# Constraint Preserving Score for Automatic Hyperparameter Tuning of Dimensionality Reduction Methods for Visualization

Vu, Viet Minh; Bibal, Adrien; Frénay, Benoît

*Published in:*
IEEE Transactions on Artificial Intelligence

*Publication date:*
2021

*Document Version*
Peer reviewed version

# Constraint Preserving Score for Automatic Hyperparameter Tuning of Dimensionality Reduction Methods for Visualization

Viet Minh Vu, Adrien Bibal, and Benoît Frénay, *Member, IEEE*

***Abstract*—In data analysis, visualization through dimensionality reduction (DR) is one of the most effective ways to understand a dataset. However, the quality of a visualization is hard to evaluate quantitatively and the hyperparameters of visualization algorithms are sometimes difficult to tune for end-users. This paper proposes a score for visualization assessment that can be used to ease the choice of hyperparameter values for widely used DR methods like $t$-distributed stochastic neighbor embedding ($t$-SNE), LargeVis, and uniform manifold approximation and projection (UMAP). We present the *constraint preserving score*, a computationally efficient score to measure visualization quality. The idea is to measure how well a visualization preserves the information encoded in pairwise constraints like group information or similarity/dissimilarity relationships between instances. Based on this quantitative measure, we use Bayesian optimization to effectively explore the solution space of all visualizations and find the most suitable one. The proposed score is flexible as it can measure quality in different ways depending on the provided constraints. Experiments show its interest for end-users, its complementarity with existing visualization quality measures, and its flexibility to easily express different quality aspects.**

***Impact Statement*—When working with high dimensional data, visualization techniques are useful tools to help us to understand patterns in data. Widely used visualization methods such as t-SNE, LargeVis and UMAP require tuning several hyperparameters, which is a tedious task for end-users. The visualizations are usually assessed qualitatively and subjectively by users since we lack quantitative measures that fit their needs. Our work tackles this problem by proposing a novel score based on user's constraints to measure visualization quality. This score can thus be used to automatically tune the hyperparameters of visualization methods. For real-world datasets, there are typically multiple aspects hidden in the data under the form of local or global structures, or relationships between data groups. One visualization gives us one vantage point to look at the data and thus reveals one specific aspect of the data. Assessing the visualization quality is still an open question and each state-of-the-art visualization quality metric is designed to capture only one specific aspect like local neighborhood structure. However, our proposed constraints preserving score can capture other different aspects of the visualization like the global structure or semantic relationships between groups according to the information encoded in the input constraints. Our score measures how well the information encoded in input constraints is preserved in a visualization, and suggests the best visualization corresponding to the users' needs. This score can have a large impact since it is very easy to use and works with any visualization method. Domain experts can express their knowledge in a simple form of similar or dissimilar groups of points. If needed, end-users can use a small amount of labeled data to express their constraints.**

V.M.Vu, A.Bibal, B.Frénay are with the University of Namur, Belgium. E-mail: { vuvietminh, adrien.bibal, benoit.frenay }@unamur.be

Manuscript received Xxx xx, 20xx; revised Xxx xx, 20xx.

***Index Terms*—Bayesian Optimization, Dimensionality Reduction, Hyperparameter Tuning, Pairwise Constraints, Visualization.**

## I. Introduction

Dimensionality reduction (DR) methods transform data from a high dimensional (HD) space into a low dimensional (LD) space while preserving relevant structures. Modern DR methods like $t$-distributed stochastic neighbor embedding ($t$-SNE) [1], LargeVis [2] and uniform manifold approximation and projection (UMAP) [3] aim to create visualizations that help users to get insights. These techniques are powerful, but their hyperparameters must be carefully tuned while often being hard to understand for end-users. Choosing good hyperparameter values is crucial, since it predetermines the quality and usefulness of the visualization [4], [5]. Typically, the most suitable visualization is chosen through trial-and-error, which is tedious and difficult for users.

Automatically finding the best hyperparameter values of DR techniques, such as the *perplexity* of $t$-SNE, is crucial before undertaking an exploratory data analysis. However, this task involves two major difficulties: (i) the definition of a measure of the visualization quality for choosing the best hyperparameter value and (ii) the search through the hyperparameter space to find this value. This paper tackles both these issues by assessing the quality of visualizations in an efficient and faithful way. Our idea is to use the semantic information in pairwise constraints to measure the quality of visualizations. This is done by transforming the relationships between object pairs into a quantitative measure. The main contribution of this paper is a new measure called the *constraint preserving score* (f_score) that measures the quality of visualizations from any DR method. This score provides a different aspect of quality than state-of-the-art visualization quality measures (e.g., [6], [7]), while being computationally efficient and flexible. An important application of our proposed score is for automatically tuning hyperparameters of visualization methods. This score can be used with Bayesian optimization [8], [9] to find a range of hyperparameter values corresponding to the visualizations that best respect the user needs encoded in the input constraints.

One key advantage of our score is that, as we find the best hyperparameter values with a model-independent measure of quality, DR methods do not need to be modified. In that sense, our approach is DR-method agnostic. Furthermore, when using

constraints for choosing the best hyperparameter values, visualizing these constraints makes it possible to explain the choice of visualization. By explaining how the visualization is chosen, a step towards interpretability of the DR process is also taken. End-users can use our method as a black-box hyperparameter tuning toolbox, but DR experts can also analyze the impact of hyperparameters on the quality of visualizations.

This paper is organized as follows. Section II presents the background on DR methods, visualization quality metrics, pairwise constraints in unsupervised learning and an overview of how the automatic hyperparameter selection for DR methods is handled in the literature. Section III presents how to transform the knowledge in the input constraints into the constraint preserving score. The experimental setting for evaluating our proposed method is described in Section IV. The main characteristics of the proposed score are empirically presented through experiments in Section V. We compare our score to other visualization quality metrics in Section VI and show how to apply Bayesian optimization on this score to automate the hyperparameter tuning task in Section VII. Finally, Section VIII concludes our work.

## II. Background and Related Work

This section presents the background and methods related to our work. Typically, dimensionality reduction (DR) methods can be useful for very high dimensional data such as hyperspectral imagery to facilitate downstream tasks [10], [11]. However, in the scope of this paper, we focus on DR methods for visualization, which are widely used for exploratory data analysis. Section II-A presents the DR techniques used in our evaluation ($t$-SNE [1], LargeVis [2] and UMAP [3]). Section II-B presents the quality measures used in the literature to assess DR embeddings. Section II-C describes how user constraints are used in clustering and in DR. Finally, Section II-D reviews the techniques to choose the hyperparameters of DR algorithms.

### A. Dimensionality Reduction for Visualization

$t$-SNE, LargeVis and UMAP are widely used for visualization and have in common that they preserve local structures in data. They consist of two main steps. First, a neighborhood graph is constructed from the high-dimensional (HD) data. This step requires an hyperparameter that determines the size of the set of $k$-nearest neighbors ($k$NN), called n_neighbors in UMAP and *perplexity* in $t$-SNE and LargeVis. This $k$NN graph is weighted in different ways to transform similarities in the data space into neighborhood probabilities. Second, these probabilities are used to project data in a low-dimensional (LD) space to obtain the visualization.

Constructing the $k$NN graph requires pairwise distances between all $n$ instances in a $d$-dimensional space and has a complexity of $\mathcal{O}(dn^2)$. $t$-SNE [1] constructs the exact $k$NN graph and thus cannot scale with large datasets. Its accelerated version, called Barnes-Hut $t$-SNE [12], uses a tree-based algorithm to reduce the complexity to $\mathcal{O}(dn \log n)$. LargeVis [2] approximates a very accurate $k$NN graph by using random projection trees to obtain neighborhood candidates for each instance. In $t$-SNE and LargeVis, edges in the $k$NN graph are weighted by an isotropic Gaussian kernel with an adapted bandwidth derived from the perplexity parameter. UMAP [3] has a different theoretical foundation and uses a more sophisticated topological data analysis to model local connectivity by a fuzzy topological structure.

In the embedding space, all three methods create a neighborhood graph and transform it to neighborhood probabilities using the Student's $t$-distribution (UMAP uses a similar but more general function). A graph layout problem must then be solved to match the neighborhood probabilities in HD and LD. $t$-SNE solves it by minimizing their Kullback-Leibler divergence. LargeVis models the probability of obtaining an edge between neighborhood nodes in the LD space and maximizes the log-likelihood of this model. UMAP considers the graphs in the HD and LD spaces as fuzzy sets and minimizes the cross entropy between them. All three methods use gradient descent for optimization.

The quality of the output embedding depends heavily on the hyperparameters of these methods, which control the construction of the $k$NN graph in the HD space and the structure of the $k$NN graph in the LD space. The perplexity/n_neighbors determines the approximate number of neighbors for each instance: small values reveal more local structures, while large values reveal more global structures in the data. UMAP also uses another hyperparameter (min_dist) to determine the minimum distance between points in the embedding in order to directly control how tight the groups are formed in the visualization. The goal of this paper is to provide a score that can be used to automatically tune these hyperparameters.

TABLE I: Properties of the five cluster-label-agnostic quality metrics considered in this paper to assess visualizations.

| metric | range | description |
|---|---|---|
| CC | $[0, 1]$ | Pearson correlation coefficient between pairwise distance vectors |
| NMS | $[0, +\infty)$ | stress based on comparison of pairwise distance orders |
| CCA | $[0, +\infty)$ | stress with emphasis put on LD |
| NLM | $[0, +\infty)$ | stress with emphasis put on HD |
| AUC$[R_{NX}]$ | $[-1, 1]$ | how neighbors in HD are preserved in LD |

### B. Visualization Quality Metrics

Several metrics exist to evaluate the quality of embeddings. In this paper, clustering-based quality measures are not considered because they need labeled data for measurement. Table I summarizes the reviewed metrics; more mathematical details are provided in Appendix A. The *correlation coefficient* (CC) [13] computes the correlation between the pairwise distances in HD and LD. The well-known *Kruskal's non-metric stress* (NMS) [14], often used as the objective function of non-metric multidimensional scaling, compares the pairwise distance orders in HD and LD. The *curvilinear component analysis stress* (CCA) [15] is a variant of Kruskal's stress with an emphasis on the embedding distances. This metric evaluates the embedding quality by focusing on the correctness

of close instances in LD. The *Sammon's non-linear mapping stress* (NLM) [16] is similar to CCA, but focuses on the closeness of instances in HD. Finally, AUC$[R_{NX}]$, a rank-based criterion, measures how well neighborhoods in HD are preserved in LD [17]. An average normalized intersection of the neighborhood sets in the two spaces is calculated for different neighborhood sizes $k$ in a logarithmic scale. The area under this curve then gives the AUC$[R_{NX}]$ score that assesses the average DR quality on all scales [7].

