from the power of randomness to analyze complex structures [64]. What is interesting here is that the case of K-Means ($K = 20$) showed the most similar results to the case of the default HDBSCAN hyperparameter both for Steadiness and Cohesiveness. This is because when clustering the extracted clusters in the opposite space, the inter-cluster structure composed of arbitrary shapes and sizes can be better represented by the fine-grained K-Means clustering result than the coarse-grained result.

However, Cohesiveness failed for all cases that used Euclidean distance as `dist`. This result shows that while designing hyperparameters functions for Steadiness and Cohesiveness, users should carefully consider the definition of the distance between two points.

7 CASE STUDIES

We report two case studies that we conducted with two ML engineers (E2, E3). During the study, we demonstrated to the engineers how Steadiness, Cohesiveness, and the reliability map works, and they explored with us the original inter-cluster structure of MNIST and Fashion-MNIST [73] test datasets, where both live in a 784-dimensional space and consist of 10 classes. The case study showed that our metrics and the reliability map supports users in 1) selecting adequate projection techniques or hyperparameter settings that match the dataset and 2) preventing users' misinterpretation that could potentially occur while conducting inter-cluster tasks (Sect. 3.1). ML engineers agreed that such support is helpful in interpreting the inter-cluster structure of high-dimensional data.

7.1 MNIST Exploration with Diverse MDP Techniques

To explore the inter-cluster structure of MNIST, we projected it with t-SNE, UMAP, PCA [53], Isomap [65], and LLE. We measured the Steadiness and Cohesiveness ($k = 100$, 500 iterations) of each projection and visualized the result with the reliability map (Fig. 5).
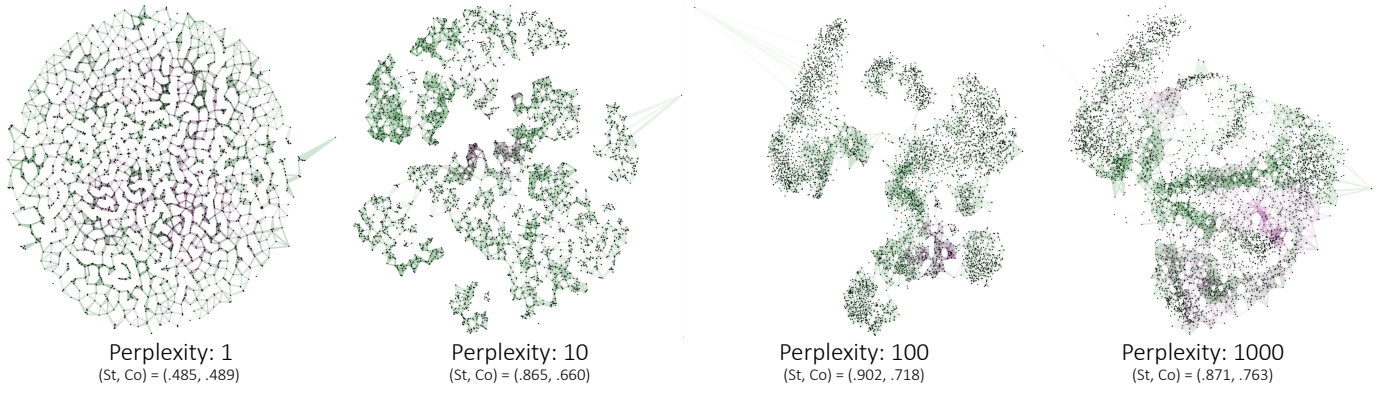
Fig. 6. The reliability maps that visualize the inter-cluster distortion of t-SNE projections ($\sigma = [1, 10, 100, 1,000]$) made for Fashion-MNIST test dataset. Steadiness and Cohesiveness scores are depicted under the the perplexity value of each projection.

We first discovered that visualizing Steadiness and Cohesiveness can prevent users from misidentifying a cluster separation of the original space (T1). For instance, in the Isomap projection, we found the region with high False Groups distortion consists of categories #4 and #7 (red dotted circle in Fig. 5). A similar region was also observed in the PCA projection (Appendix B). LLE also has the cluster with high False Groups distortion composed of categories #3, #6, #8, and #9. Without checking False Groups distortion, one could make the wrong interpretation that such a region belongs to same cluster; visualizing the distortion with the reliability map helped us avoid this misperception.

Visualizing False Groups distortion also allowed additional reasoning beyond a mere quantitative score comparison to choose the proper projection technique. We found that False Groups distortions that occurred in Isomap, PCA (overlap of category #4 and #7), and LLE (overlap of category #3, #6, #8, #9) did not occur in t-SNE or UMAP. This finding explains why the Steadiness of t-SNE and UMAP are higher than other projections, advocating the use of t-SNE and UMAP in exploring the inter-cluster structure of a MNIST dataset.

