

Data Visualization with Structural Control of Global Cohort and Local Data Neighborhoods

Tingting Mu, *Member, IEEE*, John Y. Goulermas, *Senior Member, IEEE*, Sophia Ananiadou

Abstract—A typical objective of data visualization is to generate low-dimensional plots that maximally convey the information within the data. The visualization output should help the user to not only identify the local neighborhood structure of individual samples, but also obtain a global view of the relative positioning and separation between cohorts. Here, we propose a very novel visualization framework designed to satisfy these needs. By incorporating additional cohort positioning and discriminative constraints into local neighbor preservation models through the use of computed cohort prototypes, effective control over the arrangements and proximities of data cohorts can be obtained. We introduce various embedding and projection algorithms based on objective functions addressing the different visualization requirements. Their underlying models are optimized effectively using matrix manifold procedures to incorporate the problem constraints. Additionally, to facilitate large-scale applications, a matrix decomposition based model is also proposed to accelerate the computation. The improved capabilities of the new methods are demonstrated using various state-of-the-art dimensionality reduction algorithms. We present many qualitative and quantitative comparisons, on both synthetic problems and real-world tasks of complex text and image data, that show notable improvements over existing techniques.

Index Terms—Cohort visualization, cohort separability, manifold optimization, dimensionality reduction, embedding generation.

1 INTRODUCTION

Data visualization relies on the creation of effective visual representations of the given datasets, in order to facilitate the viewers' understanding of the underlying data structure based on their cognitive and perceptual skills. Amongst different data visualisation schemes, the simplest and most popular one, is plotting high-dimensional objects in lower-dimensional spaces [1], or generating low-dimensional representations of objects from their link information (usually represented as knowledge graphs) [2]. Over the past few decades, a large amount of data and signal processing techniques have been developed to generate low-dimensional data representations, and these can also be employed for the creation of meaningful plots [3], [4].

Classic dimensionality reduction approaches highlight the global data characteristics in the reduced space. For instance, principal component analysis (PCA) [5] maximizes the data variance along dominant data projections, and independent component analysis [6] maximizes the statistical independence. Canonical correlation analysis [7], which has recently been applied to learn low-dimensional representations for words [8], seeks dominant directions maximizing the correlation between two data matrices.

Subsequent work on manifold learning, such as locally linear embedding (LLE) [9], Isomap [10] and Laplacian eigenmaps (LE) [11], examine the local character of the data and rely on the observation that high-dimensional patterns most frequently lie on low-dimensional manifolds. Following this assumption, various effective strategies to capture and visualize low-dimensional manifolds have been

proposed. Examples include Riemannian manifold learning [12], adaptive manifold learning [13], and different variations of spectral embeddings working on the discriminant, ranking, semi-supervised and multi-output cases [14]–[17].

An alternative way to study the local character of data is through modeling and preserving the joint/conditional probability distribution of object pairs based on their intrinsic neighboring structures. Examples of relevant methods include stochastic neighbor embedding (SNE) [18], neighbor retrieval visualizer [19], t-distributed SNE (t-SNE) [20]. Examples of their supervised extensions include neighborhood component analysis (NCA) [21] based on linear projections, its convex version of maximally collapsing metric learning (MCML) [22], as well as the supervised variations of t-SNE [23]. Another way is through maximum variance unfolding (MVU), which maximizes the overall variance of the embedding while preserving the local distances between the neighboring samples [24], [25]. Its extension, the colored MVU (CMVU) [26], learns the low-dimensional representation from not only the local properties of the data, but also the side information, such as class labels. Other example works which combine multiple types of information characterizing the objects from different views include [27]–[29]. These encode such multi-view information in the reduced space, to highlight more reliably the local data structure and/or enhance class separabilities.

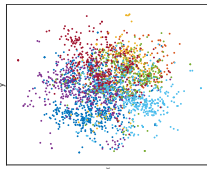
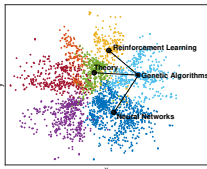
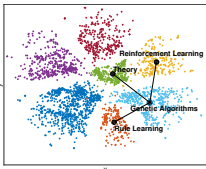
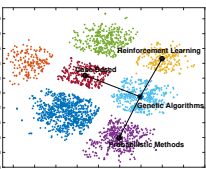
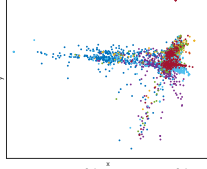
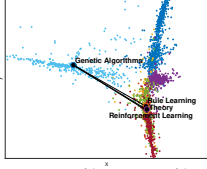
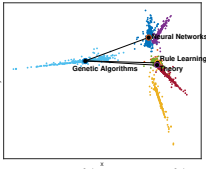
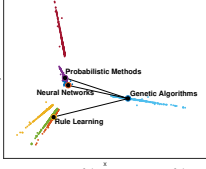
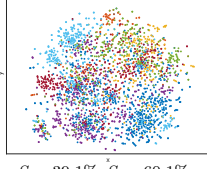
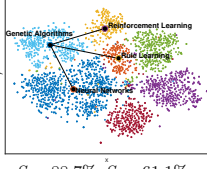
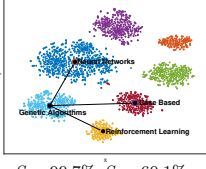
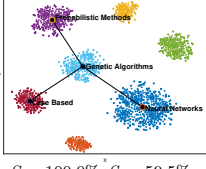
A different path for visualization is through the use of neural networks, for which recent advances in deep learning have enabled systems with deep architectures to generate low-dimensional representations of objects with improved generalization [30], [31]. Example works include those attempting to preserve the local character of the data and (or) to minimize an approximated nearest-neighbor type classification loss through a neural network based mapping function, such as the deep semi-supervised embedding [32] and

T. Mu and S. Ananiadou are with the School of Computer Science, University of Manchester, M1 7DN, UK. Email: {tingting.mu, sophia.ananiadou}@manchester.ac.uk.

J. Y. Goulermas is with the Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK. Email: j.y.goulermas@liverpool.ac.uk.

TABLE 1

Visualization of Cora publications using different techniques. For the supervised visualization, different values of the shrinking factor λ are used, and the three nearest neighbor classes of the “genetic algorithms” class are highlighted. Different classes correspond to different shadings.

Method	Unsupervised	Supervised, $\lambda = 0.7$	Supervised, $\lambda = 0.5$	Supervised, $\lambda = 0.1$
SNE	 $S_s = 41.4\%$, $S_n = 58.8\%$	 $S_s = 87.4\%$, $S_n = 59.3\%$	 $S_s = 97.5\%$, $S_n = 59.3\%$	 $S_s = 100.0\%$, $S_n = 59.1\%$
NSE	 $S_s = 36.7\%$, $S_n = 59.0\%$	 $S_s = 65.6\%$, $S_n = 57.6\%$	 $S_s = 85.4\%$, $S_n = 55.1\%$	 $S_s = 94.7\%$, $S_n = 53.1\%$
t-SNE	 $S_s = 39.1\%$, $S_n = 60.1\%$	 $S_s = 88.7\%$, $S_n = 61.1\%$	 $S_s = 99.7\%$, $S_n = 60.1\%$	 $S_s = 100.0\%$, $S_n = 59.5\%$

the deep supervised t-distributed embedding [33]. When objects are represented by a knowledge graph where link information between objects is made available, their low-dimensional representations can be learned by embedding-driven relational learning algorithms that aim at deriving object embeddings that infer link validities [2], [34].

The primary focus of this work is the visualization of high-dimensional objects. As discussed earlier, some techniques aim at highlighting the global data statistics, while some at preserving the local data character (e.g., the neighborhoods formed between objects), and/or enhancing the cohort separability (e.g., the separation achievable between different object classes). Despite the success of these methods, they do not necessarily pay attention to additional factors that can further improve and enrich the expressiveness of the visualized output.

To exemplify this, we make use of the Cora document collection [35] (described in Section 4.3), where the objective is to display the documents as two-dimensional points and demonstrate their distribution according to their word content. The classical dimensionality reduction algorithms SNE and normalized spectral embedding (NSE) [36], the popular data visualization algorithm t-SNE, as well as their supervised versions¹ are compared. Table 1 contains the visualization output of the methods and reports their intra-class neighbor preservation (S_n) and the class separation (S_s) scores (both explained in Appendix G). It can be seen that, all methods exhibit reasonably good local neighbor

1. Following the same setup as in [20], PCA firstly reduces the dimensionality to 30. Then, a Gaussian kernel is employed to convert the Euclidean distance matrix of the reduced data to a joint probability matrix \mathbf{P} . The width of the kernel is computed by setting the perplexity measure to 30, which is a smooth approximation of the effective neighbor number. NSE employs \mathbf{P} as its weight matrix from which the Laplacian is computed. The supervised versions of SNE, NSE and t-SNE are realized by following the simple linearly supervised distance transformation used in [23], which shrinks the distances between the intra-class objects by a factor of $0 < \lambda < 1$ when computing \mathbf{P} .

preservation and their supervised versions offer excellent control of class separabilities. However, the locations of the data classes in the reduced space are rather arbitrary, and as a result, the relevant positions and proximity profiles between the mapped classes do not follow any pattern. For example, the neighbor classes of the “genetic algorithms” document class varies arbitrarily across the methods.

To produce a meaningful visualization, the relative positions between object cohorts² are expected to convey important information to the viewers, instead of being recovered arbitrarily. For example, in corpus topic visualization, documents related to the “breast cancer” topic are expected to be closer to documents related to “lung cancer”, rather than the “cardiovascular” topic. Depending on the application, there often exist different types of information sources that can imply closeness relationships between cohorts, but these are frequently ignored in the process. For example, in the simplest case, the high-dimensional pattern representations themselves can be used to directly estimate similarities between object cohorts. Alternative options are to utilize some domain-specific measures and information sources external to the original data representations. With respect to the Cora corpus, for example, a citation network is available between the documents, from which the cross-citation rates between two classes of documents can be computed, and subsequently be used to control the relative positions and closeness between the classes.

One avenue for advancing data visualization and augmenting the utility of the presented information to the users, is to seek new algorithm designs with much better control over cohort arrangement. The work presented here, proceeds in this spirit and proposes various novel designs

2. Here, we extend the concept of data classes to data cohorts, to allow the inclusion of more general pattern structures, such as data clusters located by a clustering algorithm, or pattern groupings resulting from auxiliary information.

that, in addition to preserving the local data characteristics and maintaining cohort separability, they effectively control the proximity profiles between cohorts.

The principal characteristics of the presented work include the following. To achieve controllable cohort positioning and localization, we propose to use a set of cohort prototypes to underpin the access to the proximity profiles between cohorts. A set of multi-objective models are constructed by incorporating additional cohort positioning and discriminative constraints into local neighbor preservation, through the use of these prototypes. These cohort prototypes can be generated from either the input data itself or external information, according to the viewers' interests. Two types of models are developed; the embedding models that directly compute the low-dimensional representations, and the projection ones that learn projection-based mapping functions. To enforce independency between the recovered dimensions in the target space, full rank constraints are applied to the embeddings, while orthogonality constraints are applied to the projections. The resulting constrained problems are solved by applying a very effective strategy that converts the constrained optimization problem to an unconstrained one over a matrix manifold by approximating the geometric structure of that manifold. To accelerate the projection-based algorithms, an alternative projection model is also proposed by introducing an auxiliary representation of the cohort prototypes that results in a very fast one-step eigen-decomposition of a small-sized matrix. In the experiments, we examine the proposed visualization strategy using different datasets and different visualization tasks, and compare with various representative classical and state-of-the-art visualization algorithms.

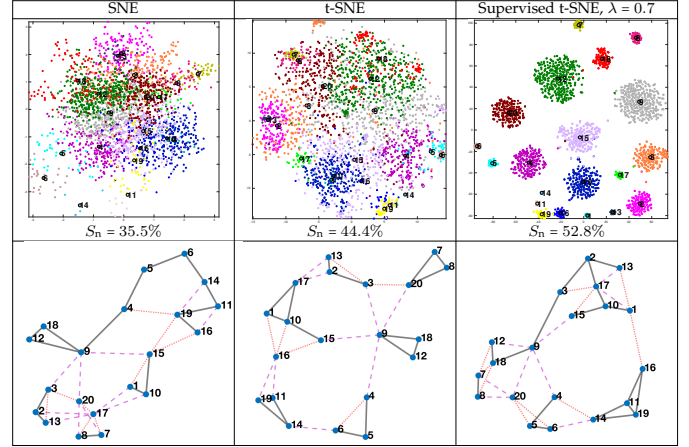
2 MOTIVATION

In the context of data visualization, there has not been a unified description of what objective a dimensionality reduction algorithm should achieve. As summarized in Section 1, the primary focus of the state of the art is to capture the local data character based on local neighbor preservation (e.g., SNE, t-SNE, LLE), or to maintain separation between classes via enhancing between-class scatter (e.g., Fisher criterion related approaches), or repositioning inter- and intra-class nearest neighbors (e.g., NCA and its equivalent convex version MCML), or to achieve both (e.g., supervised t-SNE, some deep embedding models). Although by doing so, a global view of cohort arrangement is naturally derived, the problem is that the generated cohort layouts cannot always be trusted as they can become arbitrary and unpredictable when the underlying cohort structures are complex.

We elaborate on this issue by analyzing connections between sample neighbor preservation, cohort separation and cohort localization. Preserving local neighbors of individual samples can naturally preserve in the new space those data cohorts or fractions of cohorts that contain more and stronger neighbor pairs. This can subsequently result in effective separation among cohorts with strong division. However, cohorts with weak division can be lost (e.g., by being merged with other cohorts) or broken into fractions in the new space. Alternatively, supervised algorithms attempt to preserve each data class in the new space as a whole and

TABLE 2

Illustration of twenty Cora clusters. The first row illustrates the visualized publications with different clusters highlighted by different shadings. The second row illustrates the neighbor adjacency graph between clusters. Edges in solid, dotted and dashed lines indicate true positive, false positive and false negative pairs of neighboring clusters in the new space, respectively, compared to the original space.



maintain separation between the classes. It is very important to note that neither preserving local neighbors for individual samples, nor maintaining (or enhancing) separation for cohorts can induce full control of the relevant positioning and cohort localization. This is because, given a local neighbor graph between samples and their inherent cohort division, inter-cohort neighbor links are more sparse and possess smaller similarity weights than intra-cohort links. These small and sparse weights can be effective for the algorithm to place some cohorts (or their fractions) away from each other, but not effective enough to control how far away³. Therefore, different visualization algorithms can often result in different and unpredictable cohort neighbor adjacencies.

We demonstrated in Table 1 the arbitrary positioning of the pre-defined data classes in the visualized space for various techniques using the Cora document collection. Here, we provide another demonstration in Table 2 to show the arbitrary positioning of unsupervised clusters for SNE, t-SNE and its supervised version using the same documents, which are grouped to twenty clusters using spectral clustering [37]. The two neighbor adjacency graphs between the clusters, constructed in the original and visualized spaces, are compared in the second row of Table 2, while its first row illustrates the embedded documents. The edges are constructed by identifying two effective neighbors for each targeted cohort. The incorrect neighbor links in the visualized space are highlighted in dashed and dotted lines.

To further investigate the changes in local neighborhoods, cohort separation and positioning between the new and original spaces, Table 3 provides another example using different 3-dimensional patterns divided into groups of similar sizes. The 2-dimensional embeddings are computed by SNE, t-SNE and LLE with the effective neighbor number (or its approximation by perplexity) set to 30. All the algorithms perform well in terms of local neighbor preservation

3. The algorithm can compromise the between-cohort proximities (or proximities between data patches from different cohorts) to give priority to follow more accurately the stronger and denser links (with higher weights) between neighbor samples from the same cohort.

TABLE 3
Visualization of 3D data points in 2D space using unsupervised techniques. Data partitions are highlighted using different shadings.

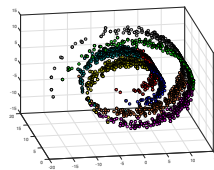
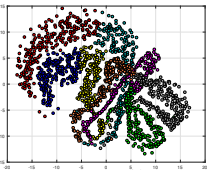
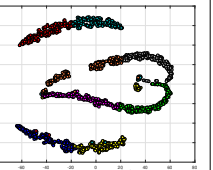
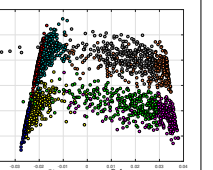
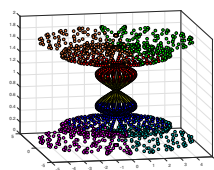
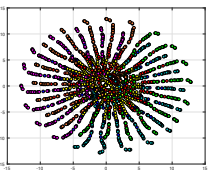
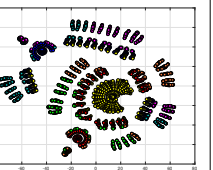
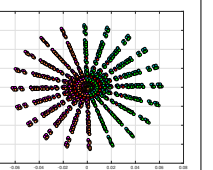
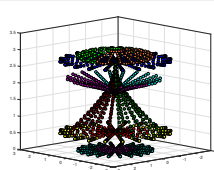
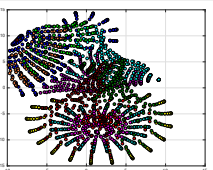
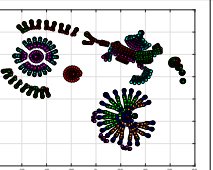
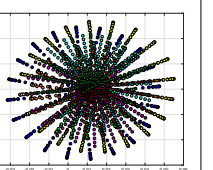
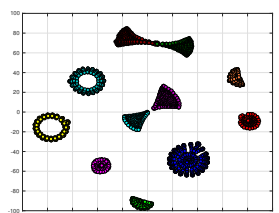
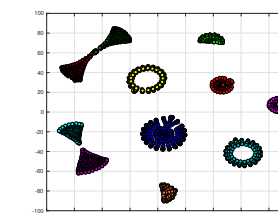
Datasets	Original	SNE	t-SNE	LLE
Swiss roll		 $S_n = 84.3\%$	 $S_n = 86.5\%$	 $S_n = 42.0\%$
Cylinder1		 $S_n = 79.8\%$	 $S_n = 74.9\%$	 $S_n = 94.0\%$
Cylinder2		 $S_n = 77.4\%$	 $S_n = 79.4\%$	 $S_n = 87.6\%$

TABLE 4
Visualization of 3D data points in 2D space using supervised techniques. Data partitions are highlighted using different shadings.

Supervised SNE, $\lambda = 0.1$	Supervised t-SNE, $\lambda = 0.1$
 $S_n = 87.8\%$	 $S_n = 87.7\%$

as indicated by the high neighbor preservation score S_n . For the simpler Swiss roll data, group separation can be well maintained and some algorithms can even match the original group positioning, e.g., LLE. However, for the more complex Cylinder datasets, none of the algorithms in Table 3 is capable of maintaining both good group separation and reasonable group positioning. By generating visualizations with the supervised versions of SNE and t-SNE in Table 4 for Cylinder2, group separation does improve, but group positioning is again arbitrary. As demonstrated via Tables 1-4, we see that existing works can produce undesirable cohort layouts. This is because they prioritize local neighbor preservation and cohort separation, and these can lead to compromised cohort positioning due to the sparse and weak inter-cohort links. So far, there is no existing work studying how to produce a visualization output with control in simultaneously and effectively capturing all three aspects. To improve and enrich the expressiveness of the visualized output, we will define a multi-objective visualization and propose a set of strategies, referred to as cohort visualization with cohort arrangement control (COVA), by incorporating additional cohort positioning and discriminative constraints that enhance local neighbor preservation through the use of computed cohort prototypes.

3 PROPOSED METHODS

The notation for the data input is as follows. We are given a set of data samples denoted by $\mathcal{D} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^l$ with $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T$ being the i th sample, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l]^T$ the entire $l \times d$ feature matrix, and \mathbb{R}^k the reduced (target or visualization) space of typical dimensionality $k = 2$ or 3 . The integer vector $\mathbf{y} = [y_1, y_2, \dots, y_l]^T$ is used to store the information of cohort memberships, with $y_i \in \{1, 2, \dots, c\}$ indicating the cohort index of the i th sample, and c the total number of existing cohorts. An equivalent cohort membership representation is the $l \times c$ binary matrix $\mathbf{Y} = [y_{ij}]$ with each element y_{ij} set to one if and only if the i th sample belongs to the j th cohort. Finally, the sample cardinality of i th cohort is denoted by n_i .

Regarding data cohorts, one type could be the assigned clusters computed from the feature matrix \mathbf{X} using a clustering algorithm [38]. The objective would be to visually display the cluster structures to the users [39]. Another type of data cohorts could be the predefined data classes relevant to a classification problem [21], [40]. Sometimes, the cohort label information can be partially available. In this case, membership of the unlabeled samples can be predicted from the labeled ones by employing a simple classifier, e.g., k -nearest neighbor (KNN), where the labeled samples are used for training.

3.1 COVA Embeddings

The low-dimensional representations or embedding $\{\mathbf{z}_i\}_{i=1}^l$ in the visualization space \mathbb{R}^k (or equivalently the $l \times k$ embedding matrix $\mathbf{Z} = [z_{ij}]$) are to be computed for all d -dimensional data samples \mathbf{x}_i . A visualization model that preserves in the target space the local neighboring pattern between the samples typically relies on the $l \times l$ similarity weight matrix $\mathbf{W} = [w_{ij}]$ that controls the neighboring adjacency structure of the underlying similarity graph of the samples $\{\mathbf{x}_i\}_{i=1}^l$. It can be sparsely constructed using a

nearest neighbor search, and the actual weights w_{ij} can be computed using any predefined similarity measure, such as the Gaussian kernel, cosine similarity, etc. Neighbor preservation through \mathbf{W} can be achieved by minimizing either an accumulated weighted error [37], according to

$$\min_{\{z_i\}_{i=1}^l} O_{\text{dist}}^{(L)} = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l w_{ij} \|z_i - z_j\|_2^2, \quad (1)$$

or an accumulated Kullback-Leibler (KL) divergence based on Student t-distribution [20], as

$$\min_{\{z_i\}_{i=1}^l} O_{\text{KL}}^{(L)} = \sum_{i=1}^l \sum_{j \neq i} p_s(w_{ij}) \log \frac{p_s(w_{ij})}{p_s\left((1 + \|z_i - z_j\|_2^2)^{-1}\right)}, \quad (2)$$

where $\|\cdot\|_2$ denotes the L_2 -norm. The normalization function $p_s(\cdot)$ is set to convert the input similarity weight to either a joint probability $p(i, j)$ by $p_s(w_{ij}) = \frac{w_{ij}}{\sum_{s=1}^l \sum_{t \neq s} w_{st}}$, or to a conditional probability $p(i|j)$ by $p_s(w_{ij}) = \frac{w_{ij}}{\sum_{s=1}^l w_{sj}}$. The same normalization is applied to the estimated similarity weights. Both formulations attempt to transmit the neighborhood information contained in \mathbf{W} to the embedding space reflected by the Euclidean distances between samples. Since the similarity weight between a sample and itself does not contribute to the neighboring patterns based on the above formulations, the diagonal elements of the matrix \mathbf{W} are set to zero. By encoding the cohort label information \mathbf{Y} within the weight matrix \mathbf{W} , e.g., by shrinking the inter-cohort similarity using the factor $0 < \lambda < 1$ [23], such that

$$\mathbf{W} \leftarrow \mathbf{W} \circ (\mathbf{Y}\mathbf{Y}^T) + \lambda \mathbf{W} \circ (1 - \mathbf{Y}\mathbf{Y}^T), \quad (3)$$

where \circ denotes the Hadamard multiplication, the separation between cohorts can be enhanced.

As discussed in Sections 1 and 2, however, the cooperation between Eqs.(1,2) and Eq.(3) does not impose effective control over the proximity profile between the data cohorts in the target space. Therefore, we propose to restrict the cohort distribution of the samples mapped in \mathbb{R}^k . An effective way to achieve this, is to construct a set of k -dimensional vectors $\{\mathbf{v}_i\}_{i=1}^c$. Specifically, for each i th data cohort, $\mathbf{v}_i = [v_{i1}, v_{i2}, \dots, v_{ik}]^T$ is defined in the target space and referred to as a *cohort prototype*. These prototypes are constructed in a way to encode information reflecting the underlying closeness between the cohorts, and are gathered in a $c \times k$ prototype matrix $\mathbf{Y} = [\mathbf{v}_i]$. We then use each prototype to tie the data samples with their corresponding cohort to control the relevant positioning of cohorts, and simultaneously, to boost the inter-cohort separabilities. The benefit of this strategy, is that it effectively delegates the comparatively complex control of groups of samples to the simpler problem of controlling a small set of low-dimensional prototypes.

3.1.1 Cohort Prototype Generation

We formulate the cohort prototype vectors $\{\mathbf{v}_i\}_{i=1}^c$ such that their pairwise distances reflect the relevance and proximities between their corresponding cohorts. Assuming a $c \times c$ dissimilarity matrix $\mathbf{D} = [d_{ij}]$ (similarities can be converted to dissimilarities) is constructed for the c cohorts, it can be subjected to a reconstruction technique to obtain the c cohort prototypes that encapsulate most of the relevant

information content contained within \mathbf{D} . Specifically, multidimensional scaling [41], [42] can be employed to obtain the sought \mathbf{Y} by taking \mathbf{D} as the input. Focusing on the neighboring patterns between cohorts, stochastic, manifold or ordinal embedding techniques can be used to preserve the cohort neighborhood structure of \mathbf{D} (Appendix H.7 contains an empirical comparison of such methods). The key question that remains, is how to capture the proximity information between cohorts in order to construct \mathbf{D} .