### C. User Constraints for Clustering and DR

Clustering is a machine learning problem whose goal is to find groups (called *clusters*) in the data. User constraints can incorporate domain expertise to enforce expected properties of the clusters. COP-KMeans [18] is the first method that combines KMeans and pairwise constraints. Must-link and cannot-link constraints indicate that two instances must be in the same cluster or cannot be in the same cluster, respectively. The popular survey by Davidson et al. [19] distinguishes *constraint-based* and *distance-based* clustering methods with instance-level constraints. In constraint-based methods, the clusters are formed to preserve the constraints as much as possible [20], [21]. In distance-based methods, the constraints are first used to train a distance function that is later used by a clustering algorithm [22], [23].

In DR, users can also inject constraints to enforce properties in the visualization. These *objective* constraints can be partial labels as in semi-supervised latent Dirichlet allocation [24], or constraints on the value of features as in bounded PCA [25]. If users interact with the visualization, they can give feedback in form of instance-level *subjective* constraints. Pairwise constraints are often used to attract points connected by similar links and repulse points connected by dissimilar links. Such constraints are used in pairwise constraint-guided feature projection [26], semi-supervised DR [27], graph-driven constrained DR via linear projection [28] and constrained locality preserving projections [29]. Sacha et al. [30] review more methods for integrating user interaction into DR techniques. Endert et al. [31] propose a wider survey on integrating machine learning into visual analysis.

### D. Choosing Hyperparameter Values of DR Methods

The best hyperparameter values of DR methods depend on the dataset characteristics such as its size, its topology, or the distribution of the instances, which makes them hard to tune. For instance, the suggested values for $t$-SNE's perplexity are between 5 and 50 [1]. However, in practice, the embedding can change drastically between two different perplexity values. Therefore, there is no evidence that the suggested perplexities are good for all datasets. The original $t$-SNE paper also proposes a simple method to select a good perplexity by looking at the Kullback-Leibler (KL) loss produced by several perplexities and choose the lowest one. However, the KL loss tends to decrease when the perplexity increases [32], which is confirmed by our experiments, as shown in Fig. 1. For this reason, using the KL loss for evaluating the embedding quality is not suitable, since a very high perplexity would always be chosen. In practice, users have to manually choose this hard-to-understand hyperparameter, which is tedious and error-prone.

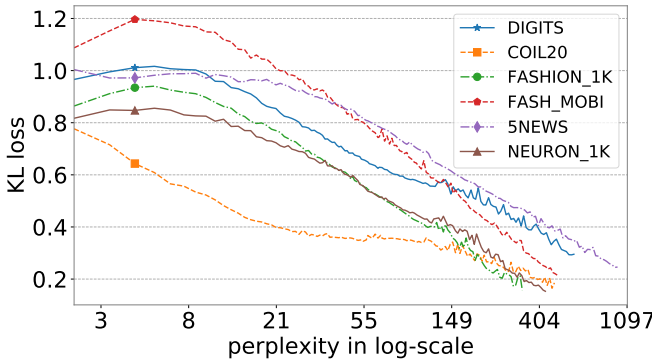



Fig. 1: Evolution of the KL loss for several datasets, which tends to decrease systematically as the perplexity increases. Perplexities are chosen in logarithmic scale from $[2, n/3]$.

Few papers in the literature attempt to derive the best hyperparameter values for DR methods automatically. Strickert [33] suggests using rank-based data to avoid perplexity calculation. Lee et al. [34] use a multi-scale approach by averaging all neighborhood sizes. Despite bypassing the perplexity selection problem, these two solutions do not solve the selection problem itself. Cao and Wang [32] try to tackle the problem by selecting the perplexity of $t$-SNE that minimizes a modified Bayesian information criteria [6]

$$\text{BIC} = 2\text{KL}(P||Q) + \frac{\text{perplexity}}{n}\log(n), \tag{1}$$

where KL$(P||Q)$ is the KL loss of $t$-SNE and $n$ is the number of instances. However, this method is only designed for $t$-SNE and cannot inject user knowledge through constraints. In summary, tuning hyperparameters for complex methods like UMAP or $t$-SNE is still an open problem.

## III. Constraint Preserving Score

This section presents the proposed constraint preserving score. We first illustrate the pairwise constraints used in this work (Section III-A), then explain how to quantify the satisfaction of these constraints to use as a score (Section III-B).

### A. Introduction to the User Pairwise Constraints

Humans can often distinguish similar and dissimilar high-dimensional objects (e.g., comparing images by visual features such as the shape, colors, or objects therein) and group them. For instance, we can easily identify three different groups among the clothing images from the Fashion product dataset in Fig. 2: the three t-shirts look similar, while being different from the shoes and the belts. Here, users move from low-level comparison between individual objects to higher-level abstractions such as groups of similar objects.

Our idea is to use the information given by pairwise links between objects to evaluate the quality of a visualization. Modern visualization methods such as $t$-SNE, LargeVis and UMAP preserve the local structures in the dataset, i.e., instances that

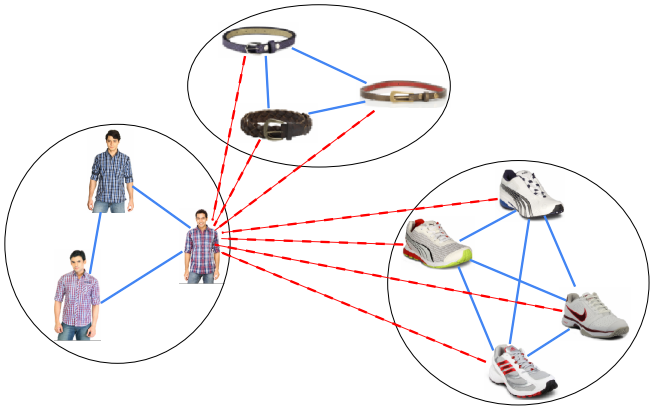


Fig. 2: Examples of the generated pairwise constraints from three different groups of sample images in Fashion product dataset. Similar links (plain blue) indicate images in the same groups. Dissimilar links (dashed red) indicate images of different groups.

are similar in HD should be close in the embedding space. These methods are considered as successful when they reveal clear groups of similar instances. If one knows in advance examples of such patterns, they can be used to assess the quality of the visualization.

Two types of pairwise constraints are considered here: similar and dissimilar links. Similar links indicate that two instances are similar and should be in a same group. On the contrary, dissimilar links indicate that two instances are dissimilar and should be in different groups. These pairwise constraints are used to measure how well local structures are preserved and to measure the quality of the visualization. In general, a convenient way to form groups is to use class labels. In the case where labels are not available, users can observe some input data points (e.g., images or documents in the dataset) and select groups of similar points to construct the pairwise constraints. However, this approach only works with datasets of images or structured documents where users can visually or semantically compare objects to find similar/dissimilar pairs. Our method may not work with unlabeled numerical data for which directly comparing data points is not straightforward.

However, several solutions exist to address this issue. If the input data are normalized, users can use a simple heuristic to select similar/dissimilar points based on their distances. Moreover, if users have prior knowledge about their dataset, they can form their groups of interest using one or several selected features. For example, in medical datasets, several standard features like sex, age, blood pressure (BP) are commonly available. The users can easily form simple groups of male and female patients, or different contrastive groups of young patients with low BP, aged patients with high BP, etc. It should be noted that we do not need the ground truth class labels, but only need groups of similar objects to form the pairwise constraints. This information can be considered as a weak supervision information [35], which can be collected efficiently and then enriched by the label propagation or active constraint selection algorithms [36]. Snorkel, a research and industrial library, is a useful tool that allows users to use their domain knowledge or heuristics to label their data [37].

### B. Defining the Constraints Preserving Score

Given a set of user pairwise constraints, the *constraint preserving score* $f_{\text{score}}$ measures how well the pairwise constraints are preserved in a particular embedding. We first propose how to quantify the satisfaction of individual constraints and we then formulate $f_{\text{score}}$ based on them.

*Constraint Measurement:* We first define the *strength* of the input pairwise constraints in a given embedding. A similar link should have a high strength and a dissimilar link should have a low strength. The strength of a constraint can be measured as the inverse of the distance between two connected points. If a Student's $t$ distribution is placed at the point $\mathbf{y}_i$ in the embedding, the strength of the constraint connecting $\mathbf{y}_i$ to another point $\mathbf{y}_j$ is defined as

$$q_{ij} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k \neq l} (1 + ||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}, \tag{2}$$

where the denominator is a normalization constant calculated from all pairs $\{(\mathbf{y}_k, \mathbf{y}_l)\}$ in the embedding.

A similar formulation is used in $t$-SNE, LargeVis and UMAP to model the neighborhood relationship in the embedding space. $q_{ij}$ can be interpreted as the probability of $\mathbf{y}_i$ and $\mathbf{y}_j$ being neighbors in the embedding space. Therefore, for each similar link $(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{S}$, $q_{ij}$ should be high. Inversely, $q_{ij}$ is expected to be low for each dissimilar link $(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{D}$.

*Constraint Preserving Score:* We propose to measure the preservation of all similar links $(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{S}$ in an embedding as the log-likelihood

$$f_{score}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \log \prod_{(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{S}} q_{ij} = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{S}} \log q_{ij}. \tag{3}$$

If all pairs of points connected by a similar link are close in the visualization, the log-likelihood $f_{score}(\mathcal{S})$ is high.