Still, as the t-SNE and UMAP projections also suffered from Missing Groups distortion, we critically interpreted that the clusters in these projections actually stay closer to each other than they look. E3 noted that this interpretation matches the ground truth that digits in MNIST stay much closer and mixed in the original space than their representations.

Moreover, we found that by using cluster selection interaction, users can accurately estimate and compare cluster sizes and shapes (T3). As we selected the local area in LLE (blue dashed ellipse in Fig. 5), the reliability map highlighted a much larger region around the selected region (black long-dashed contour in Fig. 5). This means that the original cluster containing the selected points was much larger than we can see in the projections and lost its portion as dispersed outliers (i.e., Missing Groups distortion occurred). We identified this problem through cluster selection interaction and escaped from the misinterpretation.

7.2 Fashion-MNIST Exploration with t-SNE

In the second case study, we explored Fashion-MNIST dataset using t-SNE projections with varying hyperparameters. We measured our metrics ($k = 100$, 500 iterations) on t-SNE projections generated with different perplexity values $\sigma \in [1, 10, 100, 1,000]$ and visualized the result with the reliability map (Fig. 6). Note that using a high value for σ makes the t-SNE focus more on preserving global structures [70].

As a result, we found that our metrics and the reliability map can help in selecting adequate hyperparameter settings. For example, the projection with $\sigma = 1$ had both False and Missing Groups distortion distributed uniformly across the entire projection space. This finding, which aligns with the low score of the projection, showed that low σ values are not sufficient to capture the global inter-cluster structure, which matches its actual behavior. This result justifies that it is proper to select a higher σ value to investigate Fashion-MNIST. The fact that $\sigma = 100$ and $\sigma = 1,000$ projections earned higher scores for both Steadiness and Cohesiveness compared to $\sigma = 1$ and $\sigma = 10$ projections strengthens this interpretation.

Thus, we subsequently analyzed the $\sigma = 100$ and $\sigma = 1,000$ projections and discovered that our metrics prevent users from making the wrong interpretation of the relations between clusters (T2). We first noticed that the $\sigma = 100$ projection has more compact clusters compared to the projection with $\sigma = 1,000$, where clusters are slightly more disperser and closer to each other. As the $\sigma = 100$ projection achieved a relatively high Steadiness score, we could conclude that each compact cluster also exists as a cluster also in the original space. However, as the $\sigma = 1,000$ projection increased Cohesiveness, we were not able to believe the separation of the clusters depicted in the $\sigma = 100$ projection (T2-1). According to Cohesiveness, the distances between the clusters in the original spaces is better depicted in the $\sigma = 1,000$ projection. Therefore, it is more reliable to interpret the original inter-cluster structure as a set of subclusters that constitute one large cluster rather than as a set of separated clusters (T2-2). E2 paid particular attention to this result. She pointed that it is common to perceive that projections with well-divided clusters (e.g., $\sigma = 100$ projection) better reflects the inter-cluster structure, but this result shows that such a common perception could lead to a misinterpretation of inter-cluster structure.

8 CONCLUSION AND FUTURE WORK

Although it is important to investigate the inter-cluster distortion in many MDP tasks, there were previously no metrics that directly measure such distortions. In this work, we first surveyed user tasks related to identifying the inter-cluster structures of MDP and elicited design considerations for the metrics to evaluate the inter-cluster reliability. Next, we presented Steadiness and Cohesiveness to evaluate inter-cluster reliability by measuring the False Groups and Missing Groups distortions and presented the reliability map to visualize the metrics. Through quantitative evaluations, we validated that our metrics adequately measure inter-cluster reliability. The qualitative case studies showed that our metrics can also help users select proper projection techniques or hyperparameter settings and perform inter-cluster tasks with fewer misperceptions, assisting them in interpreting the original space's inter-cluster structure.

As a future work, we plan to enhance the scalability of our algorithm. The algorithm currently computes the iterative partial distortion measurement sequentially. As each iteration works independently, we plan to accelerate the algorithm leveraging multiprocessing. We also plan to improve our metrics to consider the hierarchical aspect of inter-cluster structures and reduce the number of hyperparameters. Another interesting research direction would be to investigate how Steadiness, Cohesiveness, and their visualizations affect users' perception of the original data, which will provide an in-depth understanding of MDP, as an expansion of our case study.

REFERENCES

[1] L. Amabili. Visualizing algorithms of nonlinear dimensionality reduction techniques. 2017.