The most straightforward way is to compute d_{ij} directly from the original data \mathbf{X} , so that the generated prototypes reflect the inherent cohort proximities in the original space. This can either be based on sample pair relations

$$d_{ij} = \frac{1}{n_i n_j} \sum_{y_s=i, y_t=j} d(\mathbf{x}_s, \mathbf{x}_t), \quad (4)$$

or on relations between the cohort centers, such as

$$d_{ij} = d\left(\frac{1}{n_i} \sum_{y_s=i} \mathbf{x}_s, \frac{1}{n_j} \sum_{y_t=j} \mathbf{x}_t\right). \quad (5)$$

The dissimilarity measure $d(\cdot, \cdot)$ between two patterns, can be the Euclidean distance, or a dissimilarity quantity converted from the cosine similarity, a Gaussian or polynomial kernel, etc. Another way to compute d_{ij} is based on the weight matrix \mathbf{W} generated for local neighbor preservation. The averaged similarity is used to obtain d_{ij} as

$$d_{ij} \leftarrow \frac{1}{n_i n_j} \sum_{y_s=i, y_t=j} w_{st}, \quad (6)$$

which reflects density of neighbor links between samples from two cohorts.

More generally, the users are free to define their own cohort proximity structure according to the application at hand, which may not necessarily be under the control of the original data \mathbf{X} . Some more direct domain-specific measures and information resources external to the original data can be used. For example, when visualizing articles based on their word content (e.g., using a bag-of-words model to construct \mathbf{X}), proximity between article cohorts can be computed externally based on their cross-citation rates. Letting $\Theta = [\theta_{ij}]$ denote a binary citation matrix between articles, where each binary weight θ_{ij} indicates whether one article is cited by the other, one possible cohort similarity measure can be based on the density of citation links between two article cohorts, e.g., computed by $\frac{0.5}{n_i + n_j} \sum_{y_p=i} \sum_{y_q=j} \theta_{pq} + \theta_{qp}$. Another example is that, when visualizing images based on their pixel content (e.g., using a convolutional neural network (CNN) to construct \mathbf{X}), proximity between image cohorts can be computed externally by examining the image captions. For instance, by treating the collection of image captions from each cohort as a document and modeling the documents using a bag-of-words model, proximity between two cohorts can be computed as the cosine similarity between their two document vectors.

3.1.2 Cohort Arrangement Control

Since the prototypes communicate the cohort geometry, we control the relevant positioning and localization of the cohorts by co-locating the data samples together with the cohort prototypes $\{\mathbf{v}_i\}_{i=1}^c$ in the joint visualization space

\mathbb{R}^k . To enable data samples mapped in \mathbb{R}^k to be distributed proximately to the particular prototype that represents the cohort of the sample, a sum of penalized distances between prototypes and associated embedded samples can be minimized, according to

$$\min_{\{\mathbf{z}_i\}_{i=1}^l} O_{\text{dist}}^{(C)} = \sum_{i=1}^l \sum_{j=1}^c r_{ij} \|\mathbf{z}_i - \mathbf{v}_j\|_2^2, \quad (7)$$

where r_{ij} quantizes the degree of confidence that the i th sample belongs to the j th cohort, stored in the $l \times c$ matrix $\mathbf{R} = [r_{ij}]$. Alternatively, an accumulated KL divergence can be minimized, given as

$$\min_{\{\mathbf{z}_i\}_{i=1}^l} O_{\text{KL}}^{(C)} = \sum_{i=1}^l \sum_{j=1}^c \frac{r_{ij}}{\sum_{s=1}^l \sum_{t=1}^c r_{st}} \log \frac{\frac{r_{ij}}{\sum_{s=1}^l \sum_{t=1}^c r_{st}}}{(1 + \|\mathbf{z}_i - \mathbf{v}_j\|_2^2)^{-1}}. \quad (8)$$

As in Eq.(2), the Student's t-distribution with a single degree of freedom, e.g., $(1 + \|\mathbf{z}_s - \mathbf{v}_j\|_2^2)^{-1}$, is used to compute the confidence degree based on the Euclidean distance between a sample and the cohort prototype in the target space. Compared to the Gaussian function $\exp(-\|\mathbf{z}_s - \mathbf{v}_j\|_2^2)$, it has longer tails and it drops less rapidly as the distance increases, leading thus, to a reduced sensitivity to scale changes for points far apart [20].

The template confidence r_{ij} to be matched in the target space can be computed by examining the proximity between the i th sample and the centroid of the j th cohort in the original space, such that

$$r_{ij} = r\left(\mathbf{x}_i, \frac{1}{n_j} \sum_{y_s=j} \mathbf{x}_s\right), \quad (9)$$

where $r(\cdot, \cdot)$ is set as a similarity measure. It can also be computed as the averaged adjacency weights in \mathbf{W} between the i th sample and samples from the j th cohort, given as

$$r_{ij} = \begin{cases} \frac{1}{n_j - 1} \sum_{s \neq i, y_s=j} w_{is}, & \text{if } y_i = j, \\ \frac{1}{n_j} \sum_{y_s=j} w_{is}, & \text{otherwise.} \end{cases} \quad (10)$$

The above formulations of r_{ij} push data samples that are closer to a cohort center in the original space to stay closer to its corresponding cohort prototype in the target space. To enhance cohort separation, the links between a data sample and prototypes can be adjusted based on the following rescaling process

$$\mathbf{R} \leftarrow \begin{cases} \mathcal{L}(\mathbf{R} \circ \mathbf{Y}, [1 - \lambda_p, 1]) + \mathcal{L}(\mathbf{R} \circ (1 - \mathbf{Y}), [0, \lambda_p]), & \text{if } \lambda_p \neq 0, \\ \mathbf{R} \circ \mathbf{Y}, & \text{if } \lambda_p = 0, \end{cases} \quad (11)$$

where $0 \leq \lambda_p \ll 1$ and $\mathcal{L}(\cdot, [a, b])$ is a mapping function that linearly rescales elements in the input matrix to a given interval $[a, b]$. When $\lambda_p = 0$, cohort separation is maximally enforced by simply cutting the links between a data sample and its irrelevant prototypes by fixing r_{ij} to zero when $y_i \neq j$. An alternative for computing the confidence degree is to formulate r_{ij} as a probability, according to

$$r_{ij} = \begin{cases} p(s_{ij}), & \text{if } y_i = j, \\ \lambda_p, & \text{otherwise.} \end{cases} \quad (12)$$

Between samples and their irrelevant cohorts, r_{ij} is set as a small value $0 \leq \lambda_p \ll 1$ to equally push the samples to stay away. Between a sample and its own cohort, r_{ij} is

obtained using a probability function $p(\cdot)$ estimated from the similarity set $\{s_{tj}\}_{t=1}^{n_j}$ with s_{tj} computed by Eq.(9) or Eq.(10). The normal distribution can be assumed, with mean and standard deviation directly estimated from $\{s_{tj}\}_{t=1}^{n_j}$. When the sample size n_j is small, a Student's t-distribution can be used to obtain $p(s_{ij})$. More complex alternatives, such as mixture models or kernel density estimators can also be used to estimate $p(s_{ij})$. Compared to Eqs.(9,10), Eq.(12) changes the driving force for drawing samples to cohort prototypes. It drives data samples distributed in a denser area within a cohort to move closer to the cohort prototypes.

As a result of minimizing the sum of either the penalized distance errors in Eq.(7) or the KL divergence in Eq.(8), the data samples would form clusters in \mathbb{R}^k around the locations indicated by the cohort prototypes determined by matrix \mathbf{Y} . Cohort separation is at the same time realized by setting weak links between the data samples and the prototypes of the other cohorts. The use of small positive values for r_{ij} when $y_i \neq j$, e.g. being controlled by $\lambda_p = 0.1$, is usually favored over zeros for regularization purposes, which is particularly effective when KL divergence is used to formulate the objective.

3.1.3 Bi-objective COVA Model Construction

The different objectives correspond to different intuitive rules between data patterns and cohort prototypes. Eqs.(1,2) correspond to the rule that enables the original samples and their embedded counterparts in the target space to have accordant (dis)similarity characteristics (*rule 1*). And Eqs.(7,8) correspond to the rule of enabling the distances between data samples and prototypes in \mathbb{R}^k to reflect the membership associations between samples and cohorts (*rule 2*). In order to simultaneously control cohort arrangement, maintain cohort separations and preserve the local data geometry, the objectives in Eqs.(7,8) and Eqs.(1,2) can be combined to a multiobjective formulation. To maintain a comparable scale for distances between samples and prototypes and to avoid inflation/deflation of the objectives caused by the number of distances examined, the weights w_{ij} and r_{ij} are normalized to $\frac{w_{ij}}{\sum_{s=1}^l \sum_{t=1}^c w_{st}}$ and $\frac{r_{ij}}{\sum_{s=1}^l \sum_{t=1}^c r_{st}}$, respectively, before being used; this normalizes \mathbf{W} and \mathbf{R} to unity sum. The minimizing objective can, for example, be then formulated as $\alpha O_{\text{dist}}^{(C)} + (1 - \alpha) O_{\text{dist}}^{(L)}$, where the parameter $\alpha \in [0, 1]$ balances the two rules.

We now make several comments on the rule-specific objectives. The satisfaction of the two rules can be achieved with any of the four possible combinations of the two pairs of objectives (or of ones based on alternative rule-specific objectives constructed from similar arguments). This is advocated by the fact that the constituent objectives act differently in enforcing the rules. For instance, $O_{\text{KL}}^{(C)}$ and $O_{\text{KL}}^{(L)}$ have softer error penalization properties than the squared errors in $O_{\text{dist}}^{(C)}$ and $O_{\text{dist}}^{(L)}$. Concerning the control of cohort arrangement only, an extreme solution from minimizing only $O_{\text{dist}}^{(C)}$, will force all the samples to be mapped to the prototype of their cohort, that is $\mathbf{z}_i = \mathbf{v}_{y_i}, \forall i$. On the other hand, minimizing only $O_{\text{KL}}^{(C)}$, will make $\|\mathbf{z}_i - \mathbf{v}_j\|_2$ equal or very close to a positive value determined by r_{ij} . The former solution does not distinguish between intra-cohort samples, while the latter attempts to form certain intra-

cohort distributions. This shows that $O_{\text{dist}}^{(C)}$ is more forceful than $O_{\text{KL}}^{(C)}$ at highlighting cohort separability and creates tighter cohorts. Another observation, is that local neighbor preservation acts as a regularizer for cohort control, in the sense that it protects from cohort over-concentration. Finally, $O_{\text{dist}}^{(L)}$ cannot be minimized on its own without incorporating length constraints over z_i , because of the trivial solutions. However, combining with either objective $O_{\text{dist}}^{(C)}$ or $O_{\text{KL}}^{(C)}$ makes this constraint unnecessary.

3.1.4 Model Optimization

The aforementioned bi-objective optimization can generate the embedded patterns $\{z_i\}_{i=1}^l$. However, in order to obtain independent coordinates in the k different axes, we require \mathbf{Z} to have linearly independent columns, by solving

$$\min_{\mathbf{Z} \in \mathbb{R}^{l \times k}, \text{rank}(\mathbf{Z})=k} \alpha O^{(C)} + (1 - \alpha) O^{(L)}, \quad (13)$$

where $O^{(C)}$ and $O^{(L)}$ are any of the previously defined rule-specific objectives. The feasible set in Eq.(13) includes all the full-rank $l \times k$ matrices (we assume $l > k$), and it is an open submanifold of the vector space $\mathbb{R}^{l \times k}$, referred to as the noncompact Stiefel manifold $\mathbb{R}_*^{l \times k}$. In Appendix A we summarize the four rule-specific objective functions $O_{\text{dist}}^{(C)}(\mathbf{Z})$, $O_{\text{dist}}^{(L)}(\mathbf{Z})$, $O_{\text{KL}}^{(C)}(\mathbf{Z})$ and $O_{\text{KL}}^{(L)}(\mathbf{Z})$ in matrix form and include the calculations of their Euclidean gradients.

The model based on Eqs.(1,7) has the analytical solution

$$\mathbf{Z} = [\alpha \mathbf{D}(\mathbf{R}) + (1 - \alpha) \mathbf{L}(\mathbf{W}) + \zeta \mathbf{I}_{l \times l}]^{-1} \mathbf{R} \Upsilon, \quad (14)$$

where $\mathbf{I}_{l \times l}$ is the $l \times l$ identity matrix, and the regularization parameter $0 < \zeta \ll 1$ is used to handle matrix singularity. We explain how to derive this solution in Appendix D. For the other three objective term combinations, it is not possible to obtain analytical solutions due to the involvement of Eqs.(2,8). Here, we show how to seek solutions using iterative optimization relying on the Euclidean gradient.

Specifically, the constrained optimization can be converted to an unconstrained one in a smooth search space by working with certain manifold defined constructs, such as representations of the tangent space, retraction mapping and Riemannian gradient, facilitated by the rich geometric structure of a manifold [43]. Given an objective function $f(x)$ imposed over the manifold M , the tangent space, denoted by the composite notation $T_x M$, provides a local vector space approximation of the manifold M at $x \in M$, in which the tangent vector is defined as a generalized notion of the directional derivative. A retraction mapping $R_x M : T_x M \rightarrow M$ is used to map updates in the tangent space at $x \in M$ onto the manifold. The composite notation $R_x M$ is used to denote the retraction mapping function, with R representing retraction, x indicating the point location where the mapping is computed, and M the corresponding manifold name. It can be viewed as moving in the direction of a tangent vector whilst remaining on the manifold. The Riemannian gradient represents the first order information of the cost function on the manifold, and is denoted by $\text{grad} f(x)$. We denote the Euclidean gradient of the cost function as $\text{Grad} f(x)$, to be distinguished from the Riemannian. Finally, to solve the optimization problem in the form of $\min_{x \in M} f(x)$, many unconstrained optimization

approaches, such as gradient descent or Newton methods, can be adapted to operate over the manifold. Taking the gradient descent, as an example, the $(i+1)$ th iteration update can be implemented as

$$x_{i+1} = R_{x_i} M(-\gamma \text{grad} f(x_i)), \quad (15)$$

where the scalar $\gamma > 0$ controls the step size. The retraction mapping $R_{x_i} M$ is computed at the current solution x_i obtained in the i th iteration with an update determined by $-\gamma \text{grad} f(x_i)$.

For the current problem, we work on the noncompact Stiefel manifold using gradient descent, and this requires to compute the Riemannian gradient and to determine the retraction mapping. It can be shown (based on proposition 2.1 in [44]) that the tangent space $T_{\mathbf{Z}} M$ of the manifold $M = \{\mathbf{Z} \mid \mathbf{Z} \in \mathbb{R}^{l \times k}, \text{rank}(\mathbf{Z}) = k\}$ at $\mathbf{Z} \in M$, is actually the entire vector space of $l \times k$ matrices (henceforth, we replace the general notation x for a point on the manifold with the sought embedding \mathbf{Z}). Thus, given an arbitrary matrix $\xi \in \mathbb{R}^{l \times k}$, its orthogonal projection onto $T_{\mathbf{Z}} M$ is itself; this is denoted by $P_{\mathbf{Z}}(\xi) = \xi$. Also, because the noncompact Stiefel manifold is an embedded manifold of the vector space, the Riemannian gradient of the cost function is the orthogonal projection of its Euclidean gradient onto the tangent space at the input matrix [43]. This gives $\text{grad} f(\mathbf{Z}) = P_{\mathbf{Z}}(\text{Grad} f(\mathbf{Z})) = \text{Grad} f(\mathbf{Z})$.

Computation of the retraction mapping $R_{\mathbf{Z}_i} M(\xi)$ for the noncompact Stiefel manifold $\mathbb{R}_*^{l \times k}$ can be equivalent to the process of first moving away from \mathbf{Z}_i along the direction of ξ , which is set as $\xi = -\gamma \text{grad} f(x_i)$ for gradient descent, to get to the new point $\mathbf{Z}_i + \xi$, and then projecting $\mathbf{Z}_i + \xi$ back to the manifold. To set $R_{\mathbf{Z}_i} M(\xi)$, we employ a projection based strategy [44], [45]. When $\mathbf{Z}_i + \xi$ is a full-rank matrix, it is already on the manifold, and hence, there is no need for further processing; that is $R_{\mathbf{Z}_i} M(\xi) = \mathbf{Z}_i + \xi$. Subsequently, the gradient descent update of the embedding matrix becomes $\mathbf{Z}_{i+1} = \mathbf{Z}_i - \gamma \text{grad} f(x_i)$, which resembles the standard gradient descent for unconstrained optimization. When $\text{rank}(\mathbf{Z}_i + \xi) < k$, we project $\mathbf{Z}_i + \xi$ back to the manifold at \mathbf{Z}_i , which is equivalent to seeking a full-rank matrix that is close to $\mathbf{Z}_i + \xi$. To achieve this, a very small value ϵ can be added to the zero singular values of $\mathbf{Z}_i + \xi$. These modified singular values along with the singular vector matrices of $\mathbf{Z}_i + \xi$ are used to reconstruct a full-rank matrix close to $\mathbf{Z}_i + \xi$ to be the output of the retraction mapping.

Apart from the constraint of linearly independent columns admitting a structure of the noncompact Stiefel manifold, alternative constraints for optimizing \mathbf{Z} , could be the orthogonal feasible set $M_1 = \{\mathbf{Z} \mid \mathbf{Z} \in \mathbb{R}^{l \times k}, \mathbf{Z}^T \mathbf{Z} = \mathbf{I}_{k \times k}\}$ [37], or a relaxed orthogonal one $M_2 = \{\mathbf{Z} \mid \mathbf{Z} \in \mathbb{R}^{l \times k}, \mathbf{Z}^T \mathbf{B} \mathbf{Z} = \mathbf{I}_{k \times k}\}$ [46], where the diagonal matrix \mathbf{B} stores the scaling parameters. The first set M_1 admits a structure of the Stiefel manifold and M_2 the generalized Stiefel manifold, of which the geometries are discussed in [43], [47]. Compared to M and M_2 , M_1 is more restrictive, has lower expressive power and may create somewhat peculiar embedding distributions. Compared to M , M_2 requires extra effort to tune the parameter matrix \mathbf{B} and performance improvement is not guaranteed. Thus, in this work we choose the rank constraint to compute the

COVA embeddings. An example is provided in Appendix C to illustrate the effects of the feasible sets M , M_1 and M_2 .

3.2 COVA Projections

It is often required to have a mapping $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^k$, so that the original dataset is processed in an out-of-sample extension fashion. This is needed, for example, when dealing with incremental applications that respond to data growth, in which instead of constantly recalculating the embedding of the expanded set $[\mathbf{X}_{\text{old}}^T, \mathbf{X}_{\text{new}}^T]^T$, ϕ can directly compute⁴ the embedded versions for the newly added samples \mathbf{X}_{new} . We describe such a transformation for mapping data samples. It relies on mapping a d -dimensional sample \mathbf{x} to a (dis)similarity space, where every dimension constitutes a relational measurement $r(\cdot, \cdot)$ between \mathbf{x} and a prototype sample. Given a set of $\tilde{k} \leq l$ prototype samples $\{\mathbf{q}_i\}_{i=1}^{\tilde{k}}$ selected from $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^l$, a \tilde{k} -dimensional (dis)similarity space can be induced by the mapping $\phi_r : \mathbb{R}^d \rightarrow \mathbb{R}^{\tilde{k}}$, defined as $\phi_r(\mathbf{x}) = [r(\mathbf{x}, \mathbf{q}_1), r(\mathbf{x}, \mathbf{q}_2), \dots, r(\mathbf{x}, \mathbf{q}_{\tilde{k}})]^T$. The mappings for the entire \mathcal{D} can be collected in the $l \times \tilde{k}$ relation feature matrix $\mathbf{F} = [\phi_r(\mathbf{x}_1), \dots, \phi_r(\mathbf{x}_l)]^T$. This, in turn, can support the introduction of a $k \times k$ projection matrix $\mathbf{U} = [u_{ij}]$, which realizes the desired mapping ϕ as $\phi(\mathbf{x}) = \mathbf{U}^T \phi_r(\mathbf{x})$. The embedded patterns for all the elements in \mathcal{D} can then be obtained through the simple projection $\mathbf{Z} = \mathbf{F}\mathbf{U}$ ⁵.

3.2.1 Optimizing COVA Projections

Proceeding with solving the multiobjective optimization of Section 3.1.3, we replace \mathbf{Z} with $\mathbf{F}\mathbf{U}$ and seek to identify the projection matrix \mathbf{U} . Similar to previous works [48], [49] for projection-based embeddings, we enforce orthogonality of the projection matrix. This requires us to optimize over the feasible set $M = \{\mathbf{U} \mid \mathbf{U} \in \mathbb{R}^{\tilde{k} \times k}, \mathbf{U}^T \mathbf{U} = \mathbf{I}_{k \times k}\}$, which corresponds to the Stiefel manifold. We derive and summarize the Euclidean gradients with respect to \mathbf{U} for the contributing objectives in Appendix A. In the projection case, it is not possible to derive analytical solutions for all the possible objective combinations (see Appendix D), and instead we perform iterative optimization over the Stiefel manifold.

4. When required, both old and new embedded patterns can be optimized together, with $[\mathbf{Z}_{\text{old}}^T, \phi(\mathbf{X}_{\text{new}})^T]^T$ employed as the initializer, where ϕ is applied row-wise to its data matrix input.

5. Compared to embedding models, projection ones learn parametric mapping functions with fixed formulations to connect the original and new spaces. Therefore, they offer less freedom in generating the embedded patterns than the embedding models which directly learn the embedding coordinates. To enhance the cohort control of a projection model, one effective way is compact information in \mathbf{X} based on the cohorts. For instance, a similarity matrix \mathbf{S} can be computed between data samples and a set of representative samples from different cohorts. By applying an h -nearest neighbor search to \mathbf{S} , the averaged similarity value between a sample and its neighbor representatives from the i th cohort can be used as its i th compacted feature. This results in an $l \times c$ cohort-based feature matrix $\mathbf{X}_c = \tilde{\mathbf{S}}\mathbf{Y}_p\mathbf{M}^{-1}$, where $\tilde{\mathbf{S}} = \mathbf{S} \circ \mathbf{N}(\mathbf{S}, h)$ and each element of the matrix $\mathbf{N}(\mathbf{S}, h) = [\delta_{ij}]$ is a binary neighborhood indicator, set to one if sample \mathbf{x}_i is in the h -nearest neighbors (based on \mathbf{S}) of the j th representative or vice-versa. \mathbf{M} is a $c \times c$ diagonal matrix storing in its diagonal the number of sample prototypes from each class, and \mathbf{Y}_p denotes the cohort label matrix of the representative samples. Subsequently, another level of nonlinear transformation ϕ_r is applied to \mathbf{X}_c to obtain \mathbf{F} and then the embedded pattern is obtained by $\mathbf{F}\mathbf{U}$.

Given an arbitrary matrix $\xi \in \mathbb{R}^{\tilde{k} \times k}$, its projection onto the tangent space $T_{\mathbf{U}}M$ of the Stiefel manifold at $\mathbf{U} \in M$ is

$$P_{\mathbf{U}}(\xi) = (\mathbf{I}_{\tilde{k} \times \tilde{k}} - \mathbf{U}\mathbf{U}^T)\xi + \mathbf{U} \text{skew}(\mathbf{U}^T \xi), \quad (16)$$

where $\text{skew}(\mathbf{X}) = \frac{\mathbf{X} - \mathbf{X}^T}{2}$. Because the Stiefel manifold is an embedded manifold of the vector space, the Riemannian gradient is the orthogonal projection of its Euclidean gradient onto the tangent space, which gives $\text{grad}f(\mathbf{U}) = P_{\mathbf{U}}(\text{Grad}f(\mathbf{U}))$. Given the tangent vector $\xi_{\mathbf{U}} \in T_{\mathbf{U}}M$, the retraction mapping can either be defined through QR decomposition, such that

$$R_{\mathbf{U}}M(\xi_{\mathbf{U}}) = \text{qf}(\mathbf{U} + \xi_{\mathbf{U}}), \quad (17)$$

where $\text{qf}(\cdot)$ denotes the Q factor of the input matrix, or through polar decomposition, such that

$$R_{\mathbf{U}}M(\xi_{\mathbf{U}}) = (\mathbf{U} + \xi_{\mathbf{U}}) (\mathbf{I}_{k \times k} + \xi_{\mathbf{U}}^T \xi_{\mathbf{U}})^{\frac{1}{2}}. \quad (18)$$

A detailed description of the Stiefel manifold and its properties, as well as the derivations of Eqs.(16,17,18) are provided in Appendix B. For the application of gradient descent, the projection matrix can be updated according to

$$\mathbf{U}_{i+1} = R_{\mathbf{U}_i}M(-\gamma P_{\mathbf{U}_i}(\text{Grad}f(\mathbf{U}_i))), \quad (19)$$

where $f(\cdot)$ is the bi-objective term $\alpha O^{(C)} + (1 - \alpha)O^{(L)}$.