In contrast, the probability $q_{ij}$ for each dissimilar link $(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{D}$ should be low. For all dissimilar links, we therefore propose to use the negative log-likelihood

$$f_{score}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|} \log \prod_{(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{D}} q_{ij} = -\frac{1}{|\mathcal{D}|} \sum_{(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{D}} \log q_{ij}. \tag{4}$$

Another way to measure how well a dissimilar link $(\mathbf{y}_i, \mathbf{y}_j)$ is preserved is to use $1 - q_{ij}$. However, in practice, the value of $q_{ij}$ is very small, meaning that $1 - q_{ij}$ is close to one, which makes the log-likelihood of all dissimilar links vanish.

As the scores for the similar (Eq. 3) and dissimilar (Eq. 4) constraints are normalized by the number of similar and dissimilar constraints, an equal contribution of these different kinds of constraints is considered. The final constraint preserving score is therefore the combination with equal contribution of both similar links and dissimilar links

$$f_{score}(\mathcal{S}, \mathcal{D}) = \frac{1}{2} f_{score}(\mathcal{S}) + \frac{1}{2} f_{score}(\mathcal{D}). \tag{5}$$

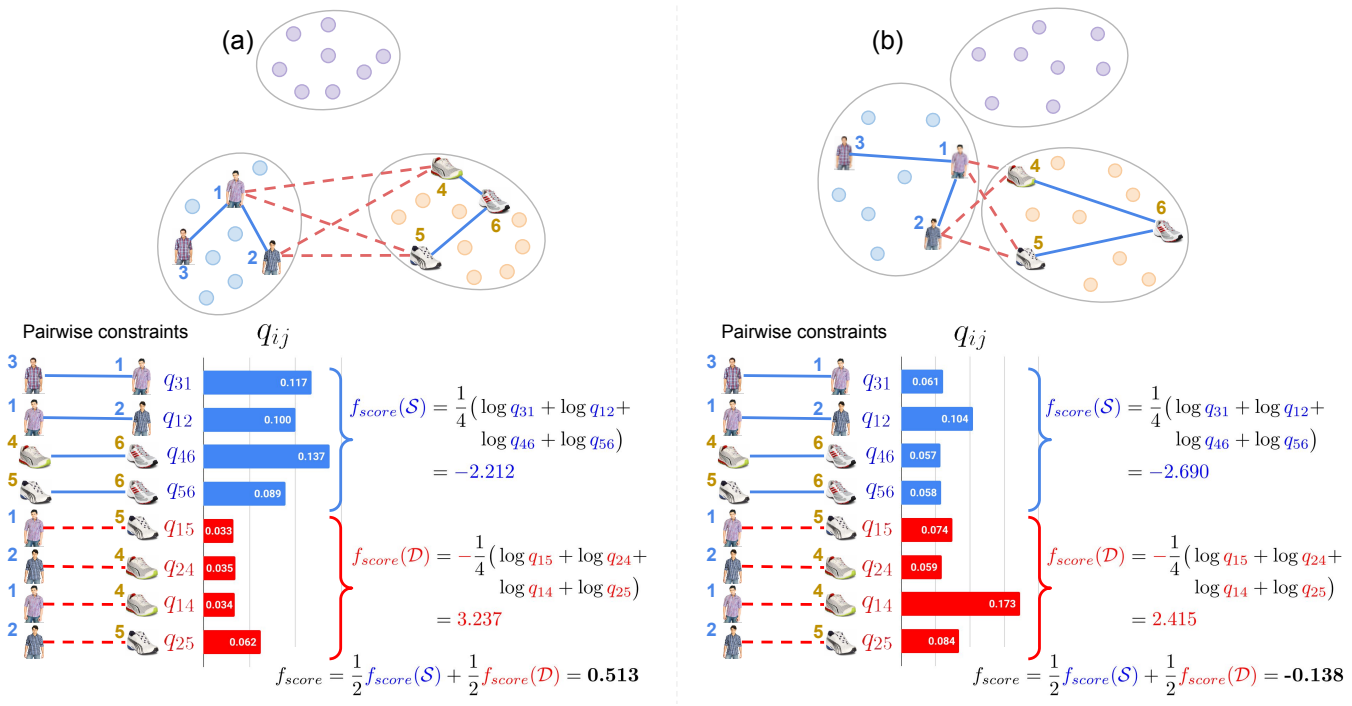



Fig. 3: Illustration of how the proposed $f_{\text{score}}$ assesses the visualizations. Two different visualizations (a) and (b) of the same dataset are shown on top with the same set of pairwise constraints including four similar links denoted by plain blue lines and four dissimilar links denoted by dotted red lines. For each pair $(i, j)$ in the input constraints, the quantitative measure $q_{ij}$ is calculated by Eq. 2 and is visualized by the bar charts. $f_{\text{score}}$ for the similar/dissimilar links and the final $f_{\text{score}}$ are calculated using Equations 3, 4 and 5. In the visualization (a), the selected images connected by must-links are close and thus the $q_{ij}$ values are large, while the images connected by cannot-links are distant, which makes the $q_{ij}$ values small. The values of $q_{ij}$ for the same set of input constraint for the visualization (b) are opposite since the similar images are placed far apart while the dissimilar ones are closer. $f_{\text{score}}$ can measure this difference and give a high score for (a) and a much lower score for (b).

An embedding that retains as much as possible the pairwise constraint corresponds to a high $f_{\text{score}}$, which means that it has a good quality with respect to the encoded knowledge. Fig. 3 illustrates the idea of how to measure the preservation of input constraints quantitatively using $f_{\text{score}}$.

## IV. Experimental Setup

In this paper, $f_{\text{score}}$ is used to assess visualizations and to find the best hyperparameters of three DR methods: $t$-SNE, LargeVis and UMAP. This section discusses the datasets used in experiments, as well as how pairwise constraints are obtained and how metrics are computed. We analyze the characteristics of $f_{\text{score}}$ in Section V, compare it with other metrics in Section VI and use this score to automatically tune hyperparameters of DR methods in Section VII.

### A. Experimental Datasets

Six datasets of gray-scale and color images, texts and gene expressions are used for evaluation. DIGITS is a subset of 1797 handwritten digits of gray-scale 8x8 images [38]. COIL20 contains 1494 gray-scale 32x32 images of 20 rotated objects [39]. FASHION_1K contains 1000 gray-scale 28x28 images sampled from the Fashion-MNIST clothing dataset [40]. FASH_MOBI contains 1494 color images of the seven most numerous classes sampled from another real-world fashion product images dataset [41]. The features are extracted with a pre-trained MobileNet [42], where the last fully connected layer is replaced by a global average pooling layer to obtain an output vector of 1280 dimensions. For these four image datasets, PCA is applied to keep 90% variance of the data. This speeds up the computation of pairwise distances and reduce the potential noise of outliers.

5NEWS contains 2957 text documents in 5 groups (*rec.autos, rec.sport.baseball, sci.space, sci.crypt* and *comp.sys.mac.hardware*) from the 20 Newsgroups dataset. We use a traditional pipeline to process the text data. Documents are first converted to TF-IDF vectors that are then fed into a latent Dirichlet allocation model [43] to extract 15 hidden topics, which are the 15 features used by the DR methods.

The last real-world dataset is NEURON_1K open dataset [44] that contains 1301 brain cells from an E18 mouse. These cells have been processed and provided by 10X Genomics. The processed data have 10 PCA features and 6 labels found by a graph-based clustering method.

### B. Constraint Generation

The proposed constraint preserving score requires a set of constraints in the form of similar and dissimilar links. As shown in Section III-B, the pairwise constraints can be

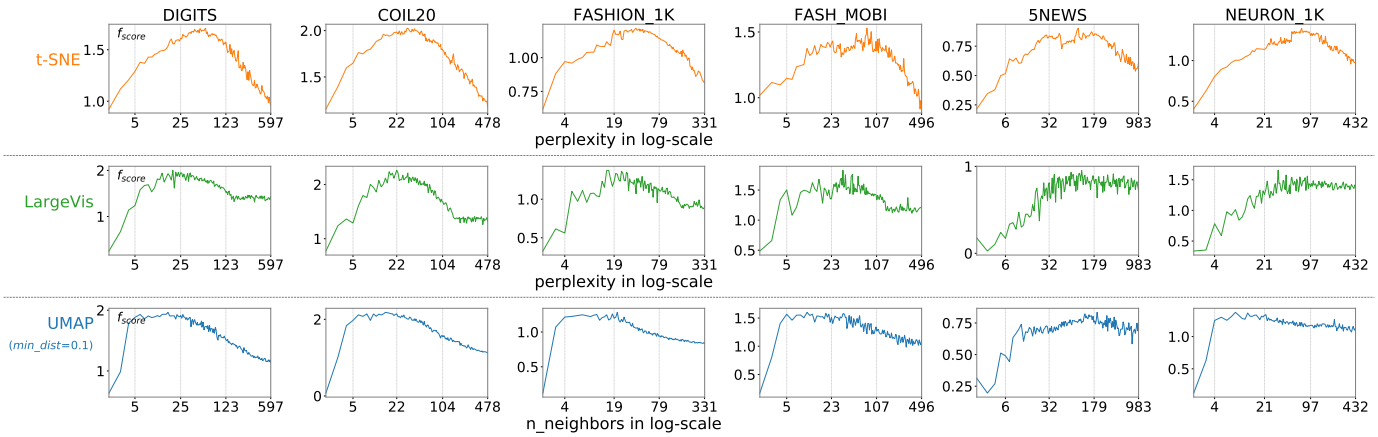



Fig. 4: Evolution of $f_{\text{score}}$ with respect to the hyperparameter of three DR methods for six datasets.

generated from groups of selected instances. Users can group the instances that they find similar to indicate that they should be connected by similar links. Similarly, instances in different groups indicate that they should be connected by dissimilar links. In order to objectively evaluate the proposed score, we use a standard setting in semi-supervised learning where only a small number of labels in the dataset are used. Pairwise constraints generated from labeled instances are used throughout our experiments as follows.

First, for a dataset of $C$ classes and $n$ instances, $k \ll n$ labeled instances are randomly selected for each class. Then, a similar link is created for each possible pair of these $k$ instances, leading to $|\mathcal{S}| = Ck(k-1)/2$ constraints. Finally, for each pair of classes, $k^2$ dissimilar links are created by considering all pairs of instances that belong to two distinct classes, leading to $|\mathcal{D}| = C(C-1)k^2/2$ constraints.