[2] M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7-9):1304–1330, 2007.

[3] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *International Donference on Database Theory*, pp. 217–235. Springer, 1999.

[4] M. Bostock, V. Ogievetsky, and J. Heer. D^3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[5] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pp. 1–8, 2014.

[6] R. J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data mining*, pp. 160–172. Springer, 2013.

[7] A. Chatzimparmpas, R. M. Martins, and A. Kerren. t-visne: Interactive assessment and interpretation of t-sne projections. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[8] L. Chen and A. Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219, 2009.

[9] J. Choo, H. Lee, J. Kihm, and H. Park. ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pp. 27–34. IEEE, 2010.

[10] B. Colange, L. Vuillon, S. Lespinats, and D. Dutykh. Interpreting distortions in dimensionality reduction by superimposing neighbourhood graphs. In *2019 IEEE Visualization Conference (VIS)*, pp. 211–215. IEEE, 2019.

[11] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[12] R. Cutura, M. Aupetit, J.-D. Fekete, and M. Sedlmair. Comparing and exploring high-dimensional data with dimensionality reduction algorithms and matrix visualizations. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pp. 1–9, 2020.

[13] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification and scene analysis*, vol. 3. Wiley New York, 1973.

[14] A. Endert, C. Han, D. Maiti, L. House, and C. North. Observation-level interaction with statistical models for visual analytics. In *2011 IEEE Conference on Visual Analytics Science and Technology*, pp. 121–130. IEEE, 2011.

[15] L. Ertöz, M. Steinbach, and V. Kumar. A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on Clustering High dimensional Data and its Applications at 2nd SIAM International Conference on Data mining*, pp. 105–115, 2002.

[16] L. Ertöz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the 2003 SIAM International Conference on Data mining*, pp. 47–58. SIAM, 2003.

[17] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, vol. 96, pp. 226–231, 1996.

[18] R. Etemadpour, L. Linsen, C. Crick, and A. G. Forbes. A user-centric taxonomy for multidimensional data projection tasks. In *IVAPP*, pp. 51–62, 2015.

[19] R. Etemadpour, R. Motta, J. G. de Souza Paiva, R. Minghim, M. C. F. De Oliveira, and L. Linsen. Perception-based evaluation of projection methods for multidimensional data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 21(1):81–94, 2014.

[20] S. G. Fadel, F. M. Fatore, F. S. Duarte, and F. V. Paulovich. Loch: A neighborhood-based multidimensional projection technique for high-dimensional sparse spaces. *Neurocomputing*, 150:546–556, 2015.

[21] C. Fu, Y. Zhang, D. Cai, and X. Ren. Atsne: efficient and robust visualization on gpu through hierarchical optimization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 176–186, 2019.

[22] A. Ghosh, M. Nashaat, J. Miller, and S. Quader. Interpretation of structural preservation in low-dimensional embeddings. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[23] D. Harel and Y. Koren. Clustering spatial data using random walks. In *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 281–286, 2001.

[24] N. Heulot, M. Aupetit, and J.-D. Fekete. Proxiviz: an interactive visualization technique to overcome multidimensional scaling artifacts. *Proceedings of IEEE InfoVis, poster*, 2, 2012.

[25] A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *26th International Conference on Very Large Databases*, pp. 506–515, 2000.

[26] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571, 2011.

[27] S. K. Lam, A. Pitrou, and S. Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6, 2015.

[28] Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[29] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[30] J. A. Lee and M. Verleysen. Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants. *Procedia Computer Science*, 4:538–547, 2011.

[31] S. Lespinats and M. Aupetit. Checkviz: Sanity check and topological clues for linear and non-linear mappings. In *Computer Graphics Forum*, vol. 30, pp. 113–125. Wiley Online Library, 2011.

[32] S. Lespinats, M. Verleysen, A. Giron, and B. Fertil. Dd-hds: A method for visualization and exploration of high-dimensional data. *IEEE Transactions on Neural Networks*, 18(5):1265–1279, 2007.

[33] R. Liu, H. Wang, and X. Yu. Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Information Sciences*, 450:200–226, 2018.

[34] S. Liu, B. Wang, P.-T. Bremer, and V. Pascucci. Distortion-guided structure-driven interactive exploration of high-dimensional data. In *Computer Graphics Forum*, vol. 33, pp. 101–110. Wiley Online Library, 2014.

[35] Y. Liu, E. Jun, Q. Li, and J. Heer. Latent space cartography: Visual analysis of vector space embeddings. In *Computer Graphics Forum*, vol. 38, pp. 67–78. Wiley Online Library, 2019.