3.2.2 Eigen-COVA Model

All the above projection models are optimized iteratively. In this section, we propose a very efficient projection model with analytical solution, which relies primarily on the two distance-based objectives $O_{\text{dist}}^{(C)}$ and $O_{\text{dist}}^{(L)}$.

We firstly create a projection to map the cohort prototypes \mathbf{v}_i to some auxiliary representations \mathbf{e}_i in \mathbb{R}^k , that have to preserve accurately the prototype distribution in Υ . To do this, we define a mapping $\psi : \mathbb{R}^k \rightarrow \mathbb{R}^k$ as

$$\mathbf{e}_i = \psi(\mathbf{v}_i) = \mathbf{V}^T \mathbf{v}_i, \text{ for } i = 1, \dots, c, \quad (20)$$

where $\mathbf{V} = [v_{ij}]$ is the $k \times k$ projection matrix. If the embedded prototypes $\{\mathbf{e}_i\}_{i=1}^c$ are collected within a $c \times k$ embedding matrix $\mathbf{E} = [e_{ij}]$, then the overall projection can be expressed as $\mathbf{E} = \Upsilon\mathbf{V}$. Using this, we can substitute the prototypes \mathbf{v}_i in $O_{\text{dist}}^{(C)}$ of Eq.(7) with their auxiliary representations, and the modified objective becomes

$$O_{\text{dist}}^{(C)} = \text{tr} \left(\begin{bmatrix} \mathbf{Z} \\ \mathbf{E} \end{bmatrix}^T \mathbf{L} \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0}_{c \times c} \end{bmatrix} \begin{bmatrix} \mathbf{Z} \\ \mathbf{E} \end{bmatrix} \right), \quad (21)$$

and needs to be optimized over both \mathbf{Z} and \mathbf{E} .

As mentioned above, we must require that $\{\mathbf{e}_i\}_{i=1}^c$ preserve the original cohort proximity information. The introduction of the new mapping necessitates the use of a third objective, enforced by maximizing the following distance-based score

$$O_{\text{aux}} = \frac{1}{2} \sum_{i=1}^c \sum_{j=1}^c \|\mathbf{v}_i - \mathbf{v}_j\|_2 \|\mathbf{e}_i - \mathbf{e}_j\|_2^2 = \text{tr}(\mathbf{E}^T \mathbf{L}(\mathbf{W}_v) \mathbf{E}), \quad (22)$$

where $\mathbf{W}_v = [w_{ij}^{(v)}]$ is a $c \times c$ weight matrix, having elements $w_{ij}^{(v)} = \|\mathbf{v}_i - \mathbf{v}_j\|_2$. Its maximization implies, that the further apart two prototypes \mathbf{v}_i and \mathbf{v}_j are, the more distant their representations \mathbf{e}_i and \mathbf{e}_j should be forced to be by the

stronger weighting term $\|\mathbf{v}_i - \mathbf{v}_j\|_2$. In cooperation with $O_{\text{dist}}^{(C)}$, the maximization of Eq.(22) additionally manages to regulate cohort closeness and between-cohort sample scatter.

For the eigen-COVA model, we optimize a multiobjective function expressed as

$$\alpha\eta O_{\text{dist}}^{(C)} + (1 - \alpha)O_{\text{dist}}^{(L)} - \alpha(1 - \eta)O_{\text{aux}}, \quad (23)$$

where the parameters $0 < \alpha, \eta < 1$ regulate the strengths of the constituent rules. Specifically, α controls the trade-off between the local neighbor preservation and the global cohort control, whereas η controls the trade-off between cohort proximity and sample separation. It is now possible, after incorporating the projections $\mathbf{Z} = \mathbf{FU}$ and $\mathbf{E} = \mathbf{YV}$, to rewrite this optimization more compactly as

$$\min_{\mathbf{P} \in \mathbb{R}^{(\tilde{k}+k) \times k}} f(\mathbf{P}) = \text{tr}(\mathbf{P}^T \mathbf{O}_{\text{eig}} \mathbf{P}), \quad (24)$$

subject to scale and orthogonality constraints, with the matrix $\mathbf{P} = [\mathbf{U}^T, \mathbf{V}^T]^T$ containing the two sought projection matrices. The symmetric $(\tilde{k} + k) \times (\tilde{k} + k)$ matrix

$$\mathbf{O}_{\text{eig}} = \mathbf{O}_1 + \alpha\eta \mathbf{T}^T \mathbf{L} \left(\begin{bmatrix} \mathbf{0}_{l \times l} & \mathbf{R} \\ \mathbf{W}_v \mathbf{R}^T & \mathbf{0}_{c \times c} \end{bmatrix} \right) \mathbf{T} \quad (25)$$

aggregates the entire problem information and the two user-defined weights α and η , where $\mathbf{L}(\cdot)$ denotes the Laplacian matrix of a square matrix input. The definition of \mathbf{O}_{eig} depends on the $(l + c) \times (\tilde{k} + k)$ matrix $\mathbf{T} = \begin{bmatrix} \mathbf{F} & \mathbf{0}_{l \times k} \\ \mathbf{0}_{c \times \tilde{k}} & \mathbf{Y} \end{bmatrix}$, and the symmetric $(\tilde{k} + k) \times (\tilde{k} + k)$ matrix $\mathbf{O}_1 = \begin{bmatrix} (1 - \alpha)\mathbf{F}^T \mathbf{L}(\mathbf{W})\mathbf{F} & \mathbf{0}_{\tilde{k} \times \tilde{k}} \\ \mathbf{0}_{\tilde{k} \times \tilde{k}} & -\alpha(1 - \eta)\mathbf{Y}^T \mathbf{L}(\mathbf{W}_v)\mathbf{Y} \end{bmatrix}$.

The solution of Eq.(24), accompanied by the constraint $\mathbf{P}^T \mathbf{P} = \mathbf{I}_{k \times k}$, is straightforward through the eigenvectors of \mathbf{O}_{eig} that correspond to the smallest k eigenvalues. The first \tilde{k} rows of \mathbf{P} store the sample projections \mathbf{U} , while the remaining rows the auxiliary projections \mathbf{V} of the cohort prototypes. The existence of multiple projections is warranted by the symmetry of \mathbf{O}_{eig} , which makes \mathbf{P} orthogonal. Nevertheless, these eigenvectors cannot give orthogonal \mathbf{U} and \mathbf{V} , since $\mathbf{P}^T \mathbf{P} = \mathbf{U}^T \mathbf{U} + \mathbf{V}^T \mathbf{V} = \mathbf{I}_{k \times k}$. Therefore, we employ an orthogonalization procedure typically used in sequential projection pursuit algorithms [50], to compute the solution in k steps. In each step, only one eigenvector of \mathbf{O}_{eig} corresponding to the smallest eigenvalue is needed (this can be done using, for example, the inverse power method). In each subsequent step, the contribution of the previously found projection is removed from the original data samples and prototypes. Specifically, if for the t th step, we denote by $\mathbf{F}^{(t)}$ and $\mathbf{Y}^{(t)}$ the matrices of the data samples and prototypes, respectively, we can use the update rule $\mathbf{F}^{(t+1)} = \mathbf{F}^{(t)} \left(\mathbf{I}_{\tilde{k} \times \tilde{k}} - \frac{\mathbf{u}^{(t)} \mathbf{u}^{(t)T}}{\mathbf{u}^{(t)T} \mathbf{u}^{(t)}} \right)$ and $\mathbf{C}^{(t+1)} = \mathbf{C}^{(t)} \left(\mathbf{I}_{k \times k} - \frac{\mathbf{v}^{(t)} \mathbf{v}^{(t)T}}{\mathbf{v}^{(t)T} \mathbf{v}^{(t)}} \right)$ with the initializations $\mathbf{F}^{(1)} = \mathbf{F}$ and $\mathbf{C}^{(1)} = \mathbf{C}$ from the original problem data. This procedure projects all the samples on the orthogonal complement of the k -dimensional subspace of $\mathbb{R}^{\tilde{k}}$ spanned by the basis $\{\mathbf{u}^{(t)}\}_{t=1}^k$, and similarly for the prototypes. In this way, the geometric characteristics of the embedded samples and prototypes associated with each dimension, can potentially capture separate aspects of

the data that enhance the exploratory value of the resulting visualization. The optimization involves k updates of the projection vectors. Each update relies on the computation of the $(\tilde{k} + k) \times (\tilde{k} + k)$ matrix \mathbf{O}_{eig} and of a single eigenvector.

3.3 Model Implementation Details

The specific COVA embedding models (Section 3.1) are referred to as COVA-E1 (using $O_{\text{dist}}^{(C)}, O_{\text{dist}}^{(L)}$), COVA-E2 (using $O_{\text{KL}}^{(C)}, O_{\text{KL}}^{(L)}$), COVA-E3 (using $O_{\text{dist}}^{(C)}, O_{\text{KL}}^{(L)}$) and COVA-E4 (using $O_{\text{KL}}^{(C)}, O_{\text{dist}}^{(L)}$), where the employed objectives are the ones which correspond to the $O^{(C)}$ and $O^{(L)}$ terms in Eq.(13). Similarly, for the COVA projection models (Section 3.2.1), we define the corresponding COVA-P1, COVA-P2, COVA-P3 and COVA-P4 models. The COVA-E1 and eigen-COVA models possess analytical solutions, while the other versions (E2-4 and P1-4) are optimized iteratively using gradient descent with standard line search. Compared to existing visualization techniques that usually possess computational complexity quadratic to the sample size [20], the increased complexity of COVA is only linear to the sample size. A detailed complexity analysis is provided in Appendix E.

To visualize large datasets, [20] proposes an effective strategy by displaying only a random subset of the samples but utilizing information gathered from the entire dataset, which we adopt for large-scale COVA visualizations. For P1-P4, their gradient computation is more expensive compared to the embedding models. In order to speed them up, we apply stochastic gradient descent through gradient estimations from a sample subset [21], instead of using the entire dataset. Parallel and multi-core executions could also be used to facilitate further accelerations. When optimizing E2-4, a rank examination of the updated matrix $\mathbf{Z}_i + \xi$, whose size is $l \times k$, is performed in each iteration, and this introduces an extra cost of $O(lk^2)$. Given its small column size ($k = 2, 3$), it is unlikely that a rank deficient embedding matrix is obtained in every iteration. Thus, we perform retraction mapping every a fixed number (e.g., $N_c = 10$) of iterations instead of every single iteration, leading to reduced computational cost (here, N_c is referred to as the rank checking condition number). To facilitate practitioners in the field, we present pseudo-code of the proposed algorithms together with initialization guidelines in Appendix F.

4 EXPERIMENTAL ANALYSIS AND RESULTS

We compare the different versions of COVA with various popular and state-of-the-art visualization and representation learning methods. These include the unsupervised embedding methods LLE [9], SNE, t-SNE, local linear coordination (LLC) [51], Isomap [10] and the unsupervised embeddings computed from the Laplacian matrix defined in local discriminant models and global integration (LDMGI) [16], the semi-supervised deep embedding [32], the supervised t-SNE embedding (S-t-SNE) [23], the supervised projection methods linear discriminant analysis (LDA) [52], NCA [21] and MCML [22], as well as the recently developed multi-modal manifold analysis (MMA) [29]⁶. Results evaluation

6. MMA represents a multi-view strategy relying on mixing multiple types of data proximities, and we use it here as a supervised embedding tool to mix the neighboring and class-based proximities.

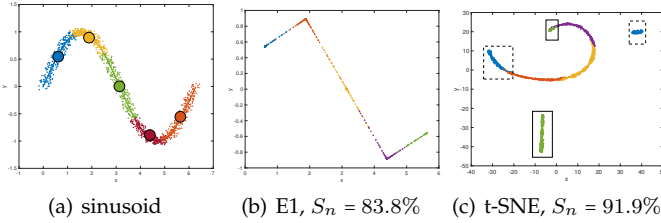


Fig. 1. Illustration of COVA-E1 and t-SNE embeddings for the 2D synthetic Sinusoid data. The used cohort prototypes for COVA-E1 are marked in (a) as solid circles.

is based on qualitative visual comparisons and quantitative analyses based on cohort separation (S_s), cohort positioning (S_c or $S_c^{(r)}$) and sample neighbor preservation (S_n) scores as described in Appendix G. To compute S_n , \bar{n} is set to the 90% of the size of the smallest cohort. For illustration purposes, we restrict the visualizations to $k = 2$ dimensions.

4.1 Demonstration with Synthetic Data

First, we provide a simple demonstration using 2D synthetic data containing the sinusoidal shape shown in Fig.1(a). We employ 2D data as it is straightforward to perform a visual comparison of the original and reorganized cohort positionings in the two corresponding spaces of equal dimensionality. The mean sample from each cohort is used as its cohort prototype (marked in Fig.1(a) with a solid circle); further implementation details are provided in Appendix H.1. The visualization output of COVA-E1 with $\alpha = 0.1$ is displayed in Fig.1(b). It can be seen that COVA can effectively control the cohort locations through the use of predefined prototypes. The output from applying t-SNE to the same data is displayed in Fig.1(c). Although it offers a good S_n score, the visualized cohort structure is different from the original one, where two cohorts split into fragments as indicated by the solid and dashed boxes of Fig.1(c).

Tables 3, 4 show that embedding methods, such as LLE, SNE and t-SNE (and their supervised versions), do not reflect well the original cohort structure for some example 3D synthetic datasets. Here, we use COVA to visualize the Cylinder2 dataset as shown in the last row of Table 3. To generate 2D cohort prototypes from the 3D input data points, t-SNE is applied to a Gaussian similarity matrix computed between the cohort centers. These prototypes, which are displayed in Fig.2(a), reveal intrinsic patterns of the original cohort arrangements and guide COVA to produce cohort structure closer to the original structure. Example output of E2 and E3 with $\alpha = 0.6$ is displayed in Figs.2(b),2(c), with the output of other COVA models shown in Appendix H.1 (see Fig.H.4).

As evidenced by Fig.1(c) and Tables 3,4, preserving only individual sample neighbors in the embedded space does not guarantee a reliable global arrangement of data cohorts. Compared to LLE, SNE and t-SNE that are specialized at neighbor preservation, COVA can sometimes violate certain sample neighbor links, but gains control over the global cohort structure (see Figs.1,2). By doing so, the benefit is that the distances between visualized cohorts start to carry meaningful information rather than being arbitrary, and the overall expressiveness of the visualized output is improved.

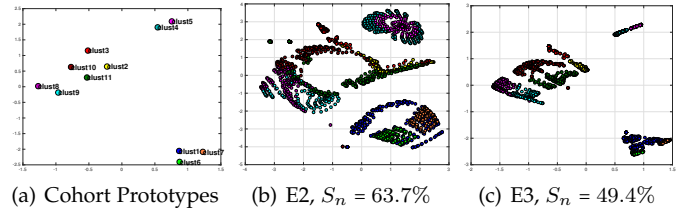


Fig. 2. Illustration of COVA-E2 and COVA-E3 embeddings computed for Cylinder2 data. (a) The computed prototypes.

Appendix H.1 displays additional results obtained with different settings of cohort prototypes and another 2D synthetic data, as well as comparative analysis between different versions of COVA models (see Tables H.1,2) and analysis of COVA parameters (see Figs.H.2,3).

4.2 Scene Image Visualization

We process 36,500 scene images from the Places2 image database [53], belonging to 365 unique scene classes, each containing 100 images. Every image is represented by 4,096 features extracted by a convolutional neural network (CNN). A $36,500 \times 36,500$ local neighbor adjacency matrix \mathbf{W} is constructed by identifying 30 effective neighbors for each image based on Gaussian similarity, where the Gaussian kernel width is uniquely decided for each image based on a fixed perplexity of 30 by following the same approach as in [20]. Each scene class is treated as a cohort, and the cohort membership matrix \mathbf{R} is computed by Eqs.(10,11).

In the following experiments, images from targeted scene classes are visualized. Cohort prototypes are computed directly from the input data to display the scene class structure as it exists in the original space. The cohort distance matrix \mathbf{D} is computed from \mathbf{W} via Eq.(6), based on which t-SNE is used to compute the feature vectors of the cohort prototypes. The relative positioning of these prototypes reflects the local neighboring relations between scene classes, and $S_c^{(r)}$ is used to evaluate the preservation of such class neighbor structure in the new space.

4.2.1 Embedding based Visualization

Images from 30 randomly selected classes are displayed with COVA weight set to $\alpha = 0.6$. Class neighbor preservation is illustrated in Fig.3, where the computed cohort prototypes preserve 60.4% of the original class neighbor links, while COVA embeddings preserve over 90% of the neighbor links indicated by the prototypes, and t-SNE only preserves 30.8% of the original class neighbor links.

Figs.3(e),3(f) display the different class arrangements from COVA-E2 and t-SNE. The prototypes approximate the original link structure⁷ and enable COVA to offer class arrangement much closer to the original space than t-SNE.

We further investigate how cohort prototypes influence COVA by replacing a random subset of the generated prototypes with random patterns. The change of COVA

7. When aggressively reducing feature dimension from 4,096 to 2 having a high number of classes, it is challenging to generate 2D prototypes to perfectly reflect all the original neighbor links between classes. Depending on the complexity of the class structure, cohort prototypes approximate such structure with varying accuracy level.

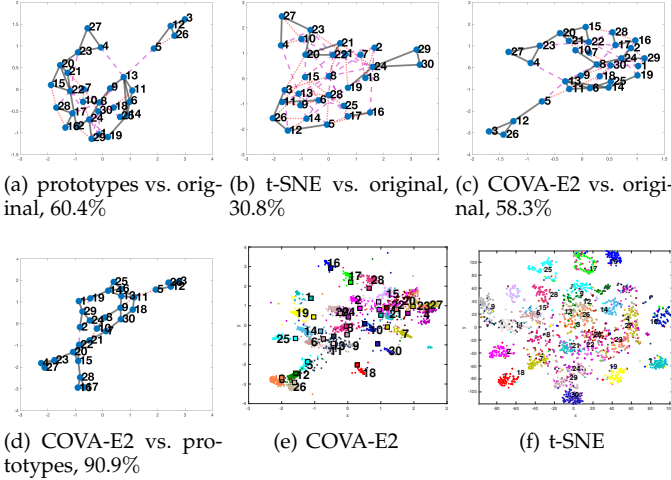


Fig. 3. (a)-(d): Illustration of class neighbor preservation (two effective neighbors are identified). For each method pair “X vs. Y”, edges in solid, dotted and dashed lines indicate true positive, false positive and false negative neighbor links of X compared to Y. Link preservation accuracies are shown in percentages. (e): COVA-E2 output, where the prototypes are shown as “□”. (f): t-SNE output. Different cohorts are numbered and correspond to different shadings.

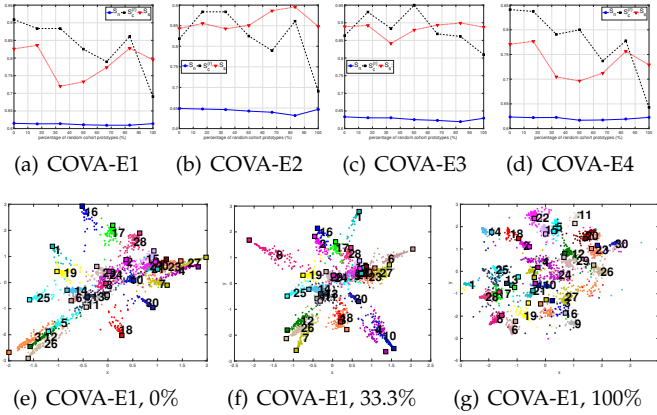


Fig. 4. (a)-(d): Performance change for COVA embedding models for varying percentages of random prototypes using Places2 images. (e)-(g): Visualization output of COVA-E1 generated by using 0%, 33.3% and 100% random prototypes, where the prototypes are shown as “□”. Different cohorts are numbered and correspond to different shadings.

performance versus varying percentages (from 0% to 100%) of random prototypes is shown in Figs.4(a)-4(d). The embedded images of COVA-E1 together with the used cohort prototypes are illustrated in Figs.4(e)-4(g). The figures show a matching arrangement between the cohorts and their corresponding prototypes, indicating robust cohort positioning control of COVA. When all the prototypes are generated randomly, there is a drop in the $S_c^{(r)}$ score. Cohort separation is related to the distribution of the used prototypes. Therefore, S_s varies without a fixed pattern which matches the random characteristic of those replaced prototypes. Figs.4(a)-4(d) show that, no matter how many random prototypes are included, local neighbor preservation performance for individual samples is stable. In general, cohort prototypes affect the global arrangement of cohorts more than the local arrangement of individual samples as indicated by the varying $S_c^{(r)}$ and S_s scores.

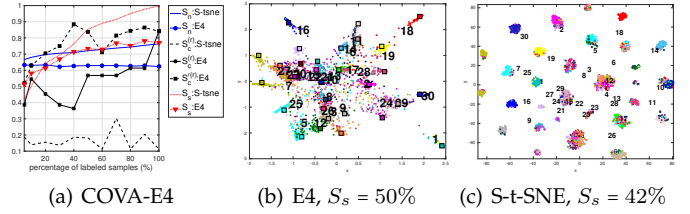


Fig. 5. (a): Performance change for COVA-E4 and S-t-SNE for varying percentages of labeled samples using Places2 images. (b): COVA-E4 output, where the prototypes are shown as “□”. (c): S-t-SNE output. Both (b) and (c) are generated with 5% images labeled.

Another experiment is conducted to study how COVA performs when only part of the samples are labeled. In each trial, n_l out of 100 images from each class are treated as labeled samples, where $n_l \in \{5, 10 : 10 : 90\}$. Class membership of the unlabeled samples is predicted by a KNN classifier trained with the labeled ones. Fig.5(a) compares the performance changes against varying number of labeled images for COVA-E4 and S-t-SNE ($\lambda = 0.7$), and results of other COVA embedding models are shown in Appendix H.2 (see Fig.H.5). In addition to $S_c^{(r)}$ computed by comparing with the gold standard cohort prototypes generated using all the images from each class based on their ground truth labels, $S_c^{(r)}$ is computed by comparing with the input prototypes generated using only the n_l labeled images for each class. In general, COVA is able to form desired cohort arrangement matching the given cohort prototypes indicated by reasonably good $S_c^{(r)}$. However, the cohort separation and positioning information can become less reliable when fewer labeled samples are available, and this results in lower S_s and $S_c^{(r)}$. S-t-SNE is very strong at enhancing cohort separation, indicated by its high S_s score computed using all the input labels (including the predicted ones for the unlabeled images). However, it can result in overfitting when there are few labeled samples available, e.g., being informed by the wrongly predicted labels. This is exemplified by Figs.5(b),5(c), where S-t-SNE generates cohorts with more mixed shadings, indicating more misplaced images from other classes, than COVA, given 5% labeled images.

In previous experiments, the COVA weight is fixed to $\alpha = 0.6$. We investigate how α affects COVA by varying it between 0 and 1. Performance changes are displayed in Figs.6(a)-6(d) for different versions of COVA, where t-SNE performance is displayed as a baseline. By increasing α , stronger cohort positioning and separation control is enforced, resulting in increased $S_c^{(r)}$ and S_s scores, which is however associated with a decreased S_n score. This delineates the trade-off between local neighbor and global cohort control. Compared to t-SNE, there is a mild drop in the S_n score of COVA. This is reasonable as COVA is designed to simultaneously achieve multiple objectives and cannot offer the highest scores for all measures. The output of COVA-E3 with different values of α is shown in Figs.6(e)-6(g). Lower α leads to less concentrated sample distributions within each cohort. In general, α should depend on whether a more spread-out scatter plot with less accurate cohort location control is preferred by the user over one with denser and more tightly controlled cohorts.

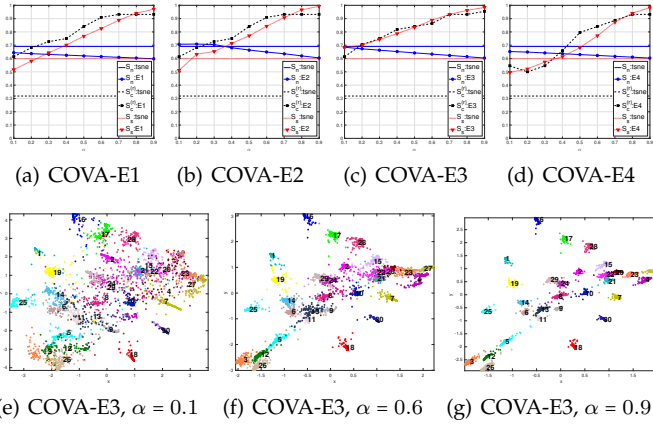


Fig. 6. (a)-(d): Performance change for the COVA embedding models by varying α from 0.1 to 0.9 using Places2 images. (e)-(g): Visualization output of COVA-E3 with $\alpha=0.1, 0.6,$ and 0.9 .

4.2.2 Projection based Visualization

In addition to visualization, we study out-of-sample extension for projection models by splitting the studied images from each class to training and test sets. The compared models project the same relation features \mathbf{F} computed by a Gaussian kernel with all the training samples used as the sample prototypes. Results are demonstrated with $\alpha=0.9$. The effects of COVA weights α and η (for eigen-COVA) are investigated in Appendix H.2 (see Fig.H.8).