### C. Computing Visualizations and Metrics

In the following sections, a grid of hyperparameter values is created for each method to compare them and to perform their hyperparameter optimization. For $t$-SNE and LargeVis, a one-dimensional grid of perplexity values is sampled in natural logarithmic scale from $[2, n/3]$. For UMAP, a two-dimensional grid is created with n_neighbors $\in [2, n/3]$ in natural logarithmic scale and min_dist $\in [0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1.0]$. For each combination of hyperparameters in the above grids, an embedding is calculated, and $f_{\text{score}}$, AUC$[R_{NX}]$ and the BIC-based score (if applicable) are computed.

## V. Characteristics of $f_{\text{score}}$

Experiments in this section show that $f_{\text{score}}$ is a well-behaved function of the input visualization (Section V-A), is stable with respect to the number of input labeled instances (Section V-B) and is flexible with respect to different sets of input constraints (Section V-C).

$f_{\text{score}}$ has a computational complexity of $\mathcal{O}(n^2)$ since it only uses pairwise distances between the $n$ points in the visualization (and not in the original HD data). Furthermore, the summation over all the input pairwise constraints can be efficiently vectorized via matrix slicing operations. In contrast, AUC$[R_{NX}]$ must access both to the HD data and the visualization. This means that AUC$[R_{NX}]$ is not scalable for large datasets due to its complexity of $\mathcal{O}(dn^2 log(n))$. The BIC-based score, despite its simplicity, can only be used for $t$-SNE. For an embedding not generated by $t$-SNE, it requires to compute the KL loss of $t$-SNE with a complexity of $\mathcal{O}(dn^2)$. In conclusion, the proposed $f_{\text{score}}$ has the advantage to be independent of the choice of the DR method and to be computationally more efficient.

### A. $f_{score}$ as a Well-Behaved Function

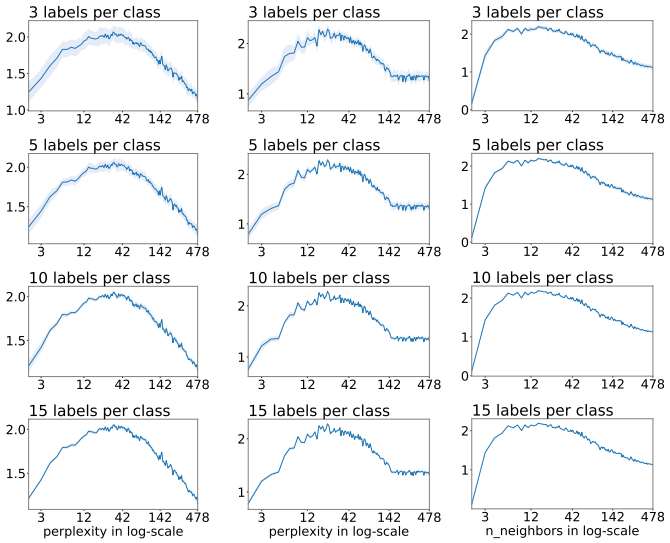
Fig. 4 shows the behavior of $f_{\text{score}}$ as a function of the perplexity for $t$-SNE and LargeVis and as a function of n_neighbors for UMAP for six datasets. In this experiment, UMAP is run with the recommended value min_dist = 0.1. This parameter is fixed, which allows us to have an overview of the evolution of $f_{\text{score}}$ with respect to the neighborhood size of all three evaluated methods. Pairwise constraints are generated from $k = 10$ labeled instances per class.

$f_{\text{score}}$ takes the form of a convex-like function of the perplexity/n_neighbors, i.e., a well-behaved function. It increases as the number of neighbors (perplexity/n_neighbors) increases, then reaches its maximum value, and finally decreases when the number of neighbors becomes too large. This also holds when $f_{\text{score}}$ is a function of two parameters (n_neighbors and min_dist) for UMAP embeddings. While $f_{\text{score}}$ is not smooth *per se*, it seems feasible to find a global maximum.

Flat regions can be found where $f_{\text{score}}$ does not change much for LargeVis. The reason is that LargeVis is designed for large datasets and, thus, when applied to medium-sized datasets, the impact of the perplexity is not significant. In contrast, $t$-SNE and UMAP are very sensitive to their hyperparameters. The experimental results of Section VI and Section VII are therefore focused on $t$-SNE and UMAP.

### B. Stability of $f_{score}$

To investigate the number of constraints needed to obtain a reliable $f_{\text{score}}$, different number $k$ of labeled instances (3, 5, 10 and 15) per class are tested. In each setting, $f_{\text{score}}$




(a) $t$-SNE (b) LargeVis (c) UMAP

Fig. 5: Stability of $f_{\text{score}}$ with the embeddings of (a) $t$-SNE, (b) LargeVis and (c) UMAP for the COIL20 dataset. The mean (blue line) and variance (filled region around the line) is calculated for each perplexity/n_neighbors with a different number $k$ of labeled instances per class (3, 5, 10 and 15).

is repeatedly evaluated 20 times. The mean and variance of $f_{\text{score}}$ for $t$-SNE, LargeVis and UMAP (with min_dist of 0.1) embeddings for the COIL20 dataset are shown in Fig. 5 as an example. It can be seen in the figure that when the number $k$ of labeled instances increases, $f_{\text{score}}$ is more stable since its variance decreases. One can also observe that the region where $f_{\text{score}}$ has a high value is stable for different number of constraints. This result is shown for COIL20, but also holds for the other datasets. Since $f_{\text{score}}$ is stable with respect to $k$, for the remaining of this paper, $k = 10$ labeled instances per class is used to calculate $f_{\text{score}}$, since it is a reasonable small number of labels for which the variance of $f_{\text{score}}$ is negligible.

### C. Flexibility of $f_{score}$

In contrast to other DR quality measures, $f_{\text{score}}$ is flexible, in the sense that the input constraints can be used to control how the visualization is assessed. In most cases, the constraints generated from class labels reflect naturally the class-relationship between the instances. However, if the labels are not available, or the users want to see other patterns in their data, they can use their specific constraints to describe their needs. This section provides concrete examples with t-SNE embeddings for three real-world datasets in which the class labels are not used. In this case, users can observe several data points and group them into different abstract groups. For example, users may not know how many categories there are in a dataset of fashion products. However, they can easily identify a footwear group of shoes and sandals, or a clothing group of dresses, shirts, and trousers. These groups can represent the semantic concepts users expect to see in the visualization. From each abstract group selected by users, $k = 10$ instances are used to generate a new set of pairwise constraints. $f_{\text{score}}$ can then find the best visualization that reflects the need of users encoded in these constraints.

The first example considers the FASH_MOBI dataset with seven sub-categories. The best visualization (perplexity = 60) shows 7 detached sub-groups as shown in the top-left plot of Fig. 6a. If the user wants to see more abstract, general groups, he or she can form higher-level groups such as

- *Accessories* as a group of { *Bag, Jewellery, Watches* },
- *Footwear* as a group of { *Sandal, Shoes* },
- *Apparel* as a group of { *Topwear, Bottomwear* }.

While the previously chosen visualization did not reveal these three higher-level groups, using them to compute $f_{\text{score}}$ lead to a new best perplexity (113) that better reveals this structure as shown in the bottom-right corner of Fig. 6a.

The second example focuses on semantic labels for the textual 5NEWS dataset. Based on the content of the news in five original classes, the user can create three general topics

- sportive records group (*rec*) as a topic of {*rec.autos, rec.sport.baseball*},
- scientific group (*sci*) as a topic of {*sci.space, sci.crypt*},
- *comp.sys.mac.hardware* stays in its own group (*comp*).

The problem of the visualization found with the constraints generated from the original class labels is that two sub-groups of the same topic can be placed far apart (bottom-left of Fig. 6b). By using the new constraints generated from the three above semantic groups, $f_{\text{score}}$ finds a better visualization in which elements in these semantic groups are placed close to each other (bottom-right of Fig. 6b).

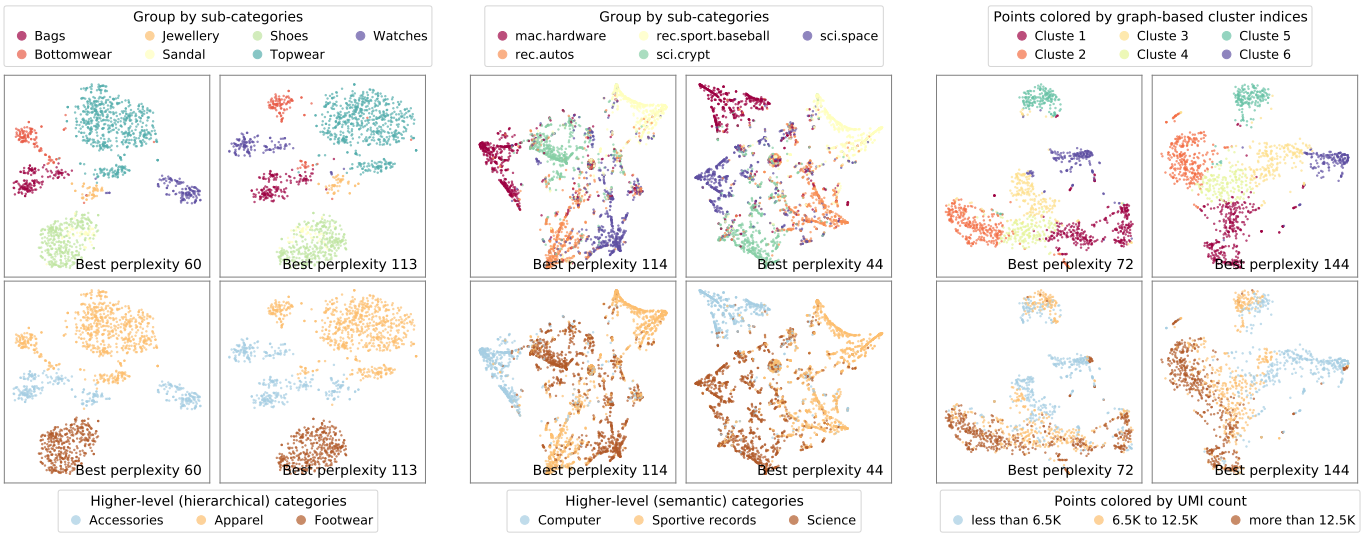
The last example uses NEURON_1K. The original 1301 cells are grouped into 6 classes found by a graph-based clustering algorithm. These classes are characterized by the transcriptome profiles of individual cells (presented in the RNA sequences). However, another important aspect to characterize individual cells is the count of absolute number of molecules: the unique molecular identifier (UMI) [45]. Therefore, the cells can be grouped into three new groups:

- the ones with less than 6.5K molecules,
- the ones having from 6.5K to 12.5K molecules,
- the ones with more than 12.5K molecules.