[36] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[37] R. M. Martins, D. B. Coimbra, R. Minghim, and A. C. Telea. Visual analysis of dimensionality reduction quality for parameterized projections. *Computers & Graphics*, 41:26–42, 2014.

[38] R. M. Martins, R. Minghim, A. C. Telea, et al. Explaining neighborhood preservation for multidimensional projections. In *CGVC*, pp. 7–14, 2015.

[39] L. McInnes and J. Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 33–42. IEEE, 2017.

[40] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[41] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[42] T. Metsalu and J. Vilo. Clustvis: a web tool for visualizing clustering of multivariate data using principal component analysis and heatmap. *Nucleic acids research*, 43(W1):W566–W570, 2015.

[43] M. Moor, M. Horn, B. Rieck, and K. Borgwardt. Topological autoencoders. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

[44] R. Motta, R. Minghim, A. de Andrade Lopes, and M. C. F. Oliveira. Graph-based measures to assist user assessment of multidimensional projections. *Neurocomputing*, 150:583–598, 2015.

[45] F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[46] E. J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre. Clustersculptor: A visual analytics tool for high-dimensional data. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pp. 75–82. IEEE, 2007.

[47] A. Narayan, B. Berger, and H. Cho. Density-preserving data visualization unveils dynamic patterns of single-cell transcriptomic variability. *bioRxiv*, 2020.

[48] J. Nickolls, I. Buck, M. Garland, and K. Skadron. Scalable parallel

programming with cuda: Is cuda the parallel programming model that application developers have been waiting for? *Queue*, 6(2):40–53, 2008.

[49] L. G. Nonato and M. Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2650–2673, 2018.

[50] L. Nováková and O. Štepanková. Radviz and identification of clusters in multidimensional data. In *2009 13th International Conference Information Visualisation*, pp. 104–109. IEEE, 2009.

[51] L. Nováková and O. Štěpánková. Visualization of trends using radviz. *Journal of Intelligent Information Systems*, 37(3):355–369, 2011.

[52] F. V. Paulovich, C. T. Silva, and L. G. Nonato. Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1281–1290, 2010.

[53] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[54] D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *In Proceedings of the 17th International Conference on Machine Learning*, vol. 1, pp. 727–734, 2000.

[55] N. Pezzotti, T. Höllt, B. Lelieveldt, E. Eisemann, and A. Vilanova. Hierarchical stochastic neighbor embedding. In *Computer Graphics Forum*, vol. 35, pp. 21–30. Wiley Online Library, 2016.

[56] J. Poco, R. Etemadpour, F. V. Paulovich, T. Long, P. Rosenthal, M. d. Oliveira, L. Linsen, and R. Minghim. A framework for exploring multidimensional data with 3d projections. In *Computer Graphics Forum*, vol. 30, pp. 1111–1120. Wiley Online Library, 2011.

[57] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[58] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):241–250, 2016.

[59] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2634–2643, 2013.

[60] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory. A taxonomy of visual cluster separation factors. In *Computer Graphics Forum*, vol. 31, pp. 1335–1344. Wiley Online Library, 2012.

[61] C. Seifert, V. Sabol, and W. Kienreich. Stress maps: Analysing local phenomena in dimensionality reduction based visualisations. In *EuroVAST @ EuroVis*, 2010.

[62] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson Education India, 2016.

[63] J. Tang, J. Liu, M. Zhang, and Q. Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, pp. 287–297, 2016.

[64] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer Science & Business Media, 2012.

[65] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[66] L. Van Der Maaten. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

[67] J. Venna and S. Kaski. Local multidimensional scaling. *Neural Networks*, 19(6-7):889–899, 2006.

[68] S. Vijendra. Efficient clustering for high dimensional data: Subspace based clustering and density based clustering. *Information Technology Journal*, 10(6):1092–1105, 2011.

[69] Y. Wang, K. Feng, X. Chu, J. Zhang, C.-W. Fu, M. Sedlmair, X. Yu, and B. Chen. A perception-driven approach to supervised dimensionality reduction for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(5):1828–1840, 2017.

[70] M. Wattenberg, F. Viégas, and I. Johnson. How to use t-sne effectively. *Distill*, 2016. doi: 10.23915/distill.00002

[71] J. Wenskovitch, M. Dowling, and C. North. With respect to what? simultaneous interaction with dimension reduction and clustering projections. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pp. 177–188, 2020.

[72] J. Xia, F. Ye, W. Chen, Y. Wang, W. Chen, Y. Ma, and A. K. Tung. Ldsscanner: Exploratory analysis of low-dimensional structures in high-dimensional datasets. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):236–245, 2017.

[73] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.