We examine COVA projections against varying number of displayed scene classes, using 30% of images from each class for training and the remaining ones to test. Similar changing patterns are observed for multiple COVA projection models. Taking P1, P2 and eigen-COVA as examples, Figs.7(a)-7(c) display their performance change, where the performance of NCA is used as the baseline. As the class number increases, S_s and $S_c^{(r)}$ drop for both COVA and NCA. This shows that it becomes harder for projection models to control effectively the arrangement of higher number of classes. Local neighbor preservation is not affected though, as indicated by stable S_n score. Example output of COVA-P1 and NCA is displayed in Figs.7(e),7(f),7(h),7(i). Overall, COVA provides similar local sample neighbor control to NCA, but much better cohort control. Cohort control of COVA projections can be improved by using compacted features with combined cohort information (see Footnote 5). By using all the training samples as the representative samples and copying the corresponding elements from \mathbf{W} to $\tilde{\mathbf{S}}$ for compact feature generation, Fig.8 demonstrates the improved cohort arrangement using COVA-P1 and eigen-COVA, showing better match to the used cohort prototypes.

Appendix H.2 contains additional results. These include visual comparisons between COVA and other existing methods in addition to t-SNE, S-t-SNE and NCA (see Figs.H.6,H.7.(a)), and results of the more challenging task of visualizing 8,000 images from 80 scene classes using COVA and t-SNE (see Fig.H.8). We observe how COVA projection methods perform given different values of COVA weights α and η (see Fig.H.9) and varying number of training samples (see Fig.H.10). We also provide an overall quantitative comparison between COVA and existing methods in terms

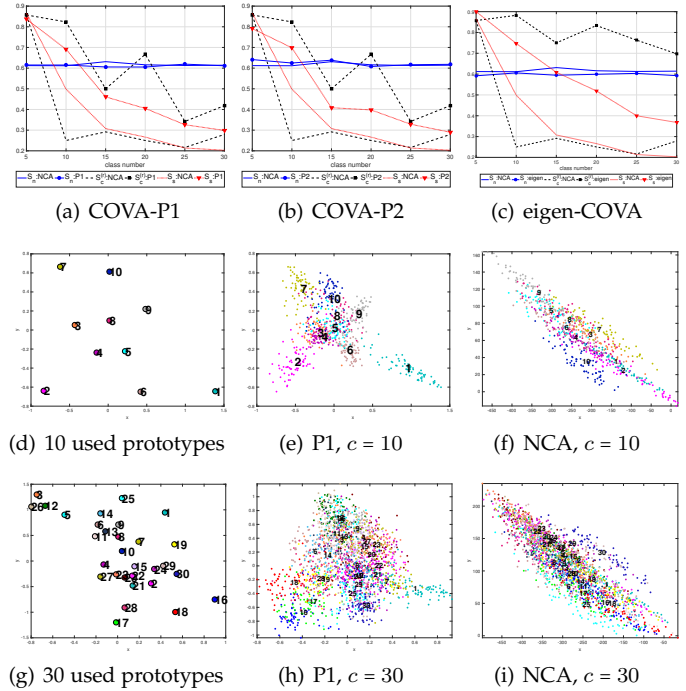


Fig. 7. (a)-(c): Performance change for the COVA projection models by varying class number from 5 to 30 using Places2 images. Example output is illustrated for $c=10, 30$, including (d),(g) displaying the used COVA cohort prototypes, (e),(h) COVA-P1 output, and (f),(i) NCA output.

of $S_n, S_c^{(r)}$ and S_s (see Table H.3) and computational time comparison between COVA and representative state-of-the-art methods (see Fig.H.11).

4.3 Publication Visualization with External Citations

We visualize 2,708 scientific publications from the Cora collection [35], classified into one of the seven predefined classes of “case based”, “genetic algorithms”, “neural networks”, “probabilistic methods”, “reinforcement learning”, “rule learning” and “theory”. Each publication is described by a binary word vector indicating the presence of 1,433 unique words. Additionally, citation link information between the publications is available, stored as a 2,708×2,708 binary matrix. Closeness between cohorts is computed externally from the citation links, as explained in Section 3.1.1, based on which cohort prototypes are computed using t-SNE so that their relative positioning approximates the citation strength between two publication cohorts. The generated cohort prototypes are displayed in Fig.9(a).

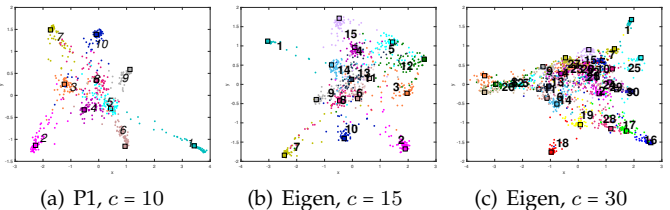


Fig. 8. Improved cohort arrangement exemplified by P1 and eigen-COVA for different class numbers. Cohort prototypes are shown as “□”. Different cohorts are numbered and correspond to different shadings.

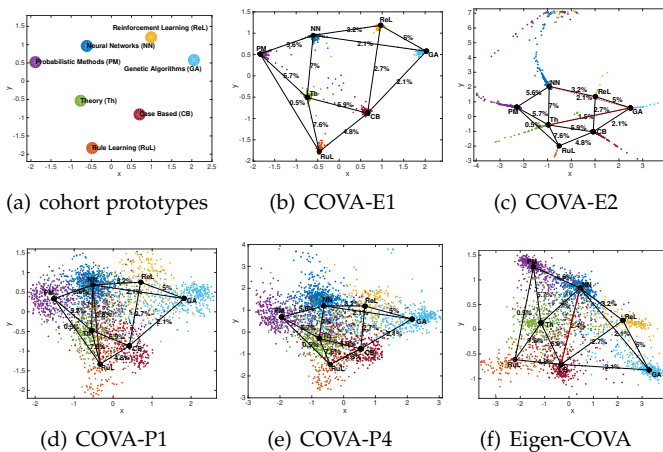


Fig. 9. COVA visualization for the Cora publications. Cohort locations are controlled by the citation information. Different cohorts correspond to different shadings. Centroids of the neighboring classes are connected with solid lines and connection mismatches are shown with dotted lines.

Both word content and citation information is used to derive enriched and compacted features for representing the documents; further details are provided in Appendix H.3. Euclidean distances are used to compute the local neighbor weight matrix \mathbf{W} by identifying 50 effective neighbors. The cohort membership matrix is computed via Eq.(12). Around 15% of the samples are randomly chosen to be the \tilde{k} sample prototypes for computing relation features. Given the simpler cohort structure of seven classes (compared to the thirty of Places2), there is a chance for COVA to preserve the exact cohort locations as indicated by the prototypes, and therefore S_c is used to measure cohort positioning quality.

Fig.9 displays the COVA output from E1, E2, P1, P4 and eigen-COVA with $\alpha=0.9$ and $\eta=0.92$. The output of other COVA models⁸ are shown in Appendix H.3 (see Fig.H.11). It is shown that the overall relative locations and proximities between the embedded cohorts match the distribution of the cohort prototypes. We connect the centroid of each class with its two undirected nearest (using Euclidean distances) neighboring classes. The between-class neighbor structure from each model can be directly compared with the between-prototype structure in Fig.9(a). It can be seen, that models E1 and P4 achieve a perfect match between the linked neighbors of the prototypes and the cohorts, while the remaining models achieve near perfect matches with only few mismatched links. The visualisation output of existing algorithms is shown in Appendix H.3 (see Fig.H.12), where the cohort locations and the closeness levels differ notably amongst them.

In Appendix H.3, we additionally examine the effect of α using COVA-E1 (see Fig.H.14), and show the quantitative performance comparison between COVA and existing methods for visualizing the seven publication cases (Table H.3). We also study the same task, as illustrated in Table 2, to visualize unsupervised publication clusters by COVA without relying on external information (see Figs.H.15,16).

8. The focus here is to compare the visualization output under the same testing environment, and thus, projection techniques are examined in the same setup as embedding ones without training-test split.

4.4 Additional Experiments

Additional experiments on visualizing different types of data objects, such as clinical trials, Flickr images and distributed semantic word vectors, are provided in Appendices H.4-H.6. Different methods that can be used to generate cohort prototypes are compared in Appendix H.7. A summary guide for the use of COVA is provided in Appendix H.8.

5 CONCLUSION

This work raised a critical issue of the current data visualization practices for high-dimensional data, with regard to their use of algorithms that generate low-dimensional representations of the data, since such methods only focus on preserving the local neighborhoods of the data patterns and maintaining or enhancing the separability between data cohorts. This results in obtaining patterns with the relevant positions and proximities between cohorts varying arbitrarily. We proposed a set of models, that can directly utilize information sources, such as the high-dimensional data itself or external domain-specific information, to administer control of the neighborhood structure in both data samples locally and cohort arrangements globally, via incorporating cohort prototypes as landmarks. For the introduced models, we also provided the mechanisms to optimize them and obtain the low-dimensional target patterns under a set of problem constraints using matrix manifold techniques. A very efficient projection model based on matrix decomposition was also proposed for large-scale applications. The notable effectiveness and improvement of the proposed algorithms over many existing methods for the purposes of visualization, was demonstrated both qualitatively and numerically using multiple synthetic and complex real-world text and image datasets.

ACKNOWLEDGMENT

The authors would like to thank the four anonymous reviewers for their very constructive and helpful comments.

REFERENCES

- [1] E. Bertini, A. Tatu, and D. Keim, "Quality metrics in high-dimensional data visualization: An overview and systematization," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, pp. 2203–2212, 2011.
- [2] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Mach. Learn.*, vol. 94, no. 2, pp. 233–259, 2014.
- [3] S. Kaski and J. Peltonen, "Dimensionality reduction for data visualization," *IEEE Sign. Proces. Mag.*, vol. 28, pp. 100–104, 2011.
- [4] R. Etemadpour, R. Motta, J. Paiva, M. O. R. Minghim, and L. Linsen, "Perception-based evaluation of projection methods for multidimensional data visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 1, pp. 81–94, 2015.
- [5] I. T. Jolliffe, *Principal Component Analysis*. New York, NY: Springer-Verlag, 1986.
- [6] P. Comon, "Independent component analysis: a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 187–134, 1994.
- [7] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3-4, pp. 321–377, 1936.
- [8] P. Dhillon, D. Foster, and L. Ungar, "Eigenwords: Spectral word embeddings," *J. Mach. Learn. Res.*, vol. 16, pp. 3035–3078, 2015.
- [9] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [10] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

- [11] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [12] T. Lin and H. Zha, "Riemannian manifold learning," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 30, no. 5, pp. 796–809, 2008.
- [13] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 34, pp. 253–265, 2012.
- [14] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, pp. 1027–1061, 2007.
- [15] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan, "A multimedia retrieval framework based on semi-supervised ranking and relevance feedback," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 723–742, 2012.
- [16] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2761–2773, 2010.
- [17] T. Mu, J. Y. Goulermas, J. Tsujii, and S. Ananiadou, "Proximity-based frameworks for generating embeddings from multi-output data," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2216–2232, 2012.
- [18] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *NIPS*, vol. 15, 2002, pp. 833–840.
- [19] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization," *J. Mach. Learn. Res.*, vol. 11, pp. 451–490, 2010.
- [20] L. J. P. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
- [21] J. Goldberger, S. T. Roweis, G. E. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *NIPS*, vol. 17, 2004.
- [22] A. Globerson and S. T. Roweis, "Metric learning by collapsing classes," in *NIPS*, 2005, pp. 451–458.
- [23] H. Kim, J. Choo, C. K. Reddy, and H. Park, "Doubly supervised embedding based on class labels and intrinsic clusters for high-dimensional data visualization," *Neurocomputing*, vol. 150, pp. 2399–2434, 2015.
- [24] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *ICML*, 2004.
- [25] K. Q. Weinberger, F. Sha, Q. Zhu, and L. K. Saul, "Graph laplacian regularization for large-scale semidefinite programming," in *NIPS*, vol. 19, 2007, p. 1489.
- [26] L. Song, A. Smola, K. Borgwardt, and A. Gretton, "Colored maximum variance unfolding," in *NIPS*, 2008.
- [27] N. Quadrianto and C. Lampert, "Learning multi-view neighborhood preserving projections," in *ICML*, 2011, pp. 425–432.
- [28] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 38, no. 1, pp. 188–194, 2015.
- [29] D. Eynard, A. Kovnatsky, M. M. Bronstein, K. Glashoff, and A. M. Bronstein, "Multimodal manifold analysis using simultaneous diagonalization of Laplacians," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2505–2517, 2015.
- [30] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [31] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [32] J. Weston, F. Rattle, and R. Collobert, "Deep learning via semi-supervised embedding," in *ICML*, 2008.
- [33] R. Min, L. Maaten, Z. Yuan, A. Bonner, and Z. Zhang, "Deep supervised t-distributed embedding," in *ICML*, 2010.
- [34] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *NIPS*, 2013.
- [35] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.
- [36] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: analysis and algorithm," in *NIPS 14*, 2001.
- [37] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [38] R. Xu, "Survey of clustering algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
- [39] E. Romero, T. Mu, and P. J. G. Lisboa, "Cohort-based kernel visualisation with scatter matrices," *Pattern Recognition*, vol. 45, no. 4, pp. 1436–1454, 2008.
- [40] I. S. Dhillona, D. S. Modhab, and W. S. Spanglerb, "Class visualization of high-dimensional data with applications," *Computational Statistics and Data Analysis*, vol. 41, pp. 59–90, 2002.
- [41] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Journal Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.
- [42] E. Pekalska, P. Paclik, and R. P. W. Duin, "A generalized kernel approach to dissimilarity-based classification," *J. Mach. Learn. Res.*, vol. 2, pp. 175–211, 2002.
- [43] P. A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton Uni. Press, 2008.
- [44] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1214–1236, 2013.
- [45] N. Boumal, B. Mishra, P. A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *J. Mach. Learn. Res.*, vol. 15, pp. 1455–1459, 2014.
- [46] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 22, pp. 888–905, 2000.
- [47] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. on Matrix Analysis and Applications*, vol. 29, no. 3, pp. 93–106, 2008.
- [48] H. Friedman and J. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Computers*, vol. C-23, no. 9, pp. 881–889, 1974.
- [49] E. Kokiopoulou and Y. Saad, "Orthogonal neighborhood preserving projections," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2143–2156, 2007.
- [50] E. Rodriguez-Martinez, J. Y. Goulermas, T. Mu, and J. F. Ralph, "Automatic induction of projection pursuit indices," *IEEE Trans. Neural Networks*, vol. 21, no. 8, pp. 1281–1295, 2010.
- [51] Y. Teh and S. Roweis, "Automatic alignment of hidden representations," in *NIPS*, 2003, pp. 841–848.
- [52] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [53] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," *arXiv preprint arXiv:1610.02055*, 2016.



Tingting Mu (M'05) received the B.Eng. degree in electronic engineering and information science from the Special Class for the Gifted Young, University of Science and Technology, Hefei, China, in 2004, and a Ph.D. in electrical engineering and electronics from the University of Liverpool in 2008. She is currently a Lecturer in the School of Computer Science, the University of Manchester. Her research interests include machine learning, mathematical modeling and optimization, with applications to language

and vision understanding.



John Goulermas (M'98, S'10) obtained the B.Sc. (1st) degree in Computation from the University of Manchester (UMIST), in 1994, and the M.Sc. and Ph.D. degrees from the Control Systems Center, UMIST, in 1996 and 2000, respectively. He is currently a Reader in the Department of Computer Science at the University of Liverpool. His current research interests include machine learning, combinatorial data analysis and mathematical modeling. He has worked with various applications, such as biomedical engineering, biomechanics, industrial monitoring and security.



Sophia Ananiadou is Director of the National Centre for Text Mining (NaCTeM), and Professor in the School of Computer Science, University of Manchester. She is the main designer of the text-mining tools and services currently used in NaCTeM. Her research projects include text mining-based visualisation of biochemical networks, data integration using text mining. She has been awarded the Daiwa Adrian prize (2004) and the IBM UIMA innovation award (2006,2007,2008) for her leading work on text-

mining tools in biomedicine.

Supplementary material for the manuscript: Data Visualization with Structural Control of Global Cohort and Local Data Neighborhoods.

T. Mu, J. Y. Goulermas, S. Ananiadou

APPENDIX A: DERIVING EUCLIDEAN GRADIENT

We summarize in the first column of Table A.1 the four rule-specific objective functions $O_{\text{dist}}^{(C)}(\mathbf{Z})$, $O_{\text{dist}}^{(L)}(\mathbf{Z})$, $O_{\text{KL}}^{(C)}(\mathbf{Z})$ and $O_{\text{KL}}^{(L)}(\mathbf{Z})$ after removing constant terms and rewriting them in matrix notation. The second column of Table A.1 includes the calculations for all their Euclidean gradients with respect to the embeddings, while Table A.2 with respect to the projections. New notations needed in the calculation include the following The $l \times c$ matrix $\mathbf{T} = [t_{ij}]$ with $t_{ij} = (1 + \|\mathbf{z}_i - \mathbf{v}_j\|_2^2)^{-1}$. The $l \times l$ matrix $\mathbf{Q} = [q_{ij}]$ with zero diagonal elements and off-diagonal ones computed by $q_{ij} = (1 + \|\mathbf{z}_i - \mathbf{z}_j\|_2^2)^{-1}$. A diagonal matrix $\mathbf{D}(\cdot)$ with its elements corresponding to the row sums of the matrix input. The $\log(\cdot)$ function is applied to a matrix element-wise. $\mathbf{0}_{m \times n}$ is the $m \times n$ matrix with zero elements, and $\mathbf{1}_n$ the n -length constant column vector of ones. In the following, we explain how to derive Euclidean gradient for COVA embeddings and projections given the four rule-specific objective functions.

A.1 OBJECTIVES $O_{\text{DIST}}^{(C)}$ AND $O_{\text{DIST}}^{(L)}$

The objective $O_{\text{dist}}^{(C)}$ can be expressed in matrix notation, as

$$\begin{aligned} O_{\text{dist}}^{(C)} &= \sum_{i=1}^l \sum_{j=1}^c r_{ij} (\mathbf{z}_i - \mathbf{v}_j)^T (\mathbf{z}_i - \mathbf{v}_j) \\ &= \sum_{i=1}^l \sum_{j=1}^c r_{ij} (\mathbf{z}_i^T \mathbf{z}_i - 2\mathbf{z}_i^T \mathbf{v}_j + \mathbf{v}_j^T \mathbf{v}_j) \\ &= \sum_{i=1}^l \left(\sum_{j=1}^c r_{ij} \right) \mathbf{z}_i^T \mathbf{z}_i - 2 \sum_{j=1}^c \sum_{i=1}^l r_{ij} \mathbf{z}_i^T \mathbf{v}_j + \sum_{j=1}^c \left(\sum_{i=1}^l r_{ij} \right) \mathbf{v}_j^T \mathbf{v}_j \\ &= \text{tr}(\mathbf{Z}^T \mathbf{D}(\mathbf{R}) \mathbf{Z}) - 2\text{tr}(\mathbf{Z}^T \mathbf{R} \mathbf{Y}) + \text{tr}(\mathbf{Y}^T \mathbf{D}(\mathbf{R}^T) \mathbf{Y}). \end{aligned} \quad (\text{A.1})$$

After removing the last constant term, the resulting formulation is reported in Table A.1. Its Euclidean gradients with respect to embeddings and projections are derived by following the first-order and second-order derivative of matrix trace, given as

$$\frac{\partial \text{tr}(\mathbf{X}^T \mathbf{A})}{\partial \mathbf{X}} = \mathbf{A}, \quad (\text{A.2})$$

$$\frac{\partial \text{tr}(\mathbf{X}^T \mathbf{A} \mathbf{X})}{\partial \mathbf{X}} = \mathbf{A} \mathbf{X} + \mathbf{A}^T \mathbf{X}, \quad (\text{A.3})$$

where \mathbf{X} denotes the variable matrix and \mathbf{A} a constant matrix. For embeddings, we have

$$\text{Grad } O_{\text{dist}}^{(C)}(\mathbf{Z}) = \frac{\partial O_{\text{dist}}^{(C)}(\mathbf{Z})}{\partial \mathbf{Z}} = 2\mathbf{D}(\mathbf{R})\mathbf{Z} - 2\mathbf{R}\mathbf{Y}. \quad (\text{A.4})$$

For projections, the objective function (with the constant term removed) becomes

$$O_{\text{dist}}^{(C)}(\mathbf{F}\mathbf{U}) = \text{tr}(\mathbf{U}^T \mathbf{F}^T \mathbf{D}(\mathbf{R}) \mathbf{F}\mathbf{U}) - 2\text{tr}(\mathbf{U}^T \mathbf{F}^T \mathbf{R} \mathbf{Y}), \quad (\text{A.5})$$

and its gradient is given as

$$\text{Grad } O_{\text{dist}}^{(C)}(\mathbf{U}) = \frac{\partial O_{\text{dist}}^{(C)}(\mathbf{U})}{\partial \mathbf{U}} = 2\mathbf{F}^T \mathbf{D}(\mathbf{R}) \mathbf{F}\mathbf{U} - 2\mathbf{F}^T \mathbf{R} \mathbf{Y}. \quad (\text{A.6})$$

Replacing $\{\mathbf{v}_j\}_{j=1}^c$ with $\{\mathbf{z}_j\}_{j=1}^l$ and r_{ij} with w_{ij} , $O_{\text{dist}}^{(L)}(\mathbf{Z})$, $\text{Grad } O_{\text{dist}}^{(L)}(\mathbf{Z})$ and $\text{Grad } O_{\text{dist}}^{(L)}(\mathbf{U})$ in matrix notation can be derived by following the same procedure.

A.2 OBJECTIVE $O_{\text{KL}}^{(C)}$

In the case of $O_{\text{KL}}^{(C)}$, with $\mathbf{R} = [r_{ij}]$ normalized to unit element sum, it can be written as

$$\begin{aligned} O_{\text{KL}}^{(C)} &= \sum_{i=1}^l \sum_{j=1}^c r_{ij} \log \frac{r_{ij}}{p(\mathbf{z}_i | j)} \\ &= \sum_{i=1}^l \sum_{j=1}^c r_{ij} \log r_{ij} - \sum_{i=1}^l \sum_{j=1}^c r_{ij} \log p(\mathbf{z}_i | j), \end{aligned} \quad (\text{A.7})$$

of which the first constant term does not affect the optimization. By focusing on the second term only, we have

$$\begin{aligned} O_{\text{KL}}^{(C)} &= - \sum_{i=1}^l \sum_{j=1}^c r_{ij} \log \left(\frac{(1 + \|\mathbf{z}_i - \mathbf{v}_j\|_2^2)^{-1}}{\sum_{s=1}^l \sum_{t=1}^c (1 + \|\mathbf{z}_s - \mathbf{v}_t\|_2^2)^{-1}} \right) \\ &= - \mathbf{1}_l^T \left(\mathbf{R} \circ \log \left(\mathbf{T} \left(\mathbf{1}_l^T \mathbf{T} \mathbf{1}_c \right)^{-1} \right) \right) \mathbf{1}_c, \end{aligned} \quad (\text{A.8})$$

where the two vectors $\mathbf{1}_l$ and $\mathbf{1}_c$ are used to realize the sum operations $\sum_{i=1}^l \sum_{j=1}^c (\cdot)$.

To assist the gradient calculation for $O_{\text{KL}}^{(C)}$, we introduce four auxiliary variables $d_{ij} = \|\mathbf{z}_i - \mathbf{v}_j\|_2$, $t_{ij} = (1 + d_{ij}^2)^{-1}$,

Table A.1
Rule-specific objective functions and their Euclidean gradients for computing COVA embeddings.

Objective function (with constant term removed):	Euclidean gradient:
$O_{\text{dist}}^{(C)}(\mathbf{Z}) = \text{tr}(\mathbf{Z}^T \mathbf{D}(\mathbf{R})\mathbf{Z}) - 2\text{tr}(\mathbf{Z}^T \mathbf{R}\mathbf{Y})$	$\text{Grad } O_{\text{dist}}^{(C)}(\mathbf{Z}) = 2\mathbf{D}(\mathbf{R})\mathbf{Z} - 2\mathbf{R}\mathbf{Y}$
$O_{\text{dist}}^{(L)}(\mathbf{Z}) = \text{tr}(\mathbf{Z}^T \mathbf{L}(\mathbf{W})\mathbf{Z})$	$\text{Grad } O_{\text{dist}}^{(L)}(\mathbf{Z}) = 2\mathbf{L}(\mathbf{W})\mathbf{Z}$
$O_{\text{KL}}^{(C)}(\mathbf{Z}) = -\mathbf{1}_l^T \left(\mathbf{R} \circ \log \left(\mathbf{T} \left(\mathbf{1}_l^T \mathbf{T} \mathbf{1}_c \right)^{-1} \right) \right) \mathbf{1}_c$	$\text{Grad } O_{\text{KL}}^{(C)}(\mathbf{Z}) = 2\mathbf{D}(\mathbf{H})\mathbf{Z} - 2\mathbf{H}\mathbf{Y}$, where $\mathbf{H} = \left(\mathbf{R} - \mathbf{T} \left(\mathbf{1}_l^T \mathbf{T} \mathbf{1}_c \right)^{-1} \right) \circ \mathbf{T}$
$O_{\text{KL}}^{(L)}(\mathbf{Z}) = -\mathbf{1}_l^T \mathbf{N} \mathbf{1}_l$, where $\mathbf{N} = \left(\mathbf{1}_l^T \mathbf{W} \mathbf{1}_l \right)^{-1} \mathbf{W} \circ \log \left(\left(\mathbf{1}_l^T \mathbf{Q} \mathbf{1}_l \right)^{-1} \mathbf{Q} \right)$	$\text{Grad } O_{\text{KL}}^{(L)}(\mathbf{Z}) = 4\mathbf{L}(\mathbf{G})\mathbf{Z}$, where $\mathbf{G} = \left(\left(\mathbf{1}_l^T \mathbf{W} \mathbf{1}_l \right)^{-1} \mathbf{W} - \left(\mathbf{1}_l^T \mathbf{Q} \mathbf{1}_l \right)^{-1} \mathbf{Q} \right) \circ \mathbf{Q}$

Table A.2

Euclidean gradients of the rule-specific objectives for computing COVA projections (\mathbf{H} , \mathbf{G} , $\mathbf{D}(\cdot)$) and $\mathbf{L}(\cdot)$ are as defined for Table A.1).