Fig. 6c illustrates the visualizations found by $f_{\text{score}}$ with the constraints generated from the original graph-based clusters and from the new groups obtained with the UMI count.

## VI. Comparison with Other Quality Scores

Our proposed $f_{\text{score}}$ can be used as a quality measure like other state-of-the-art scores such as BIC or AUC$[R_{NX}]$. This section qualitatively compares the best visualizations found by $f_{scores}$ and by two other metrics. $f_{\text{score}}$ is compared with AUC$[R_{NX}]$ and the BIC-based score for evaluating $t$-SNE embeddings (Section VI-A). $f_{\text{score}}$ is also compared with AUC$[R_{NX}]$ for evaluating UMAP embeddings (Section VI-B). As indicated previously, $f_{\text{score}}$ is used with $k = 10$ labeled instances for each class. Our experiments in Section V show that this parameter does not need to be tuned since it gives a stable score throughout the empirical measures for all three methods and six experimented datasets.




(a) FASH_MOBI dataset (b) 5NEWS dataset (c) NEURON_1K dataset

Fig. 6: Flexibility of $f_{\text{score}}$ for $t$-SNE embeddings. Each dataset is shown in four plots that correspond to the four possible combination of two perplexities and two coloring schemes. The two plots on the left (right) show the best visualization found by maximizing $f_{\text{score}}$ with the original labels (with higher-level categories used as labels). The plots in the top row are colored with the original labels, while the ones in the bottom row are colored with higher-level categories.

### *A. Comparison of $f_{score}$ with AUC$[R_{NX}]$ and the BIC-based Score for $t$-SNE*

Fig. 7 shows that, for the six selected datasets, $f_{\text{score}}$ agrees with AUC$[R_{NX}]$, the BIC-based score or both of them. The agreement between these scores can be visually revealed through the overlap of the ranges of the top 5% scores (maximum values for $f_{\text{score}}$ and AUC$[R_{NX}]$, minimum values for the BIC score) in Fig. 7.

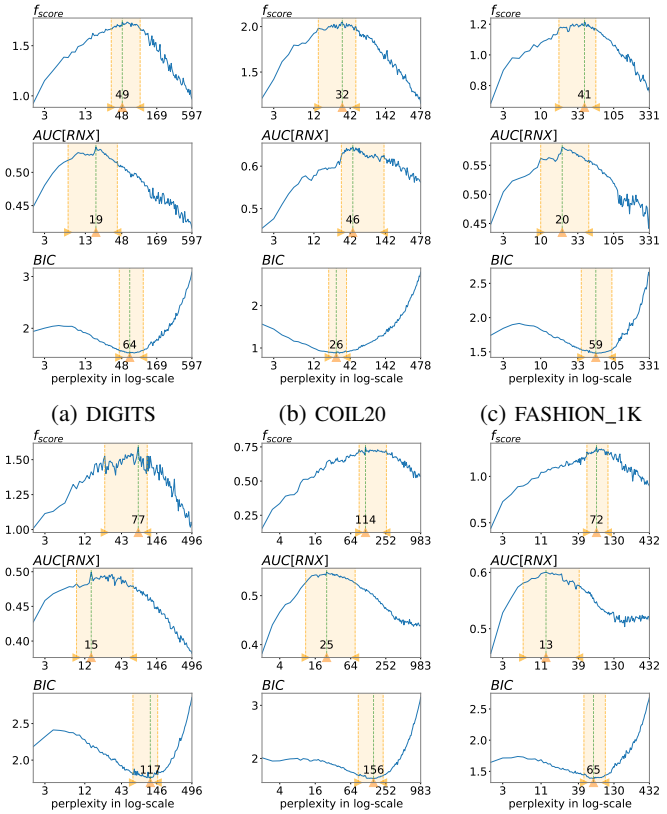
In order to compare thoroughly the best solutions found by these scores, metamaps are used for visualizing the solution space of DR methods. Each point in the metamap is a $t$-SNE embedding corresponding to a perplexity value. Two points close to each other in the metamap correspond to perplexities that provide similar visualizations. The metamap can be constructed with any embedding method such as UMAP or $t$-SNE and is extremely useful in visual analytic tools like VisCoDer [46] for discovering and comparing embeddings of different DR methods. In the case demonstrated in Fig. 9, we have more than 100 visualizations for the NEURON_1K dataset. The metamaps are built using UMAP with large values of n_neighbors = 50 and min_dist = 0.5, which allow us to have a global view of all visualizations corresponding to different perplexities.

Fig. 9 shows the metamaps for NEURON_1K and highlights several visualizations selected by different scores. The four metamaps are colored by the values of perplexity, $f_{\text{score}}$, AUC$[R_{NX}]$ and the BIC-based score. The 5% of embeddings with the highest scores are highlighted. It can be seen that the three scores reveal different visualizations: different scores can select visualization with different qualities. This is in line with Wattenberg et al. [4], who state that we need more than one visualization to understand the hidden patterns in HD data. The visualizations at the bottom of Fig. 9 serve as a qualitative evaluation of the best visualizations found by the three scores.

### *B. Comparison of $f_{score}$ with AUC$[R_{NX}]$ for UMAP*

Fig. 8 shows the evolution of $f_{\text{score}}$ and AUC$[R_{NX}]$ when the two hyperparameters n_neighbors and min_dist of UMAP are considered. For DIGITS, COIL20 and FASHION_1K, the evolution of $f_{\text{score}}$ is clearer and smoother than the one of AUC$[R_{NX}]$. For NEURON_1K, the two scores discover different optimal regions. For FASH_MOBI and 5NEWS, AUC$[R_{NX}]$ reveals clearer regions of best hyperparameters, but it mostly gives the same score for different min_dist while n_neighbors is fixed. In contrast, $f_{\text{score}}$ discovers the influence of min_dist in conjunction with n_neighbors. The combination of these two hyperparameters is important for UMAP embeddings, since while n_neighbors controls local structures (the size of local neighborhoods), min_dist controls directly how tight the groups in the visualization are.

Fig. 10 shows metamaps for UMAP embeddings and several selected visualizations for COIL20. In this case, we have more than 1000 visualizations of the COIL20 dataset corresponding to different combinations of n_neighbors and min_dist. The metamaps are built using UMAP with a large neighbor size (n_neighbors = 100, min_dist = 0.5) in order to obtain a global view of all visualizations. $f_{\text{score}}$ considers the first visualization (a) as the best one. The next two visualizations are considered good by AUC$[R_{NX}]$, but not by $f_{\text{score}}$. In the second visualisation (b), the groups clearly highlight the local structures, but are not tight enough to reveal the global structures. In the third visualization (c), the groups are retracted and heavily overlap each other. This visualization has a high AUC$[R_{NX}]$ score since the neighborhood information is well preserved, while




(a) DIGITS (b) COIL20 (c) FASHION_1K

(d) FASH_MOBI (e) 5NEWS (f) NEURON_1K

Fig. 7: Comparison of $f_{\text{score}}$, AUC$[R_{NX}]$ and the BIC-based score for $t$-SNE embeddings. (b), (c): the ranges of the top 5% of maximum values (minimum values for the BIC score) overlap each other for the three scores. (d): $f_{\text{score}}$ range mainly overlaps with AUC$[R_{NX}]$ score range. (a), (e) and (f): $f_{\text{score}}$ ranges only overlap with the BIC-based score ranges. The best perplexity selected by each score (marked by the green vertical line) gives an idea of what is the good range of perplexity values according to each score.

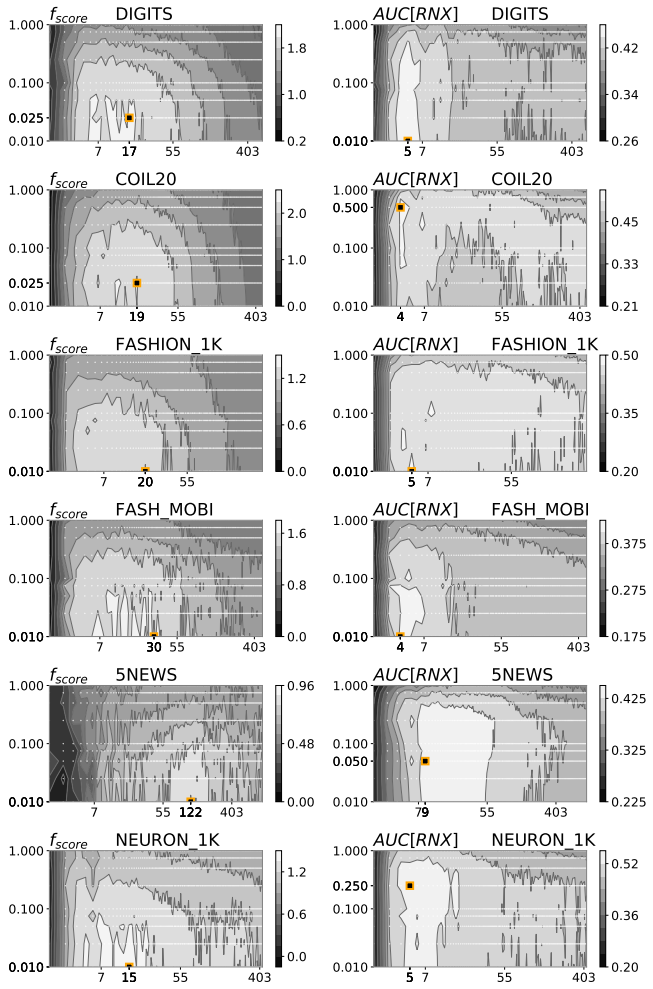



Fig. 8: Comparison of $f_{\text{score}}$ (on the left) and AUC$[R_{NX}]$ (on the right) for UMAP embeddings. The best combination of hyperparameters found by each score is located by the orange point in each dataset. In each plot, n_neighbors (on the horizontal axis) and min_dist (on the vertical axis) are shown in logarithmic scale. The light/dark region corresponds to the large/small values of the two scores.

the visualization is actually not clear. This same visualization is discouraged by $f_{\text{score}}$. The last visualization (d) belongs to the low score region in the metamap (with respect to both scores) with a too large n_neighbors and/or a too large min_dist.

It should be noted that we do not conclude which score is better than the others since each score assesses the visualization by different aspects. Indeed, among all possible visualizations of a dataset, $f_{\text{score}}$ and AUC$[R_{NX}]$ can encourage different visualizations. AUC$[R_{NX}]$ encourages the visualizations where the neighborhood is preserved. For instance, in the visualizations in Fig. 10(b), local structures, like circle patterns, can clearly be identified. In contrast, $f_{\text{score}}$ promotes visualizations where similar points are close to each other and dissimilar points are far from each other, according to the pairwise constraints. The visualization proposed by $f_{\text{score}}$ can thus give a global view of the relative relation between small clusters in visualizations like the ones in Fig. 10(a).

## VII. Bayesian Optimization for Hyperparameter Tuning with $f_{\text{score}}$

This section considers how to search through all combinations of hyperparameters to find the one with a maximum score. We propose to use Bayesian optimization (BayOpt) to solve this problem. Section VII-A introduces the advantages of this approach. Section VII-B and Section VII-C evaluate the task of tuning one hyperparameter for $t$-SNE and two hyperparameters for UMAP using the proposed $f_{\text{score}}$.

### A. Hyperparameter Tuning and Bayesian Optimization

Hyperparameters of DR methods can be tuned by trial-and-error or through a naive grid search. Better approaches exist, such as random search [47], which randomly samples combinations of hyperparameters. However, the parameter space in which the search takes place grows exponentially with respect to the number of hyperparameters.

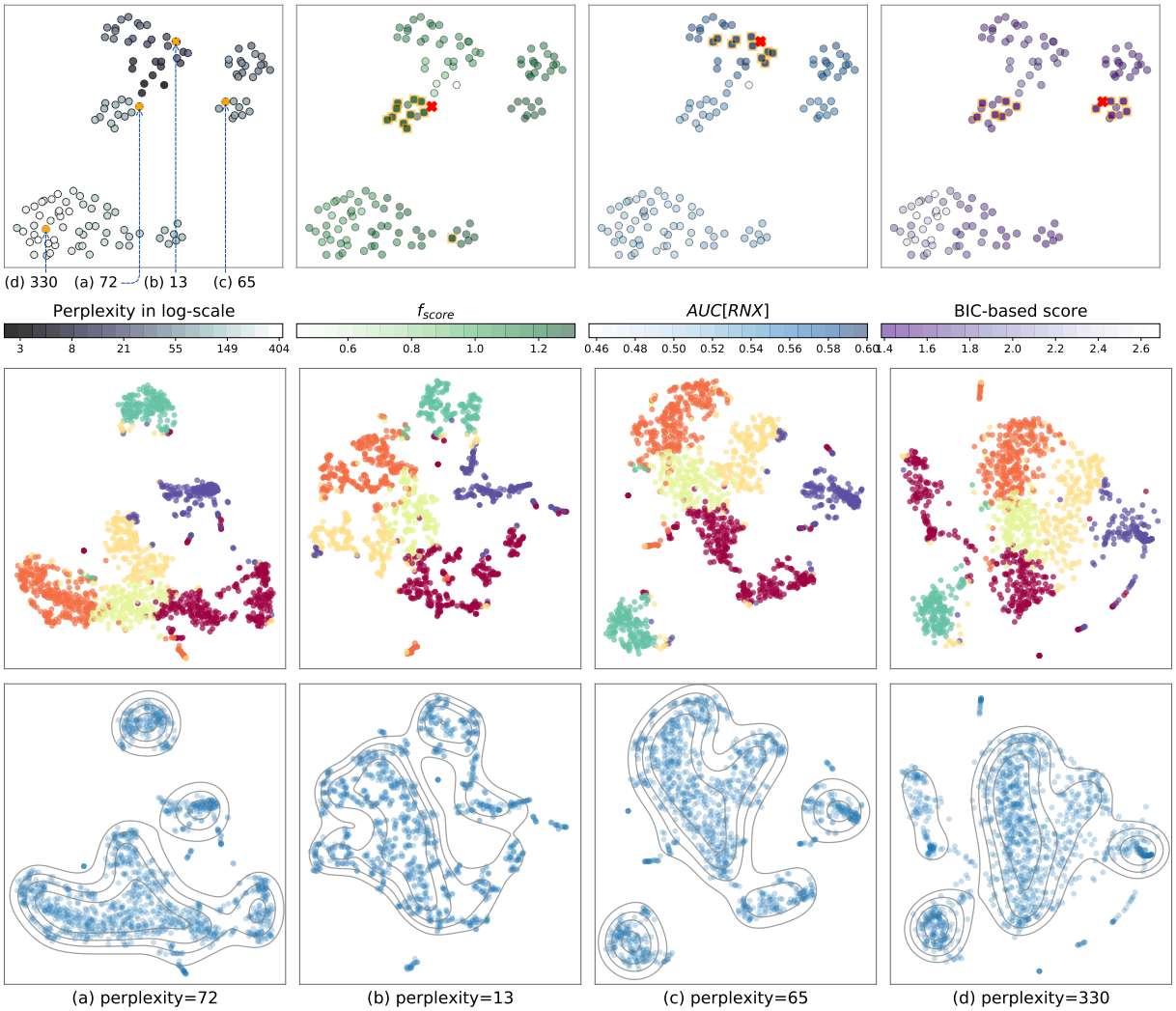



Fig. 9: Metamaps and sample visualizations for NEURON_1K. The top 5% highest scores in the metamap according to each metric are highlighted on the top row. On the middle row, the visualizations are chosen using (a) $f_{\text{score}}$, (b) AUC$[R_{NX}]$ and (c) the BIC-based score. The last one (d) is not considered good by any of the scores. On the bottom row, the same visualizations are shown without any information for supervision (i.e., no label for coloring the points). The contour in each plot shows the density estimation, which is calculated in the same way for all visualizations. Visually, several groups are correctly revealed in (a) and (c) while the whole embedding in (b) is considered as a single cluster, which makes it hard to recognize the different small groups.

Bayesian optimization (BayOpt) is a strategy for finding the extremum (minimum or maximum) of an objective function $f$ with as few evaluations as possible [8]. The objective function can be any complex non-convex black-box function that does not have a closed-form expression, or that does not have an accessible derivative. The goal of BayOpt is not to approximate this unknown function, but instead to estimate its maximum from a set of observed input samples and function values. BayOpt constructs a statistical model describing the relationship between the tuned hyperparameters and the target function. Based on past observations, BayOpt predicts the most promising hyperparameters to evaluate. As there is a trade-off between exploration and exploitation, several strategies exist to guide the optimization process to discover the parameter space: maximum probability of improvement, expected improvement and lower or upper confidence bound [48]. BayOpt successfully solves the problem of hyperparameters tuning for classification [49] or experimental design/randomized experiments [50].

In this work, the objective function to maximize under the BayOpt framework is $f_{\text{score}}$. The exploration strategy is chosen so as to explore the largest parameter space possible. The expected improvement (EI) acquisition function is thus a good choice for the surrogate function of BayOpt, as it maximizes the expected improvement over the current best parameters and has proven its efficiency in practice [49]. The parameter $\xi$ of BayOpt controls the trade-off between global search (exploration) and local optimization (exploitation). Here, $\xi$ is set to a large value (0.25) to stimulate exploration. Since there is a small variance in $f_{\text{score}}$, BayOpt takes it into account by