$\text{Grad } O_{\text{dist}}^{(C)}(\mathbf{U}) = 2\mathbf{F}^T \mathbf{D}(\mathbf{R})\mathbf{F}\mathbf{U} - 2\mathbf{F}^T \mathbf{R}\mathbf{Y}$
$\text{Grad } O_{\text{dist}}^{(L)}(\mathbf{U}) = 2\mathbf{F}^T \mathbf{L}(\mathbf{W})\mathbf{F}\mathbf{U}$
$\text{Grad } O_{\text{KL}}^{(C)}(\mathbf{U}) = 2 \sum_{i=1}^l \sum_{j=1}^c h_{ij} \left(\phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) \mathbf{U} - \phi_r(\mathbf{x}_i) \mathbf{v}_j^T \right)$
$\text{Grad } O_{\text{KL}}^{(L)}(\mathbf{U}) = 4 \left(\mathbf{F}^T \mathbf{D}(\mathbf{G})\mathbf{F} - \sum_{i=1}^l \sum_{j \neq i}^c g_{ij} \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_j) \right) \mathbf{U}$

$U = \sum_{i=1}^l \sum_{j=1}^c t_{ij}$ and $\nu_{ij} = t_{ij} U^{-1}$, and re-express $O_{\text{KL}}^{(C)}$ by utilizing $\sum_{i=1}^l \sum_{j=1}^c r_{ij} = 1$. This gives

$$\begin{aligned}
O_{\text{KL}}^{(C)} &= - \sum_{i=1}^l \sum_{j=1}^c r_{ij} \log \left(\frac{t_{ij}}{U} \right) \\
&= - \sum_{i=1}^l \sum_{j=1}^c r_{ij} \log t_{ij} + \log U \left(\sum_{i=1}^l \sum_{j=1}^c r_{ij} \right) \\
&= - \sum_{i=1}^l \sum_{j=1}^c r_{ij} \log t_{ij} + \log U. \tag{A.9}
\end{aligned}$$

It is straightforward to compute

$$\frac{\partial d_{ij}}{\partial \mathbf{z}_i} = \frac{\mathbf{z}_i - \mathbf{v}_j}{d_{ij}}, \tag{A.10}$$

$$\frac{\partial t_{ij}}{\partial d_{ij}} = -2d_{ij} t_{ij}^2, \tag{A.11}$$

$$\frac{\partial \log t_{ij}}{\partial d_{ij}} = \frac{1}{t_{ij}} \frac{\partial t_{ij}}{\partial d_{ij}} = -2d_{ij} t_{ij}, \tag{A.12}$$

$$\frac{\partial \log U}{\partial d_{ij}} = U^{-1} \sum_{s=1}^l \sum_{h=1}^c \frac{\partial t_{sh}}{\partial d_{ij}} = U^{-1} \frac{\partial t_{ij}}{\partial d_{ij}} = -2d_{ij} \nu_{ij} t_{ij}. \tag{A.13}$$

By utilizing Eqs.(A.9,A.12,A.13), we have

$$\begin{aligned}
\frac{\partial O_{\text{KL}}^{(C)}}{\partial d_{ij}} &= - \sum_{s=1}^l \sum_{h=1}^c r_{sh} \frac{\partial \log t_{sh}}{\partial d_{ij}} + \frac{\partial \log U}{\partial d_{ij}} \\
&= - r_{ij} \frac{\partial \log t_{ij}}{\partial d_{ij}} + \frac{\partial \log U}{\partial d_{ij}} \\
&= 2(r_{ij} - \nu_{ij}) d_{ij} t_{ij}. \tag{A.14}
\end{aligned}$$

Since the embedding vector \mathbf{z}_i affects $O_{\text{KL}}^{(C)}$ through $\{d_{ij}\}_{j=1}^c$, we obtain the following gradient formulation based on Eqs.(A.10,A.14)

$$\begin{aligned}
\frac{\partial O_{\text{KL}}^{(C)}}{\partial \mathbf{z}_i} &= \sum_{j=1}^c \frac{\partial O_{\text{KL}}^{(C)}}{\partial d_{ij}} \frac{\partial d_{ij}}{\partial \mathbf{z}_i} \\
&= \sum_{j=1}^c 2(r_{ij} - \nu_{ij}) t_{ij} (\mathbf{z}_i - \mathbf{v}_j). \tag{A.15}
\end{aligned}$$

Letting $h_{ij} = (r_{ij} - \nu_{ij}) t_{ij}$, which corresponds to the ij -th element of the matrix $\mathbf{H} = \left(\mathbf{R} - \mathbf{T} \left(\mathbf{1}_l^T \mathbf{T} \mathbf{1}_c \right)^{-1} \right) \circ \mathbf{T}$, we have

$$\frac{\partial O_{\text{KL}}^{(C)}}{\partial \mathbf{z}_i} = \sum_{j=1}^c 2h_{ij} (\mathbf{z}_i - \mathbf{v}_j) = 2 \left(\sum_{j=1}^c h_{ij} \right) \mathbf{z}_i - 2 \sum_{j=1}^c h_{ij} \mathbf{v}_j, \tag{A.16}$$

resulting in the following gradient in matrix notation

$$\text{Grad } O_{\text{KL}}^{(C)}(\mathbf{Z}) = \frac{\partial O_{\text{KL}}^{(C)}}{\partial \mathbf{Z}} = 2\mathbf{D}(\mathbf{H})\mathbf{Z} - 2\mathbf{H}\mathbf{Y}. \tag{A.17}$$

To compute the gradient of $O_{\text{KL}}^{(C)}$ with respect to the projection matrix utilizing $\mathbf{z}_i = \mathbf{U}^T \phi_r(\mathbf{x}_i)$, we start from the auxiliary variable d_{ij} , which is

$$\begin{aligned}
d_{ij} &= \|\mathbf{U}^T \phi_r(\mathbf{x}_i) - \mathbf{v}_j\|_2 \\
&= \left[\text{tr} \left(\mathbf{U}^T \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) \mathbf{U} \right) - 2\text{tr} \left(\mathbf{U}^T \phi_r(\mathbf{x}_i) \mathbf{v}_j^T \right) + \mathbf{v}_j^T \mathbf{v}_j \right]^{\frac{1}{2}}. \tag{A.18}
\end{aligned}$$

Assisted by Eqs.(A.2,A.3), we have

$$\frac{\partial d_{ij}}{\partial \mathbf{U}} = \frac{1}{d_{ij}} \left(\phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) \mathbf{U} - \phi_r(\mathbf{x}_i) \mathbf{v}_j^T \right). \tag{A.19}$$

Utilizing Eqs.(A.9,A.11), we have

$$\begin{aligned}
\frac{\partial O_{\text{KL}}^{(C)}}{\partial \mathbf{U}} &= - \sum_{i=1}^l \sum_{j=1}^c r_{ij} \frac{\partial \log t_{ij}}{\partial \mathbf{U}} + \frac{\partial \log U}{\partial \mathbf{U}} \\
&= - \sum_{i=1}^l \sum_{j=1}^c \frac{r_{ij}}{t_{ij}} \frac{\partial t_{ij}}{\partial \mathbf{U}} + \frac{1}{U} \sum_{i=1}^l \sum_{j=1}^c \frac{\partial t_{ij}}{\partial \mathbf{U}} \\
&= - \sum_{i=1}^l \sum_{j=1}^c \left(\frac{r_{ij}}{t_{ij}} - \frac{1}{U} \right) \frac{\partial t_{ij}}{\partial d_{ij}} \frac{\partial d_{ij}}{\partial \mathbf{U}} \\
&= 2 \sum_{i=1}^l \sum_{j=1}^c \left(\frac{r_{ij}}{t_{ij}} - \frac{1}{U} \right) d_{ij} t_{ij}^2 \frac{\partial d_{ij}}{\partial \mathbf{U}} \\
&= 2 \sum_{i=1}^l \sum_{j=1}^c (r_{ij} - \nu_{ij}) d_{ij} t_{ij} \frac{\partial d_{ij}}{\partial \mathbf{U}}. \tag{A.20}
\end{aligned}$$

By incorporating Eq.(A.19) into Eq.(A.20) and using $h_{ij} = (r_{ij} - \nu_{ij}) t_{ij}$, the final gradient can be computed by

$$\frac{\partial O_{\text{KL}}^{(C)}}{\partial \mathbf{U}} = 2 \sum_{i=1}^l \sum_{j=1}^c h_{ij} \left(\phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) \mathbf{U} - \phi_r(\mathbf{x}_i) \mathbf{v}_j^T \right). \tag{A.21}$$

A.3 OBJECTIVE $O_{\text{KL}}^{(L)}$

Working with $\mathbf{W} = [w_{ij}]$ normalized to unit element sum, the following equivalent version of $O_{\text{KL}}^{(L)}$ with constant term removed is used, given as

$$O_{\text{KL}}^{(L)} = - \sum_{i=1}^l \sum_{j \neq i} w_{ij} \log \frac{(1 + \|\mathbf{z}_i - \mathbf{z}_j\|_2^2)^{-1}}{\sum_{s=1}^l \sum_{t \neq s} (1 + \|\mathbf{z}_s - \mathbf{z}_t\|_2^2)^{-1}}. \tag{A.22}$$

Letting the zero-diagonal matrix \mathbf{Q} store $(1 + \|\mathbf{z}_i - \mathbf{z}_j\|_2^2)^{-1}$ in its off-diagonals, $O_{\text{KL}}^{(L)}$ can be expressed in matrix notations as

$$O_{\text{KL}}^{(L)} = -\mathbf{1}_l^T \left[\left(\mathbf{1}_l^T \mathbf{W} \mathbf{1}_l \right)^{-1} \mathbf{W} \circ \log \left(\left(\mathbf{1}_l^T \mathbf{Q} \mathbf{1}_l \right)^{-1} \mathbf{Q} \right) \right] \mathbf{1}_l, \quad (\text{A.23})$$

where $0 \times \log 0 = 0$ is defined to deal with the zero diagonals of \mathbf{W} and \mathbf{Q} . To derive the Euclidean gradient $O_{\text{KL}}^{(L)}$, four auxiliary variables $\tilde{d}_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|_2$, $q_{ij} = (1 + \tilde{d}_{ij}^2)^{-1}$, $U = \sum_{i=1}^l \sum_{j \neq i} q_{ij}$ and $\nu_{ij} = q_{ij} U^{-1}$ are introduced. Utilizing $\sum_{i=1}^l \sum_{j \neq i} w_{ij} = 1$, we have

$$\begin{aligned} O_{\text{KL}}^{(L)} &= - \sum_{i=1}^l \sum_{j \neq i} w_{ij} \log \left(\frac{q_{ij}}{U} \right) \\ &= - \sum_{i=1}^l \sum_{j \neq i} w_{ij} \log q_{ij} + \log U \sum_{i=1}^l \sum_{j \neq i} w_{ij} \\ &= - \sum_{i=1}^l \sum_{j \neq i} w_{ij} \log \tilde{d}_{ij} + \log U. \end{aligned} \quad (\text{A.24})$$

It is straightforward to derive the following

$$\frac{\partial \tilde{d}_{ij}}{\partial \mathbf{z}_i} = \frac{\partial \tilde{d}_{ji}}{\partial \mathbf{z}_i} = \frac{\mathbf{z}_i - \mathbf{z}_j}{\tilde{d}_{ij}}, \quad (\text{A.25})$$

$$\frac{\partial q_{ij}}{\partial \tilde{d}_{ij}} = -2\tilde{d}_{ij} q_{ij}^2, \quad (\text{A.26})$$

$$\frac{\partial \log q_{ij}}{\partial \tilde{d}_{ij}} = \frac{1}{q_{ij}} \frac{\partial q_{ij}}{\partial \tilde{d}_{ij}} = -2\tilde{d}_{ij} q_{ij}, \quad (\text{A.27})$$

Euclidean gradient of the first term of Eq.(A.24), denoted by T_1 , can be obtained by utilizing Eqs.(A.25,A.27) and the symmetry of w_{ij} , q_{ij} and \tilde{d}_{ij} , as

$$\begin{aligned} \frac{\partial T_1}{\partial \mathbf{z}_i} &= - \sum_{s=1}^l \sum_{t \neq s} w_{st} \frac{\partial \log q_{st}}{\partial \mathbf{z}_i} \\ &= - \left(w_{ij} \frac{\partial \log q_{ij}}{\partial \tilde{d}_{ij}} \frac{\partial \tilde{d}_{ij}}{\partial \mathbf{z}_i} + w_{ji} \frac{\partial \log q_{ji}}{\partial \tilde{d}_{ji}} \frac{\partial \tilde{d}_{ji}}{\partial \mathbf{z}_i} \right) \\ &= (2w_{ij}q_{ij} + 2w_{ji}q_{ji})(\mathbf{z}_i - \mathbf{z}_j) \\ &= 4w_{ij}q_{ij}(\mathbf{z}_i - \mathbf{z}_j). \end{aligned} \quad (\text{A.28})$$

Euclidean gradient of the second term of Eq.(A.24) can be obtained by utilizing Eqs.(A.25,A.26) and the symmetry of ν_{ij} , q_{ij} and \tilde{d}_{ij} , as

$$\begin{aligned} \frac{\partial \log U}{\partial \mathbf{z}_i} &= \frac{1}{U} \sum_{s=1}^l \sum_{t \neq s} \frac{\partial q_{st}}{\partial \mathbf{z}_i} = \frac{1}{U} \left(\frac{\partial q_{ij}}{\partial \tilde{d}_{ij}} \frac{\partial \tilde{d}_{ij}}{\partial \mathbf{z}_i} + \frac{\partial q_{ji}}{\partial \tilde{d}_{ji}} \frac{\partial \tilde{d}_{ji}}{\partial \mathbf{z}_i} \right) \\ &= - \left(2\tilde{d}_{ij}\nu_{ij}q_{ij} + 2\tilde{d}_{ji}\nu_{ji}q_{ji} \right) \frac{\partial \tilde{d}_{ij}}{\partial \mathbf{z}_i} \\ &= -4\nu_{ij}q_{ij}(\mathbf{z}_i - \mathbf{z}_j). \end{aligned} \quad (\text{A.29})$$

Combining Eqs.(A.28,A.29), we get

$$\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{z}_i} = \frac{\partial T_1}{\partial \mathbf{z}_i} + \frac{\partial \log U}{\partial \mathbf{z}_i} = 4(w_{ij} - \nu_{ij})q_{ij}(\mathbf{z}_i - \mathbf{z}_j). \quad (\text{A.30})$$

Letting $g_{ij} = (w_{ij} - \nu_{ij})q_{ij} = (w_{ij} - q_{ij}U^{-1})q_{ij}$, the gradient formulation can be simplified as

$$\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{z}_i} = 4g_{ij}(\mathbf{z}_i - \mathbf{z}_j). \quad (\text{A.31})$$

To derive the equivalent matrix version of Eq.(A.31), we first store $\{g_{ij}\}_{i,j=1}^l$ in the matrix $\mathbf{G} = \left((\mathbf{1}_l^T \mathbf{W} \mathbf{1}_l)^{-1} \mathbf{W} - (\mathbf{1}_l^T \mathbf{Q} \mathbf{1}_l)^{-1} \mathbf{Q} \right) \circ \mathbf{Q}$. It is obvious that Eq.(A.31) is also the gradient of

$$O = 2 \sum_{i=1}^l \sum_{j \neq i} g_{ij} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 = 2\text{tr} \left(\mathbf{Z}^T \mathbf{L}(\mathbf{G}) \mathbf{Z} \right), \quad (\text{A.32})$$

where $\mathbf{L}(\mathbf{G})$ is the Laplacian matrix of \mathbf{G} . Since

$$\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{z}_i} = \frac{\partial O}{\partial \mathbf{z}_i} \Rightarrow \frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{Z}} = \frac{\partial O}{\partial \mathbf{Z}}, \quad (\text{A.33})$$

we can simply obtain

$$\text{Grad } O_{\text{KL}}^{(L)}(\mathbf{Z}) = \frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{Z}} = \frac{\partial O}{\partial \mathbf{Z}} = 4\mathbf{L}(\mathbf{G})\mathbf{Z}. \quad (\text{A.34})$$

To compute the Euclidean gradient of $O_{\text{KL}}^{(L)}$ with respect to the projection matrix, we start from

$$\begin{aligned} \tilde{d}_{ij} &= \|\mathbf{U}^T \phi_r(\mathbf{x}_i) - \mathbf{U}^T \phi_r(\mathbf{x}_j)\|_2 \\ &= \left[\text{tr} \left(\mathbf{U}^T \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) \mathbf{U} \right) - 2\text{tr} \left(\mathbf{U}^T \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_j) \mathbf{U} \right) \right. \\ &\quad \left. + \text{tr} \left(\mathbf{U}^T \phi_r(\mathbf{x}_j) \phi_r^T(\mathbf{x}_j) \mathbf{U} \right) \right]^{\frac{1}{2}}, \end{aligned} \quad (\text{A.35})$$

Based on Eqs.(A.2,A.3), we have

$$\frac{\partial \tilde{d}_{ij}}{\partial \mathbf{U}} = \frac{1}{\tilde{d}_{ij}} \left(\phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) + \phi_r(\mathbf{x}_j) \phi_r^T(\mathbf{x}_j) - \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_j) \right) \mathbf{U}. \quad (\text{A.36})$$

Utilizing Eqs.(A.24,A.26), we have

$$\begin{aligned} \frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{U}} &= - \sum_{i=1}^l \sum_{j \neq i} w_{ij} \frac{\partial \log q_{ij}}{\partial \mathbf{U}} + \frac{\partial \log U}{\partial \mathbf{U}} \\ &= - \sum_{i=1}^l \sum_{j \neq i} \frac{w_{ij}}{q_{ij}} \frac{\partial q_{ij}}{\partial \mathbf{U}} + \frac{1}{U} \sum_{i=1}^l \sum_{j \neq i} \frac{\partial q_{ij}}{\partial \mathbf{U}} \\ &= - \sum_{i=1}^l \sum_{j \neq i} \left(\frac{w_{ij}}{q_{ij}} - \frac{1}{U} \right) \frac{\partial q_{ij}}{\partial \tilde{d}_{ij}} \frac{\partial \tilde{d}_{ij}}{\partial \mathbf{U}} \\ &= 2 \sum_{i=1}^l \sum_{j \neq i} \left(\frac{w_{ij}}{q_{ij}} - \frac{1}{U} \right) \tilde{d}_{ij} q_{ij}^2 \frac{\partial \tilde{d}_{ij}}{\partial \mathbf{U}} \\ &= 2 \sum_{i=1}^l \sum_{j \neq i} (w_{ij} - \nu_{ij}) \tilde{d}_{ij} q_{ij} \frac{\partial \tilde{d}_{ij}}{\partial \mathbf{U}} = 2 \sum_{i=1}^l \sum_{j \neq i} \tilde{d}_{ij} g_{ij} \frac{\partial \tilde{d}_{ij}}{\partial \mathbf{U}}. \end{aligned} \quad (\text{A.37})$$

As shown in Eq.(A.36), $\frac{\partial \tilde{d}_{ij}}{\partial \mathbf{U}}$ contains three terms including

$$T_{ij}^{(1)} = \tilde{d}_{ij}^{-1} \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) \mathbf{U}, \quad (\text{A.38})$$

$$T_{ij}^{(2)} = \tilde{d}_{ij}^{-1} \phi_r(\mathbf{x}_j) \phi_r^T(\mathbf{x}_j) \mathbf{U}, \quad (\text{A.39})$$

$$T_{ij}^{(3)} = -2\tilde{d}_{ij}^{-1} \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_j) \mathbf{U}. \quad (\text{A.40})$$

The first term contributes to $\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{U}}$ by

$$\begin{aligned} \left(\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{U}} \right)_1 &= 2 \sum_{i=1}^l \sum_{j \neq i} \tilde{d}_{ij} g_{ij} T_{ij}^{(1)} \\ &= 2 \left(\sum_{i=1}^l \left(\sum_{j=1}^l g_{ij} \right) \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) \right) \mathbf{U} \\ &= 2\mathbf{F}^T \mathbf{D}(\mathbf{G}) \mathbf{F} \mathbf{U}, \end{aligned} \quad (\text{A.41})$$

the second term by

$$\begin{aligned} \left(\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{U}} \right)_2 &= 2 \sum_{i=1}^l \sum_{j \neq i} \tilde{d}_{ij} g_{ij} T_{ij}^{(2)} \\ &= 2 \left(\sum_{j=1}^l \left(\sum_{i=1}^l g_{ij} \right) \phi_r(\mathbf{x}_j) \phi_r^T(\mathbf{x}_j) \right) \mathbf{U} \\ &= 2 \mathbf{F}^T \mathbf{D}(\mathbf{G}) \mathbf{F} \mathbf{U}. \end{aligned} \quad (\text{A.42})$$

and the last term by

$$\begin{aligned} \left(\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{U}} \right)_3 &= 2 \sum_{i=1}^l \sum_{j \neq i} \tilde{d}_{ij} g_{ij} T_{ij}^{(3)} \\ &= -4 \left(\sum_{i=1}^l \sum_{j \neq i} g_{ij} \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) \right) \mathbf{U}. \end{aligned} \quad (\text{A.43})$$

The final Euclidean gradient of $O_{\text{KL}}^{(L)}$ with respect to the projection matrix is thus given by

$$\begin{aligned} \frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{U}} &= \left(\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{U}} \right)_1 + \left(\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{U}} \right)_2 + \left(\frac{\partial O_{\text{KL}}^{(L)}}{\partial \mathbf{U}} \right)_3 \\ &= 4 \left(\mathbf{F}^T \mathbf{D}(\mathbf{G}) \mathbf{F} - 4 \left(\sum_{i=1}^l \sum_{j \neq i} g_{ij} \phi_r(\mathbf{x}_i) \phi_r^T(\mathbf{x}_i) \right) \right) \mathbf{U}. \end{aligned} \quad (\text{A.44})$$

APPENDIX B: STIEFEL MANIFOLD

The set of $\tilde{k} \times k$ matrices is denoted as the Stiefel manifold

$$M = \left\{ \mathbf{U} \mid \mathbf{U} \in \mathbb{R}^{\tilde{k} \times k}, \mathbf{U}^T \mathbf{U} = \mathbf{I}_{k \times k} \right\}, \quad (\text{B.1})$$

which is an embedded manifold of the vector space $\mathbb{R}^{\tilde{k} \times k}$ with the dimension $\tilde{k}k - \frac{1}{2}k(k+1)$. In the task of dimensionality reduction, we always have $\tilde{k} \geq k$. We provide below a property summary of Stiefel manifold and show how to derive several manifold defined constructs that are related to our work. More details on optimization on matrix manifold can be found in [1].

The tangent vector of a manifold M at the point $\mathbf{U} \in M$ is defined as a mapping $\xi_{\mathbf{U}} : \mathfrak{F}_{\mathbf{U}}(M) \rightarrow \mathbb{R}$ from a collection of smooth real-valued functions $\mathfrak{F}_{\mathbf{U}}(M)$ to \mathbb{R} . Each function $f \in \mathfrak{F}_{\mathbf{U}}(M)$ is defined on a neighborhood of \mathbf{U} in the manifold. Definition of the mapping $\xi_{\mathbf{U}}(f)$ relies on the existence of a curve that lies within the manifold and starts from \mathbf{U} , denoted by $u(t) \in M$ with $u(0) = \mathbf{U}$. When working with an embedded manifold, we have

$$\xi_{\mathbf{U}}(f) = \dot{u}(t)f = \left. \frac{df(u(t))}{dt} \right|_{t=0} = D\bar{f}(\mathbf{U})[u'(0)], \quad (\text{B.2})$$

where \bar{f} is a real-valued function in a neighborhood of \mathbf{U} in the vector space $\mathbb{R}^{\tilde{k} \times k}$, and f is its restriction to the manifold. Therefore, a tangent vector can be viewed as a generalization of the notion of a directional derivative. A natural correspondence can be established between a tangent vector $\xi_{\mathbf{U}}$ and the $\tilde{k} \times k$ matrix $u'(0)$.

The Stiefel manifold satisfies the equation $\mathbf{U}^T \mathbf{U} = \mathbf{I}_{k \times k}$. Therefore, the curve that realizes the tangent vector $\xi_{\mathbf{U}}$ at the point $\mathbf{U} \in M$ should satisfy $u^T(t)u(t) = \mathbf{I}_{k \times k}$. By differentiating both sides of the equation, it is

$$u'^T(t)u(t) + u^T(t)u'(t) = \mathbf{0}_{k \times k}. \quad (\text{B.3})$$

Letting $t = 0$, this results in

$$\xi_{\mathbf{U}}^T \mathbf{U} + \mathbf{U}^T \xi_{\mathbf{U}} = \mathbf{0}_{k \times k}. \quad (\text{B.4})$$

Given $\mathbf{U} \in M$ and its normalized orthogonal complement \mathbf{U}_{\perp} of size $\tilde{k} \times (\tilde{k} - k)$ that satisfies $\mathbf{U}_{\perp}^T \mathbf{U} = \mathbf{0}_{(\tilde{k}-k) \times k}$ and $\mathbf{U}_{\perp}^T \mathbf{U}_{\perp} = \mathbf{I}_{(\tilde{k}-k) \times (\tilde{k}-k)}$, an arbitrary matrix $\xi \in \mathbb{R}^{\tilde{k} \times k}$ can be represented as

$$\xi = \mathbf{U} \Omega_1 + \mathbf{U}_{\perp} \Omega_2, \quad (\text{B.5})$$

where $\Omega_1 \in \mathbb{R}^{k \times k}$ and $\Omega_2 \in \mathbb{R}^{(\tilde{k}-k) \times k}$. In order to enable ξ to be a tangent vector of the Stiefel manifold, Eq.(B.4) needs to be satisfied, which results in

$$\begin{aligned} (\mathbf{U} \Omega_1 + \mathbf{U}_{\perp} \Omega_2)^T \mathbf{U} + \mathbf{U}^T (\mathbf{U} \Omega_1 + \mathbf{U}_{\perp} \Omega_2) &= \mathbf{0}_{k \times k} \\ \Rightarrow \Omega_1^T \mathbf{U}^T \mathbf{U} + \Omega_2^T \mathbf{U}_{\perp}^T \mathbf{U} + \mathbf{U}^T \mathbf{U} \Omega_1 + \mathbf{U}^T \mathbf{U}_{\perp} \Omega_2 &= \mathbf{0}_{k \times k} \\ \Rightarrow \Omega_1^T + \Omega_1 &= \mathbf{0}_{k \times k}. \end{aligned} \quad (\text{B.6})$$

This indicates that the matrix Ω_1 should be skew-symmetric in order to generate a tangent vector. Therefore, the tangent space of the Stiefel manifold can be expressed as

$$T_{\mathbf{U}} \xi = \left\{ \mathbf{U} \Omega_1 + \mathbf{U}_{\perp} \Omega_2 \mid \Omega_1 \in \mathbb{R}^{k \times k}, \Omega_1 = -\Omega_1^T, \Omega_2 \in \mathbb{R}^{(\tilde{k}-k) \times k} \right\}. \quad (\text{B.7})$$

The projection of an arbitrary matrix $\xi \in \mathbb{R}^{\tilde{k} \times k}$ onto the tangent space $T_{\mathbf{U}} \xi$ can be obtained by computing the closest point in $T_{\mathbf{U}} \xi$ to ξ , which is equivalent to finding the least squared solution [2] of the following problem

$$\min_{\substack{\Omega_1 \in \mathbb{R}^{k \times k}, \\ \Omega_1 = -\Omega_1^T, \\ \Omega_2 \in \mathbb{R}^{(\tilde{k}-k) \times k}}} O = \|\mathbf{U} \Omega_1 + \mathbf{U}_{\perp} \Omega_2 - \xi\|_F^2, \quad (\text{B.8})$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Utilizing $\mathbf{U}^T \mathbf{U} = \mathbf{I}_{k \times k}$, $\mathbf{U}_{\perp}^T \mathbf{U}_{\perp} = \mathbf{I}_{(\tilde{k}-k) \times (\tilde{k}-k)}$ and $\mathbf{U}_{\perp}^T \mathbf{U} = \mathbf{0}_{(\tilde{k}-k) \times k}$, the objective function can be written as

$$\begin{aligned} O &= \text{tr} \left[(\mathbf{U} \Omega_1 + \mathbf{U}_{\perp} \Omega_2 - \xi)^T (\mathbf{U} \Omega_1 + \mathbf{U}_{\perp} \Omega_2 - \xi) \right] \\ &= \text{tr} \left(\Omega_1^T \Omega_1 \right) - 2 \text{tr} \left(\Omega_1^T \mathbf{U}^T \xi \right) + \\ &\quad \text{tr} \left(\Omega_2^T \Omega_2 \right) - 2 \text{tr} \left(\Omega_2^T \mathbf{U}_{\perp}^T \xi \right) + \text{tr} \left(\xi^T \xi \right). \end{aligned} \quad (\text{B.9})$$

Since the matrix Ω_2 is unconstrained, its optimal solution can be easily computed by setting $\frac{dO}{d\Omega_2} = 0$, which gives

$$\frac{dO}{d\Omega_2} = 2\Omega_2 - 2\mathbf{U}_{\perp}^T \xi = 0, \quad (\text{B.10})$$

thus

$$\Omega_2^* = \mathbf{U}_{\perp}^T \xi. \quad (\text{B.11})$$

Differently, the matrix Ω_1 is constrained as a skew-symmetric matrix. Therefore, we express it as $\Omega_1 = \frac{1}{2}(\mathbf{A} - \mathbf{A}^T)$ with $\mathbf{A} \in \mathbb{R}^{k \times k}$ and incorporate this into Eq.(B.9) so that \mathbf{A} is optimized without any constraint. Euclidean gradient of O with respect to \mathbf{A} is given by

$$\begin{aligned} \frac{dO}{d\mathbf{A}} &= \frac{1}{4} \frac{d \text{tr} \left[(\mathbf{A}^T - \mathbf{A})(\mathbf{A} - \mathbf{A}^T) \right]}{d\mathbf{A}} - \frac{d \text{tr} \left[(\mathbf{A}^T - \mathbf{A}) \mathbf{U}^T \xi \right]}{d\mathbf{A}} \\ &= (\mathbf{A} - \mathbf{A}^T) - (\mathbf{U}^T \xi - \xi^T \mathbf{U}) \\ &= 2\Omega_1 - (\mathbf{U}^T \xi - \xi^T \mathbf{U}). \end{aligned} \quad (\text{B.12})$$

By setting the gradient $\frac{dO}{d\mathbf{A}}$ to zero, it gives

$$\Omega_1^* = \frac{1}{2} \left(\mathbf{U}^T \boldsymbol{\xi} - \boldsymbol{\xi}^T \mathbf{U} \right) = \text{skew} \left(\mathbf{U}^T \boldsymbol{\xi} \right). \quad (\text{B.13})$$

Finally, by incorporating Eqs.(B.11,B.13), the desired orthogonal projection onto the tangent space $T_{\mathbf{U}}\boldsymbol{\xi}$ is given as

$$\begin{aligned} P_{\mathbf{U}}(\boldsymbol{\xi}) &= \mathbf{U}\Omega_1^* + \mathbf{U}_{\perp}\Omega_2^* \\ &= \mathbf{U} \text{skew} \left(\mathbf{U}^T \boldsymbol{\xi} \right) + \mathbf{U}_{\perp} \mathbf{U}_{\perp}^T \boldsymbol{\xi} \\ &= \mathbf{U} \text{skew} \left(\mathbf{U}^T \boldsymbol{\xi} \right) + \left(\mathbf{I}_{k \times k} - \mathbf{U}\mathbf{U}^T \right) \boldsymbol{\xi}. \end{aligned} \quad (\text{B.14})$$

The retraction mapping $R_{\mathbf{U}}M : T_{\mathbf{U}}M \rightarrow \mathbf{U}$, which is a mapping function from the tangent space to the manifold, induces a curve $u_{\boldsymbol{\xi}_{\mathbf{U}}} : t \rightarrow R_{\mathbf{U}}M(t\boldsymbol{\xi}_{\mathbf{U}})$ capable of characterizing a moving directed by the tangent vector $\boldsymbol{\xi}_{\mathbf{U}} \in T_{\mathbf{U}}M$ along the manifold. A retraction mapping needs to satisfy the properties of (1) $u_{\boldsymbol{\xi}_{\mathbf{U}}}(0) = \mathbf{U}$ so that the moving is initiated at the point \mathbf{U} in the manifold, and (2) $\dot{u}_{\boldsymbol{\xi}_{\mathbf{U}}}(0) = \boldsymbol{\xi}_{\mathbf{U}}$ so that the moving along this curve is in the direction of $\boldsymbol{\xi}_{\mathbf{U}}$. Every manifold that admits a Riemannian metric has a retraction mapping defined by the Riemannian exponential mapping, which however can be expensive to calculate in some cases. For an embedded manifold, it is possible to turn the computation of a retraction mapping into a procedure of first moving away from \mathbf{U} along the direction of $\boldsymbol{\xi}_{\mathbf{U}}$ to arrive at a new point $\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}$, and then projecting $\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}$ back to the manifold. Whether this procedure qualifies as a well-defined retraction mapping is determined by how to project $\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}$, which meanwhile should also be computationally efficient. In the case of Stiefel manifold, the following function qualifies as a retraction mapping [1]

$$R_{\mathbf{U}}M(\boldsymbol{\xi}_{\mathbf{U}}) = \pi_1(\phi^{-1}(\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}})), \quad (\text{B.15})$$

where $\pi_1 : M \times N \rightarrow M$ is a mapping function $\pi_1(\mathbf{F}, \mathbf{G}) = \mathbf{F}$ that returns its first input component. The function $\phi : M \times N \rightarrow \mathbb{R}_*^{k \times k}$ is a diffeomorphism mapping $\phi(\mathbf{F}, \mathbf{G}) = \mathbf{V}$, where \mathbf{F} belongs to the Stiefel manifold M , $\mathbf{V} \in \mathbb{R}_*^{k \times k}$ is from an open subset of the vector space $\mathbb{R}^{k \times k}$, \mathbf{G} is from an abstract manifold N such that $\dim(M) + \dim(N) = \dim(\mathbb{R}^{k \times k})$. There is a neutral element $\mathbf{I} \in N$ such that $\phi(\mathbf{F}, \mathbf{I}) = \mathbf{F}$.

Both QR factorization and polar decomposition result in natural definitions of the mapping ϕ . The QR factorization of the input matrix $\mathbf{V} \in \mathbb{R}_*^{k \times k}$ is expressed as $\mathbf{V} = \text{qf}(\mathbf{V})\text{rf}(\mathbf{V})$, where $\text{qf}(\mathbf{V})$ is an orthogonal matrix (referred to as the Q factor) belonging to M and $\text{rf}(\mathbf{V}) \in N$ is an upper triangular matrix (referred to as the R factor). Given $\phi_{\text{QR}}(\text{qf}(\mathbf{V}), \text{rf}(\mathbf{V})) = \mathbf{V}$ and letting $\mathbf{V} = \mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}$, the following retraction mapping is obtained¹

$$R_{\mathbf{U}}M(\boldsymbol{\xi}_{\mathbf{U}}) = \pi_1(\phi_{\text{QR}}^{-1}(\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}})) = \text{qf}(\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}), \quad (\text{B.16})$$

which is simply the Q factor of $\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}$. The polar decomposition of an input matrix $\mathbf{V} \in \mathbb{R}_*^{k \times k}$ is $\mathbf{V} = \mathbf{F}\mathbf{G}$, where $\mathbf{G} = (\mathbf{V}^T \mathbf{V})^{\frac{1}{2}} \in N$, and $\mathbf{F} = \mathbf{V}(\mathbf{V}^T \mathbf{V})^{-\frac{1}{2}}$ belonging

1. When updating the projection matrix \mathbf{U} using gradient descent, by controlling $\xi_{\mathbf{U}}$ through adjusting the step-size, it is always possible to arrive at a new point $\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}$ that is full rank and it thus belongs to an open subset of the vector space $\mathbb{R}^{k \times k}$.

to the Stiefel manifold². This defines a desired mapping $\phi_{\text{p}} \left(\mathbf{V}(\mathbf{V}^T \mathbf{V})^{-\frac{1}{2}}, (\mathbf{V}^T \mathbf{V})^{\frac{1}{2}} \right) = \mathbf{V}$. Utilizing the tangent vector condition in Eq.(B.4) and the manifold condition of $\mathbf{U}^T \mathbf{U} = \mathbf{I}_{k \times k}$, the following retraction mapping is obtained

$$\begin{aligned} R_{\mathbf{U}}M(\boldsymbol{\xi}_{\mathbf{U}}) &= \pi_1(\phi_{\text{p}}^{-1}(\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}})) \\ &= (\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}) \left((\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}})^T (\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}) \right)^{-\frac{1}{2}} \\ &= (\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}) \left(\mathbf{U}^T \mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}^T \mathbf{U} + \mathbf{U}^T \boldsymbol{\xi}_{\mathbf{U}} + \boldsymbol{\xi}_{\mathbf{U}}^T \boldsymbol{\xi}_{\mathbf{U}} \right)^{-\frac{1}{2}} \\ &= (\mathbf{U} + \boldsymbol{\xi}_{\mathbf{U}}) \left(\mathbf{I}_{k \times k} + \boldsymbol{\xi}_{\mathbf{U}}^T \boldsymbol{\xi}_{\mathbf{U}} \right)^{-\frac{1}{2}}. \end{aligned} \quad (\text{B.17})$$

APPENDIX C: COMPARATIVE EXAMPLE OF EMBEDDING CONSTRAINTS

We provide an example to illustrate the effect of the three feasible sets M , M_1 and M_2 , for the minimization of $O_{\text{KL}}^{(L)}$ only, using the Cora dataset. In this simple case, the resulting embedding just preserves the local character of the samples in the target space. Specifically, cosine similarities are used to construct the document proximities, and then a 10-nearest neighbor search is applied to construct local neighbor adjacency matrix. For M_2 , we use $\mathbf{B} = \mathbf{D}(\mathbf{W})$, to follow the traditional setup [3], [4], and we execute the actual minimizations with the aid of the Manopt toolbox [5]. The computed 2-dimensional embeddings are displayed in Fig.C.1. It can be seen that, for this example, M and M_2 provide smoother and more spread out data distributions than M_1 .

APPENDIX D: ANALYSIS OF COVA-E1, COVA-P1

The analytical solution of the COVA embedding model based on Eqs.(1,6), referred to as COVA-E1 in the manuscript, can be derived. Its minimizing objective function can be formulated as

$$O = \alpha \text{tr} \left(\mathbf{Z}^T \mathbf{D}(\mathbf{R}) \mathbf{Z} \right) - 2\alpha \text{tr} \left(\mathbf{Z}^T \mathbf{R} \mathbf{Y} \right) + 2(1 - \alpha) \mathbf{Z}^T \mathbf{L}(\mathbf{W}) \mathbf{Z}. \quad (\text{D.1})$$

By setting its Euclidean gradient equal to zero, we have

$$\alpha \mathbf{D}(\mathbf{R}) \mathbf{Z} - \alpha \mathbf{R} \mathbf{Y} + (1 - \alpha) \mathbf{L}(\mathbf{W}) \mathbf{Z} = 0. \quad (\text{D.2})$$

This results in

$$\mathbf{Z} = [\alpha \mathbf{D}(\mathbf{R}) + (1 - \alpha) \mathbf{L}(\mathbf{W}) + \zeta \mathbf{I}_{l \times l}]^{-1} \mathbf{R} \mathbf{Y}, \quad (\text{D.3})$$

where $\mathbf{I}_{l \times l}$ is the $l \times l$ identity matrix. The regularization parameter $0 < \zeta \ll 1$ is introduced to avoid matrix singularity, and it is set to zero if the matrix $\alpha \mathbf{D}(\mathbf{R}) + (1 - \alpha) \mathbf{L}(\mathbf{W})$ is invertible. The two matrices of \mathbf{R} and \mathbf{Y} are naturally full rank given independent cohorts, and thus the embedding matrix \mathbf{Z} is the multiplication of three full-rank matrices and belongs to the noncompact Stiefel manifold.

Regarding to its projection version referred to as COVA-P1 in the manuscript, the minimization objective function can be expressed as

$$\begin{aligned} \min_{\substack{\mathbf{U} \in \mathbb{R}^{k \times k}, \\ \mathbf{U}^T \mathbf{U} = \mathbf{I}_{k \times k}}} O &= \alpha \text{tr} \left(\mathbf{U}^T \mathbf{F}^T \mathbf{D}(\mathbf{R}) \mathbf{F} \mathbf{U} \right) - 2\alpha \text{tr} \left(\mathbf{U}^T \mathbf{F}^T \mathbf{R} \mathbf{Y} \right) \\ &+ (1 - \alpha) \text{tr} \left(\mathbf{U}^T \mathbf{F}^T \mathbf{L}(\mathbf{W}) \mathbf{F} \mathbf{U} \right). \end{aligned} \quad (\text{D.4})$$

2. Letting $\mathbf{V} = \mathbf{W}\boldsymbol{\Sigma}\mathbf{P}^T$ denote the singular value decomposition, we have $\mathbf{G} = (\mathbf{V}^T \mathbf{V})^{\frac{1}{2}} = \mathbf{P}\boldsymbol{\Sigma}\mathbf{P}^T$, $\mathbf{F} = \mathbf{V}(\mathbf{V}^T \mathbf{V})^{-\frac{1}{2}} = \mathbf{W}\mathbf{P}^T$, $\mathbf{F}^T \mathbf{F} = \mathbf{I}_{k \times k} \in M$.

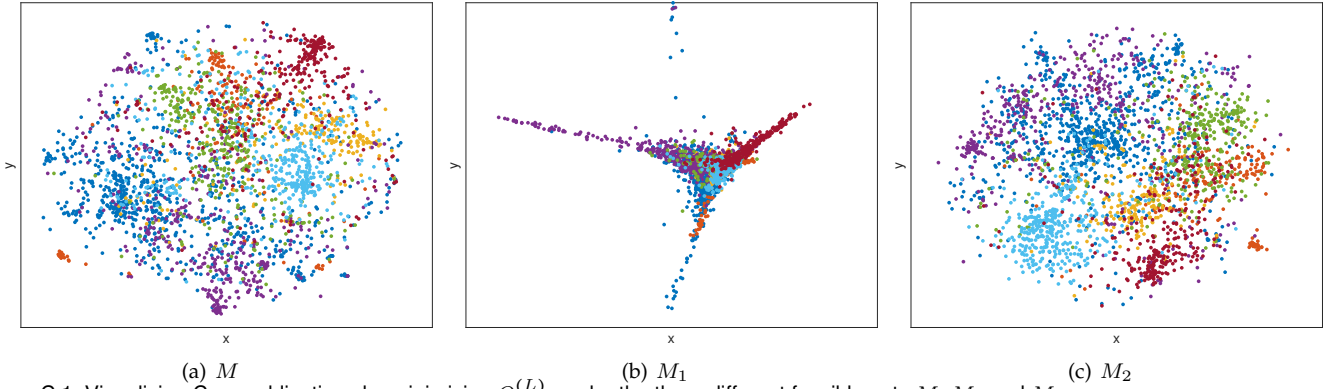


Figure C.1. Visualizing Cora publications by minimizing $O_{\text{KL}}^{(L)}$, under the three different feasible sets M , M_1 and M_2 .

By introducing a symmetric Lagrange multiplier matrix $\Theta \in \mathbb{R}^{k \times k}$, stationary values of the following Lagrange function are sought

$$L = O + \text{tr} \left(\Theta \left(\mathbf{U}^T \mathbf{U} - \mathbf{I}_{k \times k} \right) \right). \quad (\text{D.5})$$

Its gradient with respect to \mathbf{U} is given as

$$\frac{\partial L}{\partial \mathbf{U}} = 2\mathbf{F}^T [\alpha \mathbf{D}(\mathbf{R}) + (1 - \alpha)\mathbf{L}(\mathbf{W})] \mathbf{F} \mathbf{U} + \mathbf{U} (\Theta + \Theta^T) - 2\alpha \mathbf{F}^T \mathbf{R} \mathbf{Y}. \quad (\text{D.6})$$

Letting $\mathbf{A}_1 = 2\mathbf{F}^T [\alpha \mathbf{D}(\mathbf{R}) + (1 - \alpha)\mathbf{L}(\mathbf{W})] \mathbf{F}$ and $\mathbf{A}_2 = 2\alpha \mathbf{F}^T \mathbf{R} \mathbf{Y}$ and setting the gradient equal to zero, we have

$$\mathbf{A}_1 \mathbf{U} + \mathbf{U} (\Theta + \Theta^T) = \mathbf{A}_2. \quad (\text{D.7})$$

This resembles a constrained version of Sylvester equation by orthogonality. This, together with the undetermined Lagrange multiplier matrix Θ , does not leave us an easy analytical solution. Alternatively, a gradient descent approach operating over the Stiefel manifold through the use of Riemannian gradient and retraction mapping is used to optimize the projection matrix.

APPENDIX E: COMPLEXITY ANALYSIS

Some COVA models require iterative embedding optimization (E2-4), some iterative projection optimization (P1-4), and some have analytical solutions (E1 and eigen-COVA). The existing iterative embedding techniques (e.g., t-SNE and SNE) usually have a computational complexity around $O(l^2)$ and the existing iterative projection ones (e.g., NCA and MCML) around $O(l^2 d^2)$. Compared to their equivalent existing types, E2-4 and P1-4 have a moderately increased complexity that is linear in the sample size. E1 has comparable complexity to many existing spectral embedding techniques (e.g., LE and LLE), and eigen-COVA has a particularly low complexity due to its use of prototype and projection based feature transformation. In the following, we provide a detailed complexity analysis for different COVA models based on their model input preparation and optimization.

Construction of the neighbor adjacency matrix \mathbf{W} is a common process required by both COVA and typical visualization techniques. Its computational complexity is of $O(l^2)$ in the sample size l [6]. Additionally, COVA requires to construct the cohort confidence matrix \mathbf{R} and cohort prototype matrix \mathbf{Y} . The complexity for computing \mathbf{R} is of

$O(lc)$, and for \mathbf{Y} is quadratic (or cubic) in the cohort size c depending on the used reconstruction technique. COVA projection also requires to prepare a fourth input of relation feature matrix \mathbf{F} , which is of $O(l\tilde{k})$. Because c is in general small and the sample prototype size \tilde{k} can be set to a reasonably small value, both $O(lc)$ and $O(l\tilde{k})$ correspond to computationally efficient operations that mainly depend on the sample size l .

Operating on the prepared input, COVA-E1 and eigen-COVA offer analytical formulations of the optimal solution. Their complexity is dominated by computing the inverse of an $l \times l$ matrix for COVA-E1, and computing a single eigenvector for a total of k matrices of size $(\tilde{k} + k) \times (\tilde{k} + k)$ for eigen-COVA. Therefore, the complexity of computing COVA-E1 solution is of $O(l^2 \log l)$, which is comparable to those spectral embedding techniques based on eigen-decomposition of a square matrix of sample size that is $O(l^3)$. The complexity for computing the eigen-COVA solution is of $O(k(\tilde{k} + k)^3)$, as k decompositions are needed. This is low because for visualization purposes k is typically 2 or 3 and the size of sample prototypes \tilde{k} is usually reasonably small.

The iterative COVA models perform gradient descent optimization over a matrix manifold. Complexity for computing their Euclidean gradient in each iteration is $O(l^2 + la)$ for E2-4, where a is c^2 or ck depending on the employed version, and also $O(l^2 \tilde{k} + lb)$ for P1,4 and $O(l^2 \tilde{k}^2 + lb)$ for P2,3, where b is the product between combinations of k , \tilde{k} , c depending on the model versions. Compared to existing iterative embedding techniques, such as t-SNE and SNE, which compute their gradient in each iteration with a complexity of $O(l^2)$, the added complexity of E2-4 is linear in the sample size, which is moderate due to the small cohort size c and output dimensionality k . Compared to existing iterative projection techniques, such as NCA and MCML with a complexity of $O(l^2 d^2)$, the use of prototype based relation features in P1-4 reduces the complexity when $\tilde{k} \ll d$. When d and \tilde{k} are of similar scale, the increased complexity of P1,4 compared to existing ones becomes $O(lb)$ which is linear in the sample size.

In addition to the Euclidean gradient computation, optimization of E2-4 and P1-4 requires extra cost to convert the Euclidean gradient to the Riemannian gradient and also extra cost to project the updated solution back to the manifold through retraction mapping at each iteration. For E2-

Table F.1
Pseudo-code of E2-4 models for learning COVA embeddings.