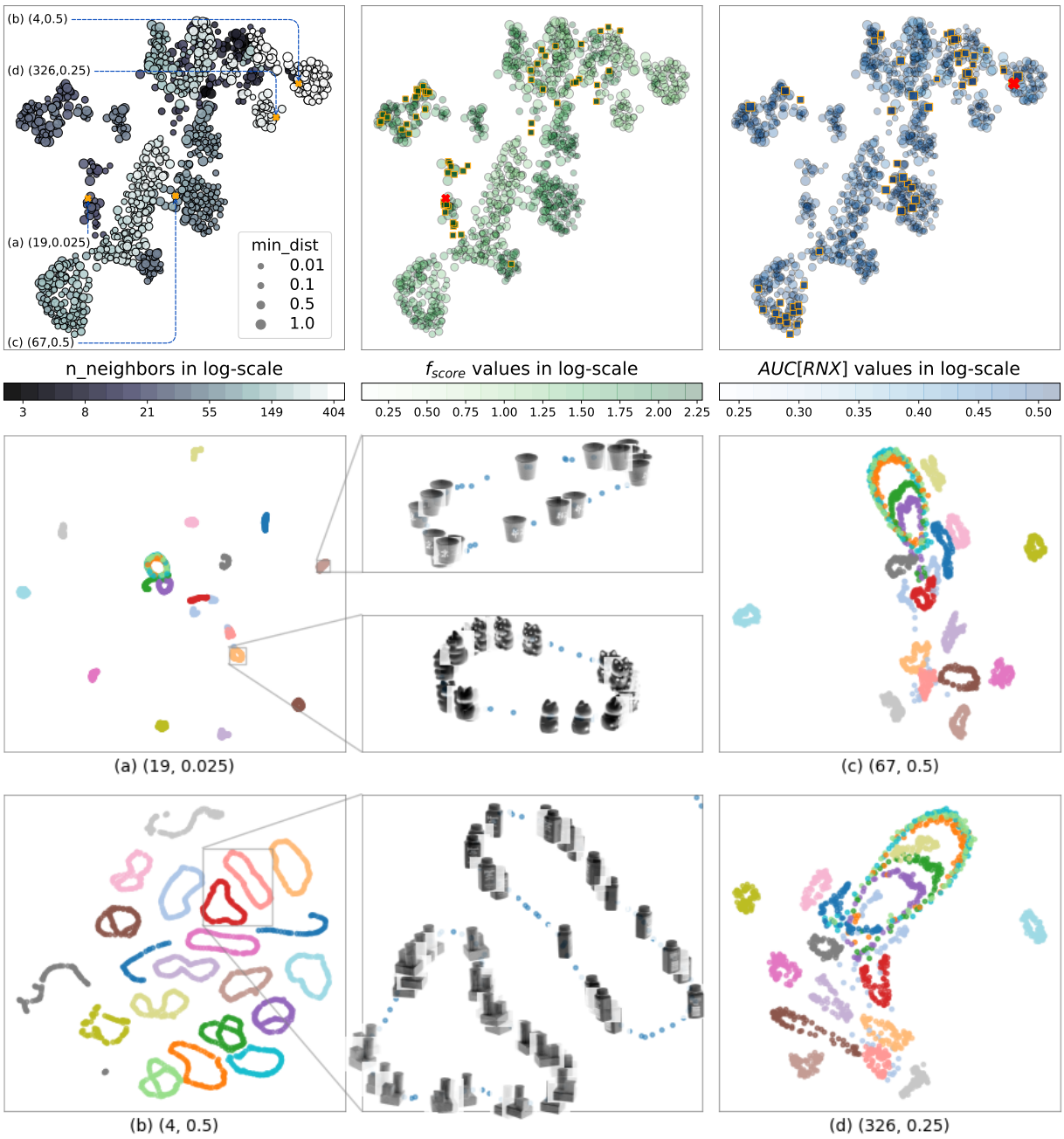



Fig. 10: Metamaps and sample visualizations for COIL20. The top 5% highest scores in the metamap according to each metric are highlighted on the top row. On the bottom row, (a) is chosen by $f_{\text{score}}$ and (b) is chosen by AUC$[R_{NX}]$. (c) is considered good by AUC$[R_{NX}]$ but not by $f_{\text{score}}$, and (d) is not considered good by any score. The detailed views when zooming in on several small groups in (a) are shown. The circle patterns are similar to the patterns in (b). However, the visualization in (a) reveals the global structure, while the one in (b) does not. When zooming in on several zones of the visualization in (b), objects in one group are closer to objects in other groups rather than to the ones in the same group.

adding small values to the diagonal of the kernel function of the underlying Gaussian process model.

### *B. Tuning One Hyperparameter for t-SNE*

Fig. 11 demonstrates how BayOpt works for tuning $t$-SNE's perplexity for all six selected datasets. Remarkably, $f_{\text{score}}$ needs to be evaluated for only 15 selected perplexities. These perplexity values are selected by BayOpt iteratively, starting with five random perplexities. The pairs of perplexity and the corresponding $f_{\text{score}}$ are used to update the BayOpt model at each iteration. The next predicted perplexity to evaluate is the most promising perplexity value that does not decrease $f_{\text{score}}$. It should be noted that BayOpt does not explicitly approximate the score function, but it tries to find the maximum value instead. BayOpt does not only find the best hyperparameter values, but also indicates the region in which it is not certain

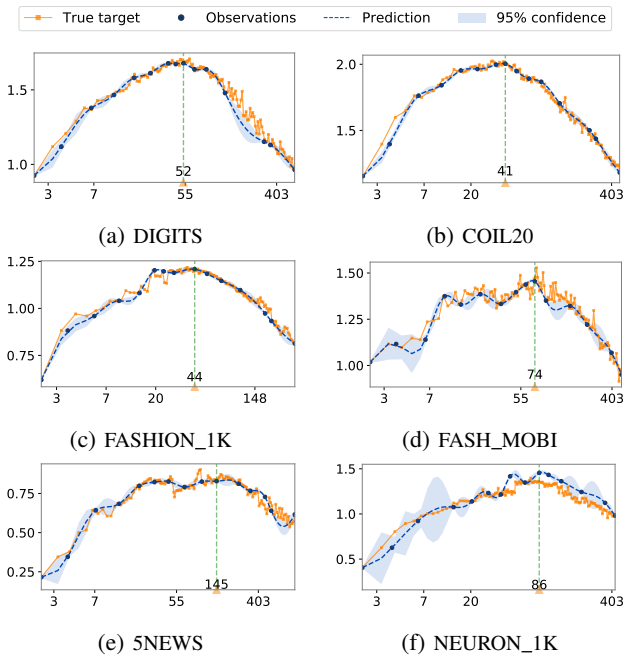



Fig. 11: Tuning $t$-SNE's perplexity for six datasets using BayOpt. $f_{\text{score}}$ is evaluated only for the embeddings of 15 selected perplexities shown by the dark blue points. The dotted blue line presents the predicted $f_{\text{score}}$ for all other perplexities. The filled blue region represents the uncertainty of the prediction. The green vertical line indicates the best predicted perplexity. The orange lines are the true values of $f_{\text{score}}$, only used as references to see how well the BayOpt prediction approximates the true target values.

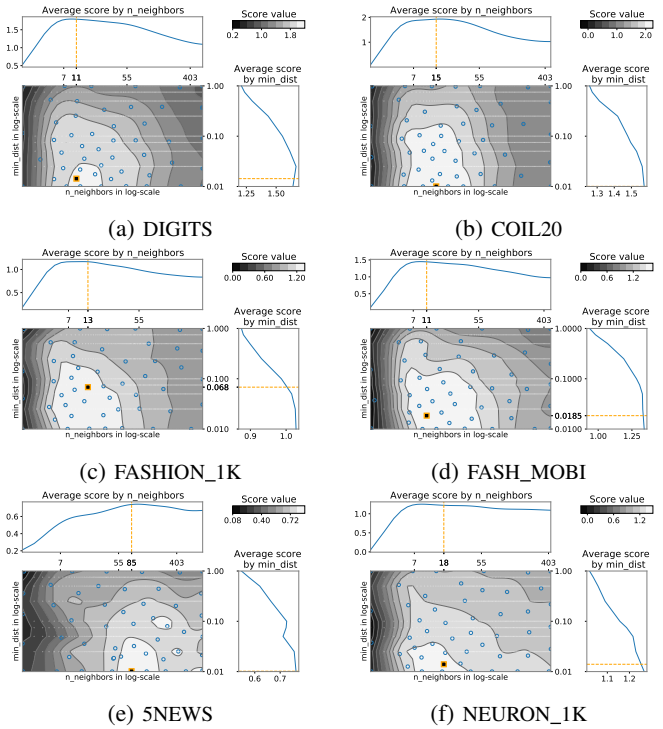



Fig. 12: Tuning two hyperparameters of UMAP using BayOpt. In each plot, 50 points (combinations of n_neighbors and min_dist) are evaluated and shown by the blue dots. The contour plots are constructed from the predicted $f_{\text{score}}$ for all other points in the grid. The light/dark region corresponds to the large/small values of $f_{\text{score}}$. The orange points indicate the best predicted hyperparameters.

about its prediction, which is usually the region of too high or too low perplexity values.

### C. Tuning Two Hyperparameters for UMAP

Tuning hyperparameters for UMAP is a more difficult task, since its two-dimensional hyperparameter grid is larger than the one-dimensional grid of $t$-SNE. Instead of evaluating thousands of combinations of values for two hyperparameters, BayOpt converges after only 50 iterations for all six experimental datasets. Fig. 12 shows how BayOpt finds the region with the best combinations for the six datasets. The uncertainty of BayOpt prediction is not shown in this plot. In comparison with the full grid used for $f_{\text{score}}$ in Fig. 8, BayOpt approximates the region of highest score more efficiently with a very limited number of evaluations.

In practice, BayOpt is used to tune multiple hyperparameters. Contour plots of every pair of hyperparameters are used to investigate the region with the best combinations. One advantage of the BayOpt approach is that it does not only maximize the target score function, but it also gives predicted scores for all hyperparameter combinations. Indeed, in each plot in Fig. 12, only 50 points are exactly evaluated. The contour is calculated upon the predicted value of the BayOpt's underlying Gaussian process model for all other points. Without spending too much resources to obtain a full grid, the estimated score given by BayOpt is reliable enough to point out the best hyperparameters.

## VIII. Conclusion and Future Work

This work tackles the problem of automatically tuning the hyperparameters of DR methods, which requires to search through all visualizations and rank them by their quality in order to find the best one. A new constraint-based score is introduced to measure the quality of visualizations by evaluating how well the information encoded in input pairwise constraints is preserved in the visualization. The proposed score, called $f_{\text{score}}$, is a simple, efficient and flexible quality metric. It does not require to calculate neighborhood information in the HD space or the expensive objective function of a non-linear DR method. Furthermore, it is complementary to other quality metrics, while being flexible (as the score can change with respect to the user's input constraints) and cheaper to compute. Based on this score, we propose to use Bayesian optimization to efficiently find the best hyperparameters instead of traditional search-based methods. The proposed workflow facilitates the use of DR methods by making the choice of difficult-to-understand hyperparameters easier and helps users to discover different visualizations with various perspectives on the structure of data.

In future work, we plan to evaluate the quality of the selected visualizations through a user-based experiment. Users

would select the instances that they consider should be in the same or in different groups in order to generate the similar and dissimilar links. Users' feedback could also be directly incorporated into the BayOpt framework to accelerate the convergence of the optimization. Another perspective is to consider richer constraints like contrastive [51] or triplet constraints [52] in order to build a more robust constraint-based quality score.

## APPENDIX A: QUALITY METRICS

Let $D^X$ and $D^Y$ be the pairwise distance matrices for all pairs of points in the HD and LD spaces, respectively. Let $D^Y_{ij}$ be the distance between two instances $i$ and $j$ in the LD space. Here are the mathematical formulas for the five selected metrics reviewed in this work.