-
- 1) **Input:** $l \times d$ data matrix \mathbf{X} , $l \times c$ cohort label matrix \mathbf{Y} , output dimension k , COVA parameter α , iteration number N , rank checking condition number N_c .
 - 2) **Model input preparation:**
 - a) Compute $l \times l$ weight matrix \mathbf{W} and $l \times c$ cohort confidence matrix \mathbf{R} from \mathbf{X} and \mathbf{Y} .
 - b) Compute $c \times k$ cohort prototype matrix $\mathbf{\Upsilon}$ from either \mathbf{X} or external information (Section 3.1.1).
 - 3) **Model initialization:** Set embedding matrix \mathbf{Z}_0 according to a chosen initialization scheme, and $i = 0$.
 - 4) **Model optimization:** While $i < N$:
 - a) Compute Riemannian gradient $\text{grad}O(\mathbf{Z}_i)$ that is equal to Euclidean gradient $\text{Grad}O(\mathbf{Z}_i)$ (Table A.1).
 - b) Perform standard line search to find step size γ .
 - c) Update embedding by

$$\mathbf{Z}_{i+1} = \begin{cases} R_{\mathbf{Z}_i} M(-\gamma \text{grad}O(\mathbf{Z}_i)) & \text{if } \tau = 1, \\ -\gamma \text{grad}O(\mathbf{Z}_i), & \text{otherwise,} \end{cases}$$
 where the control variable $\tau \in \{0, 1\}$ is set to perform retraction mapping every N_c iteration and in the last iteration.
 - d) Set $i \leftarrow i + 1$.
 - 5) **Output:** Return embedding matrix \mathbf{Z}_N .
-

4, the Euclidean and Riemannian gradients are equal in the noncompact Stiefel manifold. Therefore, the increased complexity is only caused by the retraction mapping, which needs a singular value decomposition of the updated $l \times k$ embedding matrix with a complexity of $O(lk^2)$. For P1-4, the gradient conversion and retraction mapping are implemented based on Eq.(B.14) and Eq.(B.16 or B.17), of which the overall complexity is around $O(\tilde{k}^2k + \tilde{k}k^2 + k^3)$ and it is reasonably low due to the small values of \tilde{k} and k .

APPENDIX F: PSEUDO-CODE

Among different COVA models, E1 has an analytical solution which can be directly computed by Eq.(15), E2-4 and P1-4 are optimized iteratively by performing gradient descent over a matrix manifold with standard line search, with iterations terminating after reaching the maximum iteration number or when the norm of the Riemannian gradient is smaller than a predefined tolerance. The implementation of P1-4 is based on stochastic gradient descent, which uses a subset of data samples to estimate the gradient (so-called batch training), to speed up the gradient computation. Projections of eigen-COVA are computed based on eigen-decomposition of small-size matrices and sequential orthogonalization procedure. We provide pseudo-code for E2-4 in Table F.1, for P1-4 in Table F.2, and for eigen-COVA in Table F.3.

With regard to the initialization for E2-4 and P1-4, apart from a random initialization, we suggest two alternative schemes for E2-E4. One is to start from the eigenvectors of the Laplacian matrix of the local weights \mathbf{W} , that correspond to the smallest nonzero eigenvalues. These initial embeddings preserve the local neighborhood structure indicated by \mathbf{W} . Subsequently, the gradient descent update will modify the embedding to assume the desired global cohort arrangement. Another option is to initially restrict the

Table F.2
Pseudo-code of P1-4 models for batch training COVA projections.

-
- 1) **Input:** $l \times d$ data matrix \mathbf{X} , $l \times c$ cohort label matrix \mathbf{Y} , prototype sample size \tilde{k} , output dimension k , COVA parameter α , iteration number N , batch size l_{batch} , cycle number T .
 - 2) **Model input preparation:**
 - a) Compute $l \times l$ weight matrix \mathbf{W} and $l \times c$ cohort confidence matrix \mathbf{R} from \mathbf{X} and \mathbf{Y} .
 - b) Compute $l \times k$ relation feature matrix \mathbf{F} from \mathbf{X} (and \mathbf{Y} if necessary).
 - c) Compute $c \times k$ cohort prototype matrix $\mathbf{\Upsilon}$ from either \mathbf{X} or external information (Section 3.1.1).
 - 3) **Model initialization:** Set projection matrix \mathbf{U}_0 according to a chosen initialization scheme, and $j = 1$.
 - 4) **Model optimization:**
 - 1: **while** $j \leq T$ **do**
 - 2: Shuffle all data samples and set $l_{\text{used}} = 0$.
 - 3: **while** $l_{\text{used}} < l$ **do**
 - 4: Set $i = 0$, and extract sample batch with indices

$$I = l_{\text{used}} + 1 : \min(l_{\text{used}} + l_{\text{batch}}, l).$$
 - 5: **while** $i < N$ **do**
 - 6: Batch estimation of $\text{Grad}O(\mathbf{U}_i)$ (Table A.2).
 - 7: Compute Riemannian gradient by Eq.(B.14):

$$\text{grad}O(\mathbf{U}_i) = P_{\mathbf{U}_i}(\text{Grad}O(\mathbf{U}_i)).$$
 - 8: Perform standard line search to find step size γ .
 - 9: Update projection by Eq.(B.16 or B.17):

$$\mathbf{U}_{i+1} = R_{\mathbf{U}_i} M(-\gamma \text{grad}O(\mathbf{U}_i)).$$
 - 10: Set $i \leftarrow i + 1$.
 - 11: **end while**
 - 12: Set $\mathbf{U}_0 \leftarrow \mathbf{U}_N$, $l_{\text{used}} \leftarrow l_{\text{used}} + l_{\text{batch}}$.
 - 13: **end while**
 - 14: Proceed to the next training cycle, $j \leftarrow j + 1$.
 - 15: **end while**
 - 5) **Output:** Return projection matrix \mathbf{U}_N .
-

embedded patterns of all the samples from the i th cohort, to coincide with its prototype c_i . This is equivalent to forcing all the original samples to gather at one fixed location determined by their cohort prototype. Subsequently, the descent update will spread the patterns to assume the desired local data structure. For P1-P4, apart from random initialization, the optimization can also be initialized with the eigenvectors of $\mathbf{F}^T \mathbf{L}(\mathbf{W}) \mathbf{F}$ corresponding to the smallest eigenvalues. According to orthogonal locality preserving projection [7], this initial solution linearly projects the relation features \mathbf{F} to an embedding space where the local neighborhood structure in \mathbf{W} is preserved. The gradient descent will then gradually update the projection matrix, so that the desired global cohort distribution is brought into consideration.

APPENDIX G: MEASURES FOR VISUALIZATION ASSESSMENT

To evaluate the visualization output with respect to the cohort separation, cohort relevant positioning and sample local neighboring patterns for improved visualization, we make use of three score functions. The first one, denoted by S_s , measures the level of separability between cohorts expected to be shown in the target space. One direct way for quantifying this, is via the one-nearest-neighbor classification rate that examines the compatibility between the

Table F.3
Pseudo-code of eigen-COVA for computing COVA projections.

1) **Input:** $l \times d$ data matrix \mathbf{X} , $l \times c$ cohort label matrix \mathbf{Y} , prototype sample size \tilde{k} , output dimension k , COVA parameter α and η .

2) **Model input preparation:**

- Compute $l \times l$ weight matrix \mathbf{W} and $l \times c$ cohort confidence matrix \mathbf{R} from \mathbf{X} and \mathbf{Y} .
- Compute $l \times \tilde{k}$ relation feature matrix \mathbf{F} from \mathbf{X} (and \mathbf{Y} if necessary).
- Compute $c \times k$ cohort prototype matrix $\mathbf{\Upsilon}$ from either \mathbf{X} or external information (Section 3.1.1).

3) **Projection Computation:**

- Set $t = 1$, $\mathbf{F}_1 = \mathbf{F}$ and $\mathbf{\Upsilon}_1 = \mathbf{\Upsilon}$.
- while** $t \leq k$ **do**
- Compute matrix \mathbf{O}_{eig} using \mathbf{F}_t and $\mathbf{\Upsilon}_t$ by Eq.(25).
- Compute \mathbf{O}_{eig} 's eigenvector \mathbf{p} with smallest eigenvalue

$$\mathbf{p} = \begin{bmatrix} \mathbf{u}^{(t)} \\ \mathbf{v}^{(t)} \end{bmatrix}.$$
- Update feature and cohort prototype matrices by

$$\mathbf{F}^{(t+1)} = \mathbf{F}^{(t)} \left(\mathbf{I}_{\tilde{k} \times \tilde{k}} - \frac{\mathbf{u}^{(t)} \mathbf{u}^{(t)T}}{\mathbf{u}^{(t)T} \mathbf{u}^{(t)}} \right),$$

$$\mathbf{\Upsilon}^{(t+1)} = \mathbf{\Upsilon}^{(t)} \left(\mathbf{I}_{k \times k} - \frac{\mathbf{v}^{(t)} \mathbf{v}^{(t)T}}{\mathbf{v}^{(t)T} \mathbf{v}^{(t)}} \right).$$
- Set $t \leftarrow t + 1$.
- end while**

4) **Output:** Return projection matrix $\mathbf{U} = [\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)}]$.

learned embedding and the cohort memberships. A higher classification indicates a stronger separability between cohorts.

The second score, denoted by S_c , checks whether the data samples from each cohort are distributed around the location indicated by its cohort prototype. It relies on the classification rate using the cohort prototypes $\{\mathbf{c}_i\}_{i=1}^c$ (or $\{\mathbf{e}_i\}_{i=1}^c$ for eigen-COVA) as the training set and the embedded samples as the test set. A higher rate indicates that more samples are distributed closer to the correct prototypes of their own cohorts, as this demonstrates a better preservation of the desired cohort arrangement.

Sometimes, when the data cohort structure of interest is complex, it can be difficult for the algorithm to locate a cohort in an exact position pointed by the prototype. However, such an exact matching can be unnecessary as long as a desired relevant positioning structure of cohorts is maintained, which can be assessed by directly examining the proximity profiles between cohorts. Let the two $c \times c$ binary matrices $\Delta_c^{(v)}$ and $\Delta_c^{(d)}$ store the visualized and desired neighbor adjacency links between cohorts, where the ij th element indicates if the i th cohort is in the κ -nearest neighbors of the j th cohort or vice-versa. An alternative score

$$S_c^{(r)} = \frac{\mathbf{1}_c^T \left(\Delta_c^{(v)} \circ \Delta_c^{(d)} \right) \mathbf{1}_c}{\mathbf{1}_c^T \Delta_c^{(d)} \mathbf{1}_c} \quad (\text{G.1})$$

can be computed. It represents the ratio between the number of correctly preserved cohort neighbor links ($\Delta_c^{(v)} \circ \Delta_c^{(d)}$) in the target space and the number of desired links $\Delta_c^{(d)}$. For instance, $\Delta_c^{(v)}$ can be constructed by comparing Euclidean

distances between the cohort centers in the target space, and $\Delta_c^{(d)}$ by comparing Euclidean distances between the given cohort prototypes. Between the two scores of S_c and $S_c^{(r)}$, S_c examines the cohort arrangement by looking at the relevant positioning between individual samples and the exact cohort prototypes, while $S_c^{(r)}$ does so in a more abstract manner by looking at the average closeness between cohorts and focusing on their neighboring relationships.

Finally, the local neighborhood of each individual data sample is expected to be preserved. Since the relevant distances between cohorts are reorganized based on the pre-defined cohort prototypes, it is not meaningful to consider the inter-cohort neighbors when comparing the original and embedded data. Only the local neighbor structure within each cohort is examined. We use $N(\mathbf{x}, \kappa)$ to denote the index set of the samples that are the κ nearest intra-cohort neighbors to \mathbf{x} (excluding self) searched in the original feature space. We then compare the compatibility between the neighbors in the original and embedding spaces for each data sample, by calculating the ratio $\frac{|N(\mathbf{x}_i, \kappa) \cap N(\mathbf{z}_i, \kappa)|}{\kappa}$. Its average over all the data samples is

$$S_n(\kappa) = \frac{1}{l\kappa} \sum_{i=1}^l |N(\mathbf{x}_i, \kappa) \cap N(\mathbf{z}_i, \kappa)|. \quad (\text{G.2})$$

To make this score insensitive to local scales, we finally use its average $S_n = \frac{1}{\tilde{\kappa}} \sum_{\kappa=1}^{\tilde{\kappa}} S_n(\kappa)$, over the multiple neighborhood sizes (up to $\tilde{\kappa}$ defined later).

APPENDIX H: ADDITIONAL EXPERIMENTS AND RESULTS

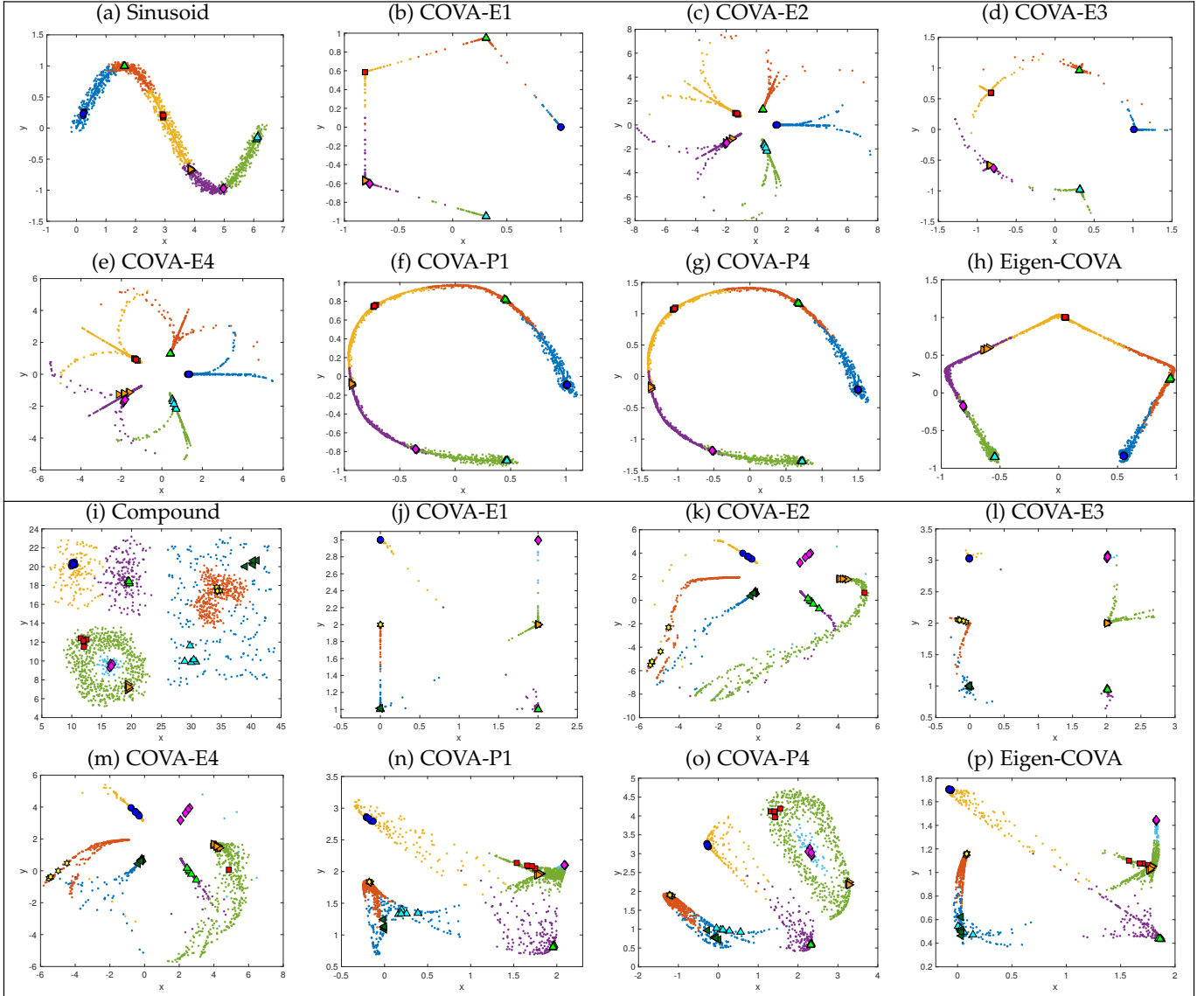
H.1 2D AND 3D SYNTHETIC DATA

When experimenting with synthetic datasets, Euclidean distances are used for all models to compute the local weight matrix \mathbf{W} , the cohort membership matrix \mathbf{R} and the relation feature matrix \mathbf{F} . Specifically, the weight matrix is computed as $\mathbf{W} = \mathbf{S} \circ \mathbf{N}(\mathbf{S}, 50)$, where the similarity matrix \mathbf{S} between the samples is taken from the Euclidean by subtracting it from its maximum matrix element, and the cohort membership matrix is computed via Eq.(12). For the projection models, the relation features \mathbf{F} are computed using $k=500$ randomly chosen sample prototypes.

We demonstrate COVA's capability of cohort control using the two 2D synthetic datasets Sinusoid and Compound [8], plotted in Table H.1(a,i). The goal is to show that locations of data cohorts can be fully controlled by predefined cohort prototypes through COVA. The distributions of the predefined cohort prototypes are illustrated in Table H.2. They are chosen to present cohort arrangements that are very different from the original ones in order to demonstrate the cohort control effectively. The COVA projection models are demonstrated using P1, P4 and eigen-COVA, as we have observed that P2 and P3 provide very similar performance to P1 and P4 for both datasets. Performance is reported by setting the parameter α to a high value of 0.95, to focus more on cohort arrangement than local neighborhood preservation, and the parameter η for eigen-COVA only is fixed to 0.8. In order to inspect the neighbor preservations, we randomly choose several example points from different cohorts and identify their four nearest neighbors using

Table H.1

Illustration of the COVA models using the Sinusoid (top half) and Compound (bottom half) datasets. Cohort locations are controlled by the manually defined prototypes shown in Table H.2. Different cohorts correspond to different shadings (and these also correspond to the Table H.2 prototypes). Example sample points randomly selected from different cohorts, together with their four nearest neighbors, are also highlighted in both original and target spaces, using the same shading and marker (\square , \triangle , or \diamond , etc.).



Euclidean distances in the original space. We finally mark them in both the original and the target spaces as shown Table H.1.

It can be seen from Table H.1 that, for these two datasets, COVA models E1 and E3 provide more forceful control of the cohort distribution than models E2 and E4, leading to more exact matches towards the cohort arrangements predefined in Table H.2, and tighter sample distributions within each cohort. For instance, the points marked by \square and \triangleright from the same circularly shaped cohort of Table H.1(i), appear superimposed in the middle-right cohort of Table H.1(j) generated by E1. This can be justified by the relatively high value of α , which prioritizes rule 1 through the objective $O_{\text{dist}}^{(1)}$ used by both E1 and E3. The COVA models E2 and E4, on the other hand, employ $O_{\text{KL}}^{(1)}$ with softer behavior for rule 1. Moreover, comparing E1 and E3, as can be seen in Table H.1(d,l), E3 generates slightly less

concentrated cohorts than E1 in Table H.1(b,j). This is due to E3 using KL divergence to build the objective $O_{\text{KL}}^{(2)}$ for the preservation of local neighborhoods. In general, from the above observations and as discussed in Section 3.1.3, we observe that the distance error objectives are more forceful than the divergence ones in terms of data shaping control. Although E1 and E3 create tighter sample distributions within each cohort than E2, E2 possesses slightly higher S_c score in the Sinusoid case (see Table H.2). This is caused by several individual samples that are misplaced in a wrong cohort around the boundary areas by the model. It can be seen from Fig.H.1, that E1 and E3 generate more such samples than E2, which consequently lower slightly their S_c scores. When only focusing on the neighboring profiles of the cohorts by using $S_c^{(r)}$ to assess the cohort arrangement, E1-3 possess the same score of $S_c^{(r)}=100\%$ (the effective neighbor number is set as two). In this case, errors caused

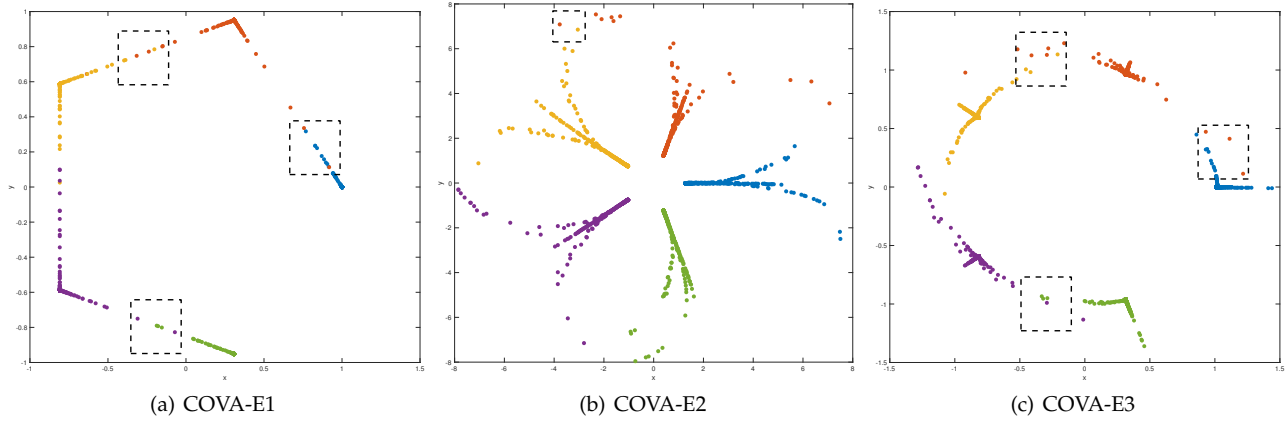
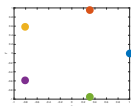
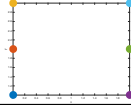


Figure H.1. Illustration of misplaced boundary points highlighted in boxes.

Table H.2

Comparison between the COVA models using the synthetic Sinusoid and Compound datasets, and the measures S_n , S_c and S_s . The first column illustrates the corresponding 2D cohort prototypes explicitly created for the purpose of the demonstration.

Sinusoid	COVA-E1	COVA-E2	COVA-E3	COVA-E4	COVA-P1	COVA-P4	Eigen-COVA
	S_n 70.1%	51.7%	59.3%	53.1%	94.2%	94.3%	85.1%
	S_c 99.4%	99.8%	99.0%	98.1%	94.4%	94.4%	94.8%
	S_s 99.4%	99.9%	99.3%	98.1%	94.3%	94.0%	94.4%
Compound	COVA-E1	COVA-E2	COVA-E3	COVA-E4	COVA-P1	COVA-P4	Eigen-COVA
	S_n 39.0%	25.0%	32.1%	26.7%	58.9%	65.2%	41.9%
	S_c 99.4%	75.4%	99.2%	69.9%	89.2%	65.6%	96.3%
	S_s 99.6%	80.5%	99.3%	88.5%	88.1%	77.8%	96.7%

by a few misplaced individual samples are ignored.

As evidenced by the higher S_n scores for both datasets in Table H.2, all projection models provide better local neighbor control than the embedding ones. Between the Sinusoid and the Compound sets, it is more challenging to map the Compound data, because its cohort structure in the original space is less in tune with the structure of the predefined prototypes. In this case, model P4 in Table H.1(o), fails to produce the desired cohort arrangement, since it uses $O_{KL}^{(1)}$ with less forceful cohort control. However, as seen in Table H.1(f,g,h), for the simpler Sinusoid data, P4 is as successful as the other projection models.

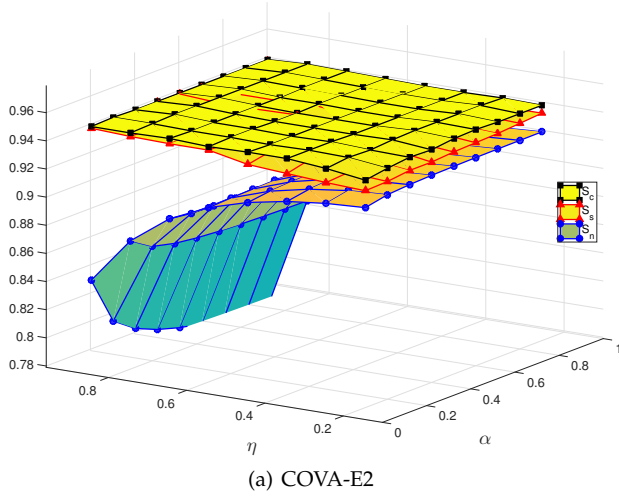
So far, we have fixed the COVA parameters to $\alpha=0.95$, and $\eta=0.8$. To explore possible intrinsic trends on how these parameters affect the visualization output, we experiment with different values of α (ranging from 0 to 1 with a step-size of 0.1) for COVA-E2 and different value combinations of (α, η) (each ranging from 0.1 to 0.9 with a step-size of 0.1) for eigen-COVA using Sinusoid data. The changes in the S_n , S_c and S_s scores versus different parameter settings are plotted in Fig.H.2 and Fig.H.3, where we also display the visualization output for several example parameter settings. In the case of COVA-E2, similar observations are made to those using Places2 and Cora data. As α increases, stronger cohort arrangement control is enforced, traded off by the reduced local sample neighbor control. For the eigen-COVA case with Sinusoid data, the performance is less sensitive to α than to η . As η increases, local neighbor control weakens.

Fig.H.4 displays more examples of COVA embeddings

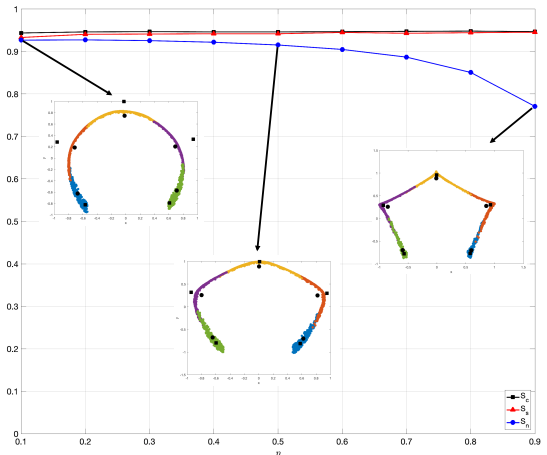
computed for the 3D Cylinder2 dataset in addition to those shown in Fig.2 of Section 4.1. The two figures together suggest the order $E1>E4>E3>E2$ in terms of their cohort control strength, which is slightly different from the 2D synthetic data case. When the same setting of $\alpha=0.3$ is used, by moving from E3, E4 to E1, tighter cohorts are obtained. In order to obtain equally tight cohorts as E1, E3 needs to have its setting increased to $\alpha = 0.6$. Amongst all methods, E2 generates the least tight cohorts when using the highest α . With regard to local neighbor preservation, E2 is the best for this data.