- The Correlation Coefficient is defined as

$$\text{CC} = \text{pearson\_correlation}(D^X, D^Y) = \frac{\text{Cov}(D^X, D^Y)}{\sigma(D^X)\sigma(D^Y)}.$$

- For measuring the distance order in NMS, an isotonic transformation $D^{iso}$ is performed on $D^X$. The Kruskal's stress is then computed using the transformation

$$\text{NMS} = \sqrt{\frac{\sum_{ij}(D^{iso}_{ij} - D^Y_{ij})^2}{\sum_{ij} D^Y_{ij}}}.$$

- The Curvilinear Component Analysis Stress function is defined as

$$\text{CCA} = \sum_{ij}(D^X_{ij} - D^Y_{ij})^2 F_\lambda(D^Y_{ij}),$$

where $F_\lambda(D^Y_{ij})$ is a decreasing-weighting function of $D^Y_{ij}$. Examples of weighting functions include the step function or $1 - sigmoid(D^Y_{ij})$.

- The stress function of Sammon's Nonlinear mapping is

$$\text{NLM} = \frac{1}{\sum_{ij} D^X_{ij}} \sum_{ij} \frac{(D^X_{ij} - D^Y_{ij})^2}{D^X_{ij}}.$$

- The quality measure $AUC[R_{NX}]$ can be defined as follows. Let $k$ be the number of neighbors considered, $n$ the number of instances, $\nu^k_i$ the set of the $k$ closest neighbors of $i$ in the embedding and $\rho^k_i$ the set of the $k$ closest neighbors of $i$ in the HD space, $Q_{NX}$ is defined as

$$Q_{NX}(k) = \frac{1}{nk} \sum_{i=1}^{n} |\nu^k_i \cap \rho^k_i|,$$

$R_{NX}(k)$, the rescaled version of $Q_{NX}(k)$, is defined as

$$R_{NX}(k) = \frac{(n-1)Q_{NX}(k) - k}{n - 1 - k}$$

and $AUC[R_{NX}]$ is computed by taking the area under the $R_{NX}(k)$ curve in the log-scale of $k$

$$AUC[R_{NX}] = \left(\sum_{k=1}^{n-2} \frac{R_{NX}(k)}{k}\right) / \left(\sum_{k=1}^{n-2} \frac{1}{k}\right).$$

## REFERENCES


[1] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[2] J. Tang, J. Liu, M. Zhang, and Q. Mei, "Visualizing large-scale and high-dimensional data," in *Proc. WWW*, Montréal, Canada, Apr. 2016, pp. 287–297.

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

[4] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-sne effectively," *Distill*, vol. 1, no. 10, p. e2, 2016.

[5] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[6] G. Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[7] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen, "Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure," *Neurocomputing*, vol. 169, pp. 246–261, 2015.

[8] J. Močkus, "On bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, G. I. Marchuk, Ed., 1975, pp. 400–404.

[9] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of bayesian methods for seeking the extremum," *Towards Global Optimization*, vol. 2, pp. 117–128, 1978.

[10] F. Luo, H. Huang, Y. Duan, J. Liu, and Y. Liao, "Local geometric structure feature for dimensionality reduction of hyperspectral imagery," *Remote Sensing*, vol. 9, no. 8, p. 790, 2017.

[11] G. Shi, H. Huang, and L. Wang, "Unsupervised dimensionality reduction for hyperspectral imagery via local geometric structure feature learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 8, pp. 1425–1429, 2019.

[12] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, pp. 3221–3245, 2014.

[13] X. Geng, D.-C. Zhan, and Z.-H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 6, pp. 1098–1107, 2005.

[14] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[15] P. Demartines and J. Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 148–154, 1997.

[16] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 18, no. 5, pp. 401–409, 1969.

[17] J. A. Lee, E. Renard, G. Bernard, P. Dupont, and M. Verleysen, "Type 1 and 2 mixtures of kullback–leibler divergences as cost functions in dimensionality reduction based on similarity preservation," *Neurocomputing*, vol. 112, pp. 92–108, 2013.

[18] K. Wagstaff, C. Cardie, S. Rogers, and S. e. a. Schrödl, "Constrained k-means clustering with background knowledge," in *Proc. ICML*, Williamstown, MA, USA, Jun. 2001, pp. 577–584.

[19] I. Davidson and S. Basu, "A survey of clustering with instance level constraints," in *Proc. ACM TKDD*, 2007, pp. 1–41.

[20] S. Basu, A. Banerjee, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Proc. SIAM SDM*, Brighton, UK, Nov. 2004, pp. 333–344.

[21] I. Davidson and S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm," in *Proc. SIAM SDM*, Houston, Texas, USA, Nov. 2005, pp. 138–149.

[22] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proc. ICML*, Washington DC, USA, Aug. 2003, pp. 11–18.

[23] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Proc. NIPS*, Vancouver, Canada, Dec. 2003, pp. 521–528.

[24] M. Sugiyama, T. Idé, S. Nakajima, and J. Sese, "Semi-supervised local fisher discriminant analysis for dimensionality reduction," *Machine Learning*, vol. 78, pp. 35–61, Jan. 2008.

[25] P. Giordani and H. A. Kiers, "Principal component analysis with boundary constraints," *Journal of Chemometrics*, vol. 21, no. 12, pp. 547–556, Oct. 2007.

[26] W. Tang and S. Zhong, "Pairwise constraints-guided dimensionality reduction," in *Computational Methods of Feature Selection*. Chapman & Hall, 2007, pp. 295–312.
[27] D. Zhang, Z.-H. Zhou, and S. Chen, "Semi-supervised dimensionality reduction," in *Proc. SIAM SDM*, Minnesota, USA, Apr. 2007, pp. 629–634.
[28] I. Davidson, "Knowledge driven dimension reduction for clustering." in *Proc. IJCAI*, California, USA, Jul. 2009, pp. 1034–1039.
[29] H. Cevikalp, J. Verbeek, F. Jurie, and A. Klaser, "Semi-supervised dimensionality reduction using pairwise equivalence constraints," in *Proc. VISAPP*, Funchal, Portugal, Jan. 2008, pp. 489–496.
[30] 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*, vol. 23, pp. 241–250, 2017.
[31] A. Endert, W. Ribarsky, C. Turkay, B. Wong, I. Nabney, I. D. Blanco, and F. Rossi, "The state of the art in integrating machine learning into visual analytics," *Computer Graphics Forum*, vol. 36, no. 8, pp. 458–486, 2017.
[32] Y. Cao and L. Wang, "Automatic selection of t-SNE perplexity," in *ICML AutoML Workshop*, Sydney, Australia, Oct. 2017, pp. 1–7.
[33] M. Strickert, "No perplexity in stochastic neighbor embedding," in *Workshop New Challenges in Neural Computation*, Graz, Austria, Aug. 2012, pp. 68–115.
[34] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen, "Multiscale stochastic neighbor embedding: Towards parameter-free dimensionality reduction," in *Proc. ESANN*, Bruges, Belgium, Apr. 2014, pp. 177–182.
[35] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 2018.
[36] A. J. Ratner, C. D. Sa, S. Wu, D. Selsam, and C. Ré, "Data programming: Creating large training sets, quickly," in *Proc. NIPS*, vol. 29, Barcelona, Spain, 2016, pp. 3567–3575.
[37] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," *The VLDB Journal*, vol. 29, no. 2, pp. 709–730, 2020.
[38] C. Kaynak, "Methods of combining multiple classifiers and their applications to handwritten digit recognition," *Unpublished master's thesis, Bogazici University*, 1995.
[39] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," Tech. Rep., 1996.
[40] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.
[41] Kaggle Open Datasets. (2019) Fashion product images dataset. [Online]. Available: https://www.kaggle.com/paramaggarwal/fashion-product-images-dataset
[42] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
[43] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
[44] 10X Genomics. (2018) 1k brain cells from an e18 mouse. [Online]. Available: https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/neuron_1k_v3
[45] T. Kivioja, A. Vähärautio, K. Karlsson, M. Bonke, S. Linnarsson, and J. Taipale, "Counting absolute number of molecules using unique molecular identifiers," *Nature Proceedings*, pp. 1–18, 2011.
[46] R. Cutura, S. Holzer, M. Aupetit, and M. Sedlmair, "Viscoder: A tool for visually comparing dimensionality reduction algorithms," in *Proc. ESANN*, Bruges, Belgium, Apr. 2018, pp. 105–110.
[47] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. NIPS*, Grenada, Spain, Dec. 2011, pp. 2546–2554.
[48] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.
[49] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Proc. NIPS*, Nevada, USA, Dec. 2012, pp. 2951–2959.
[50] B. Letham, B. Karrer, G. Ottoni, E. Bakshy *et al.*, "Constrained bayesian optimization with noisy experiments," *Bayesian Analysis*, vol. 14, no. 2, pp. 495–519, 2019.
[51] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," in *Proc. ICLR*, Vancouver, Canada, May. 2018, pp. 1–16.
[52] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, Boston, USA, Jun. 2015, pp. 815–823.

**Viet Minh Vu** received the double M.S. degrees in computer science from Vietnam National University, Hanoi, Vietnam, and the University of La Rochelle, La Rochelle, France, in 2017. He is currently working toward the Ph.D. degree in computer science with the University of Namur, Namur, Belgium.

He is also working under the supervision of Professor Benoît Frénay on the subject of integrating constraints into dimensionality reduction methods for visualization.

**Adrien Bibal** received the M.S. degree in computer science and the M.A. degree in philosophy from the Université catholique de Louvain, Louvain-la-Neuve, Belgium, in 2013 and 2015, respectively, and the Ph.D. degree in machine learning, which was on the interpretability and explainability of nonlinear dimensionality reduction mappings, from the University of Namur, Namur, Belgium, in 2020.

He is currently a Postdoctoral Researcher with the University of Namur, Namur, Belgium.

**Benoît Frénay** (Member, IEEE) received the Ph.D. degree in engineering science from the Université catholique de Louvain, Louvain-la-Neuve, Belgium, in 2013.

He is currently an Associate Professor with the Université de Namur, Namur, Belgium. His research interests in machine learning include interpretability, interactive machine learning, dimensionality reduction, label noise, robust inference, and feature selection. Dr. Frénay was the recipient of the Scientific Prize IBM Belgium for Informatics in 2014 for his Ph.D. thesis on uncertainty and label noise in machine learning.