H.2 PLACES2 IMAGES

In this Appendix, additional experiments and results using Places2 images are presented. Fig.H.5 displays performance changes of the three COVA embedding models E1-3 for varying numbers of labeled images, where the performance of S-t-SNE ($\lambda = 0.7$) is used as the baseline. Similar observations to those based on Fig.5(a) in Section 4.2.1 can be made. Fig.H.6 and Fig.H.7(a) visually compare the output produced by various existing techniques. Contrary to existing methods generating arbitrary cohort layouts, COVA produces output with more controllable cohort arrangements. The two embedding techniques of COVA and t-SNE are compared in Fig.H.8 in a more challenging task for images from 80 different classes. There is some similarity between the arrangements of some cohorts from t-SNE and COVA; e.g., cohorts 3, 66 and 29 are positioned close to each other by both. However, there is also significant difference



(a) COVA-E2



(b) Eigen-COVA

Figure H.2. Changes of S_n , S_c and S_s scores versus different settings of Eigen-COVA weights α and η evaluated using Sinusoid data. Several examples of visualization are also demonstrated in (b) for different values of η with $\alpha = 0.9$, where the projected prototypes and the learned cohort centers are marked by “□” and “○”, respectively.

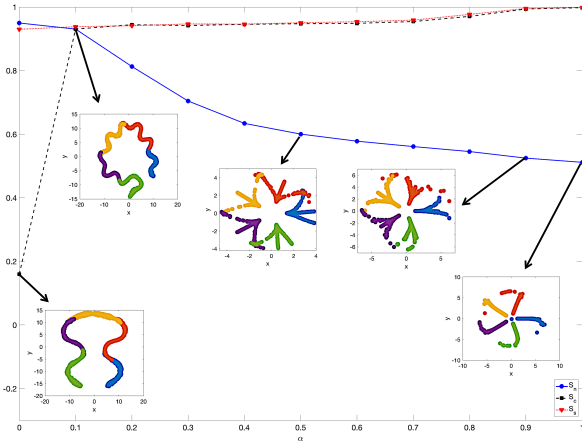


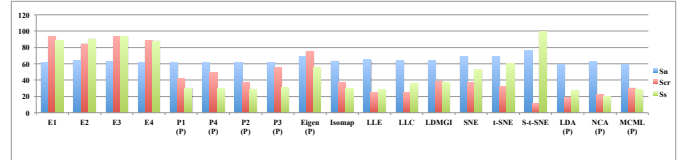
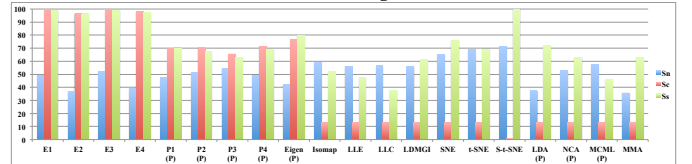
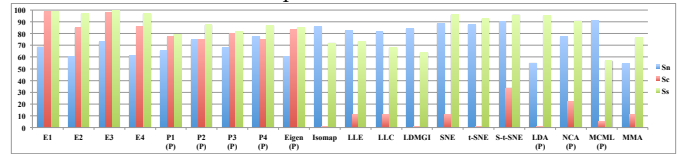
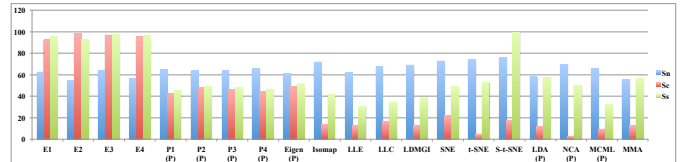
Figure H.3. Changes of S_n , S_c and S_s scores versus different settings of COVA weight α using COVA-E2 for Sinusoid data. Several examples of visualization output are also demonstrated for different parameters.

between their output; e.g., the three cohorts 59, 8 and 50 are placed by t-SNE in completely different positions from those estimated from the original space (as indicated by the prototypes).

We investigate the effect of α and η for COVA projection models. P1-P4 rely on a single parameter α , and they follow very similar pattern when α changes, for which the output of P1 is illustrated in Fig.H.9(a) as an example. It can be seen that its performance is much less sensitive to α compared to the embedding models shown in Fig.6, indicated by almost zero change in $S_c^{(r)}$ and S_n scores and a mild change in S_s score. The effect of α and η is jointly investigated for eigen-COVA with results shown in Fig.H.9(b). Eigen-COVA is less sensitive to α than COVA embedding models E1-E4, but is more sensitive to α than P1-P4. Higher values of α and η could lead to stronger cohort positioning control but this can be associated with a slight drop in local neighbor preservation score. When η is high, sometimes there can exist some output with very low S_s score and peculiar

Table H.3

Numerical comparison of different algorithms and datasets based on measures defined in Appendix G. Projection models are marked by (P). For datasets with comparatively small number of data cohorts, S_n is used, measuring an exact cohort match between individual samples and the used prototypes. For datasets with larger cohort number, $S_c^{(r)}$ (shown as S_{cr} for scene images) is used, measuring an approximated neighbor match between cohorts.

Places2 scene images, $c = 30$ Cora publications, $c = 7$ Clinical trials, $c = 9$ Flickr Images, $c = 8$

cohort shapes; an example is shown in Fig.H.9(d).

We observe how COVA projection methods perform given varying number of training samples. The results are shown in Figs.H.10(a),10(b) using P1 and P4 as examples (other COVA models perform similarly), visualizing 10 randomly selected scene classes. Using 10% of the given

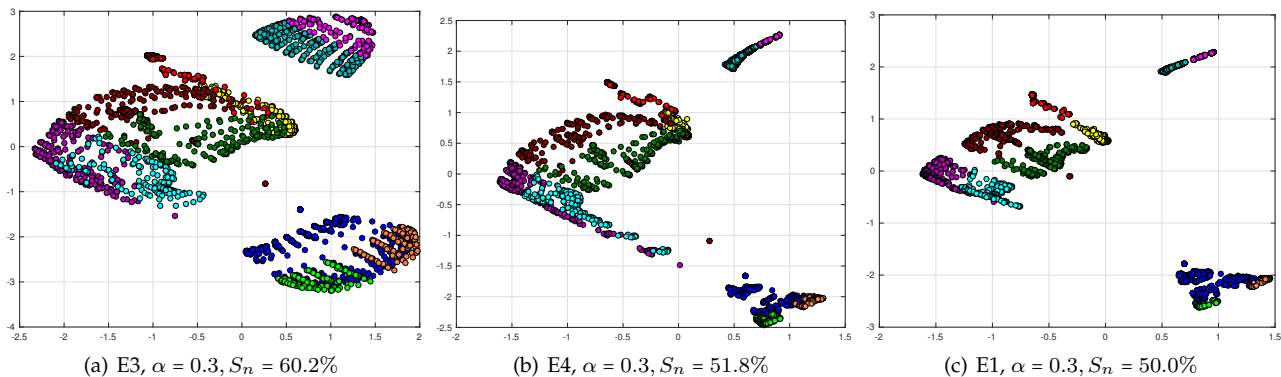


Figure H.4. Illustration of COVA embeddings computed for Cylinder2 data.

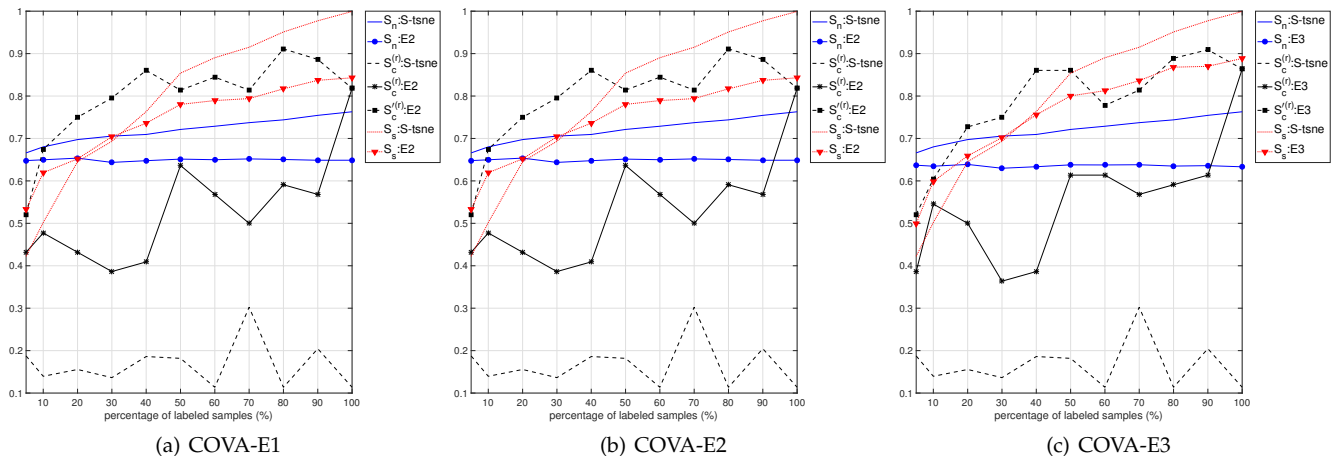


Figure H.5. Demonstration of performance change for COVA E1-3 and S-t-SNE against varying percentages of labeled samples using 30 classes of Places2 images.

images for training, COVA-P4 output for the remaining images are displayed in Fig.H.10(c). It can be seen from Figs.H.10(a),10(b) that both cohort separation and positioning scores vary within a reasonable range, which implies good generalization for COVA. The S_n score becomes higher when there are more training images. This could be related to the reduced number of image pairs being examined, caused by increasing N_{tr} and decreasing N_{te} .

In the first row of Table H.3, we quantitatively compare COVA and existing methods based on their S_n , $S_c^{(r)}$ and S_s scores. It is apparent that the existing methods do not possess the ability to control cohort locations, as manifested by the low $S_c^{(r)}$ scores. It can be seen that the S_s scores for the COVA models are amongst the top within both the embedding and projection categories. Meanwhile, the COVA models also achieve reasonably good S_n scores showing good neighbor preservation. It has to be noted, that since the COVA models are designed to simultaneously achieve the three goals, it is not realistic that they offer the highest scores for all three measures compared to those methods that only focus on one or two objectives.

Finally, we empirically compare the computational times between our implementations of COVA and existing implementations of three representative techniques of t-SNE, SNE

and NCA³. These three are considered to be the state-of-the-art in the visualization field and cover both embedding and projection techniques. Results are produced by running MATLAB 2017a on a Mac with 4GHz Intel Core i7. COVA projection models P1-4 are accelerated by stochastic gradient estimation and partial parallel implementation using the multiple cores. Fig.H.11 illustrates the resulting computing times for increasing number of samples (from 500 to 7,000) processing 4,096 data dimensions. It can be seen from Fig.H.11(a) that eigen-COVA is extremely fast and t-SNE is the second fastest. The three COVA models of E1, E2 and P1 possess comparable computing time to SNE, while P2 and P4 are more time consuming. Compared to the existing NCA implementation, all of the COVA implementations are much faster. We have also run experiments for MCML using its existing implementation, which however, is even slower than NCA and therefore we do not include it in the figure.

H.3 CORA PUBLICATIONS

For the Cora dataset, we use the $2,708 \times 1,433$ binary matrix \tilde{X} to store the word presences, the $2,708 \times 2,708$ binary matrix $\Theta = [\theta_{ij}]$ for the citation links, and the $2,708 \times 7$ binary matrix Y as the class indicators. In order to have better data

3. Their implementations are downloaded from <https://lvdmaaten.github.io/drtoolbox>, and the same for MCML.

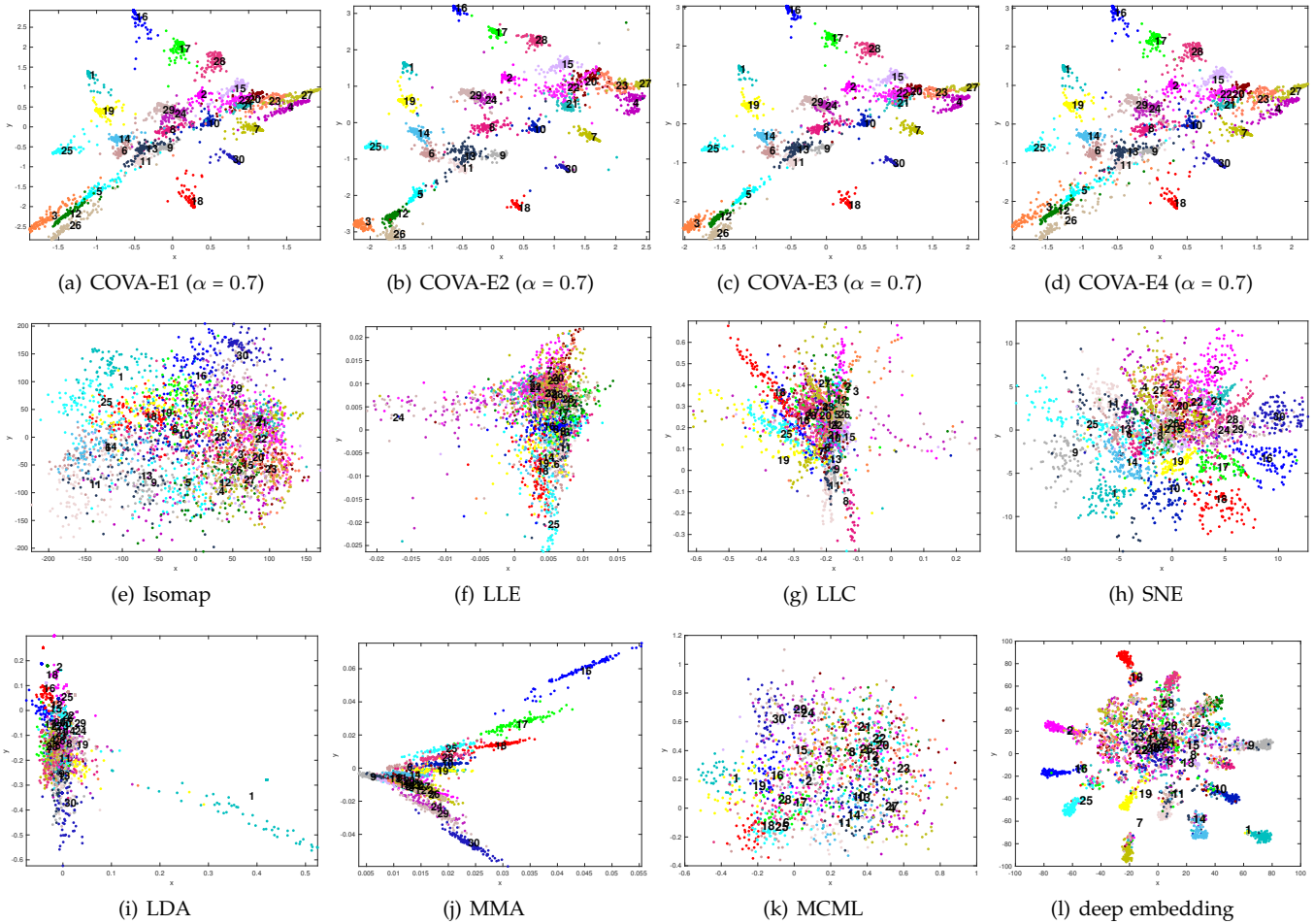


Figure H.6. Visual comparison between COVA and existing embedding methods using 30 classes of Places2 images. Different cohorts are numbered and correspond to different shadings.

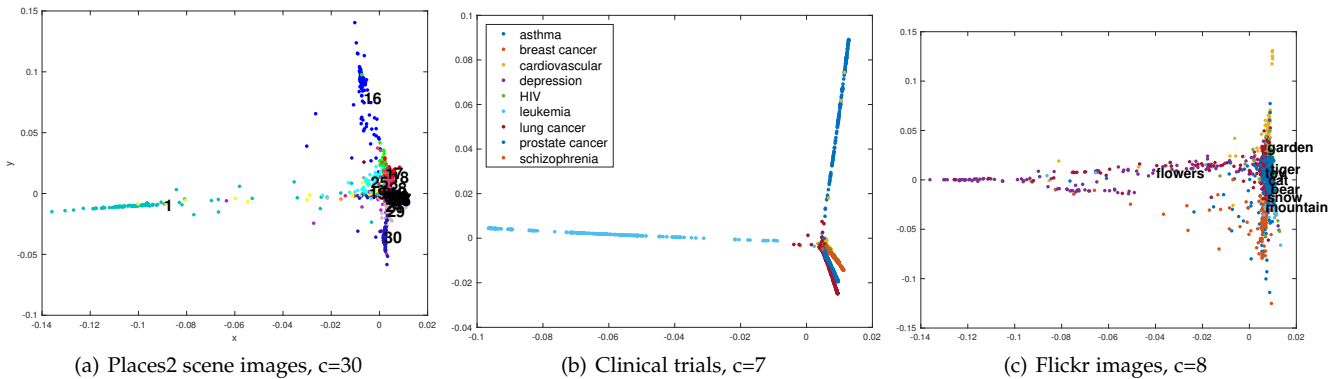


Figure H.7. Visualization output generated by the existing unsupervised embedding method LDMGI for different datasets.

utilization, we enrich the word features with citation information and compact them to derive the infused features. Specifically, the $2,708 \times 2,708$ similarity matrix $\tilde{\mathbf{S}} = \mathbf{S} \circ \mathbf{N}(\mathbf{S}, 30)$ is constructed, with \mathbf{S} computed from $\tilde{\mathbf{X}}$ using cosine similarities. The compacted word features $\tilde{\mathbf{S}}\mathbf{Y}\mathbf{M}^{-1}$ reflect the frequency accumulation of the co-occurring words between a publication and its neighbors from different classes. Similarly, compacted citation features are computed as $\Theta\mathbf{Y}\mathbf{M}^{-1}$, reflecting the citation count accumulation between a publication and its cited ones from different classes. The combined 14-dimensional features $\mathbf{X} = [\tilde{\mathbf{S}}\mathbf{Y}\mathbf{M}^{-1}, \Theta\mathbf{Y}\mathbf{M}^{-1}]$ are

used as the algorithm input.

Fig.H.12 displays visualization output of other COVA models in addition to those shown in Fig.9. These are compared in Fig.H.13 with existing algorithms, where the nearest neighboring classes of “genetic algorithms” (for Cora) are highlighted for several representative methods, such as in Figs.H.13(g),13(h). It can be seen that the cohort locations and the closeness levels differ substantially amongst them. We investigate the effect of α using COVA-E1 as an example and demonstrate the results in Fig.H.14. We observe similar patterns to Fig.6 generated using Places2

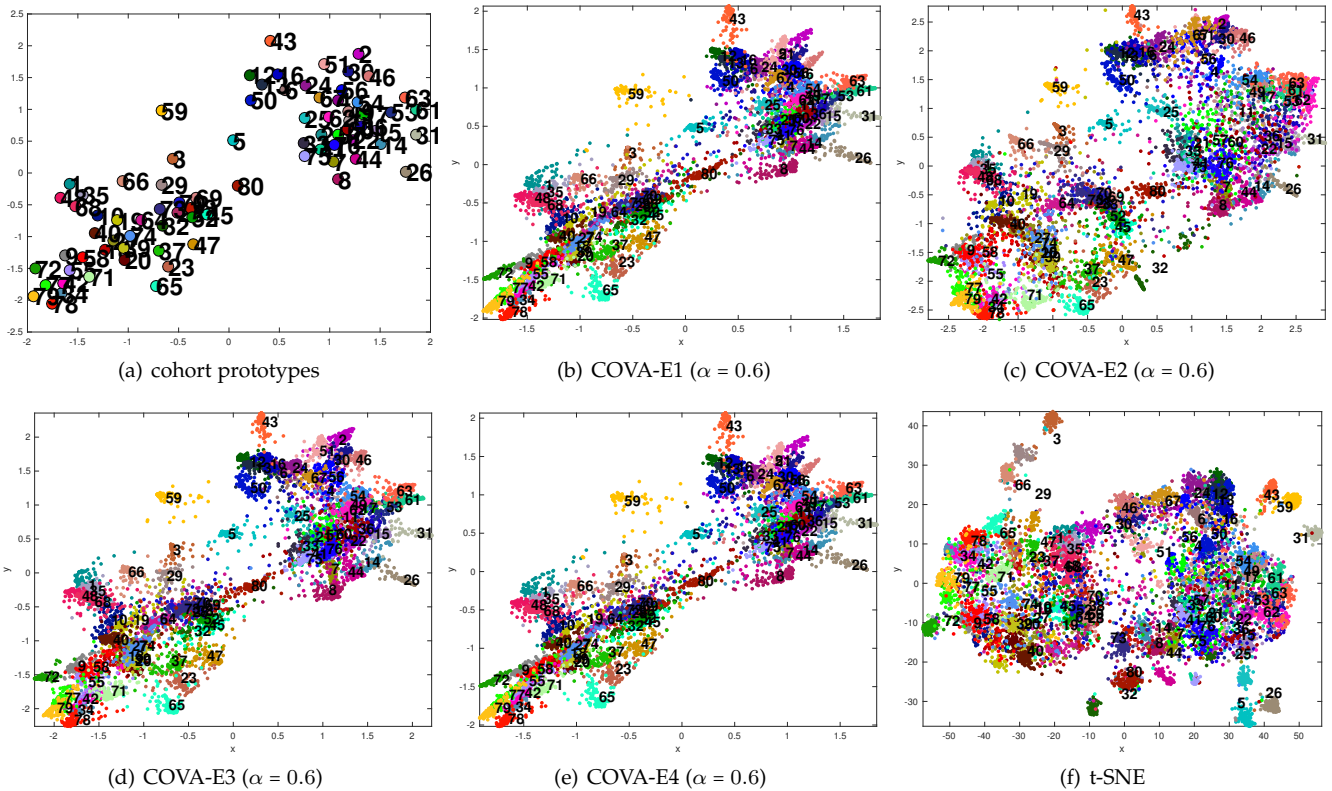


Figure H.8. Visualization of COVA and t-SNE embeddings for 8,000 Places2 images belonging to 80 classes. Different cohorts are numbered and correspond to different shadings.

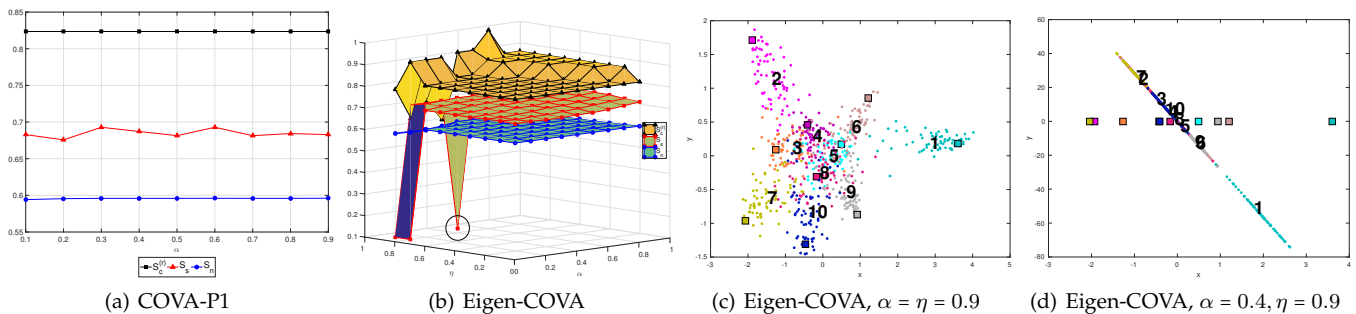


Figure H.9. (a),(b): Illustration of COVA-P1 and eigen-COVA performance change versus varying values of α and η using ten classes of Places2 images. (c),(d): Demonstration of two example outputs of eigen-COVA, where the embedded samples are displayed together with the projected prototypes (indicated by “□”). The example output (presenting a failure case) displayed in (b) is highlighted in (a) by a circle.

images. As α increases from 0.1 to 0.95, S_n decreases from 56.0% to 48.9%, while S_s increases from 81.8% to 99.2%, and S_c from 74.1% to 99.1%, showing denser and more tightly controlled cohorts. The second row of Table H.3 quantitatively compares COVA with existing methods in terms of S_n , S_c and S_s scores, from which similar observations to the first row based on Places2 images can be made.

In addition to visualizing supervised document classes with between-class proximity externally determined based on citation information, we perform another experiment of using COVA to visualize unsupervised publication clusters. This is the same task as that in Section 2 for the three existing methods of t-SNE, SNE and S-t-SNE. Cohort prototypes are generated by t-SNE, taking a between-cluster neighbor adjacency graph as its input. This graph is constructed by first computing the cosine similarity between the cluster centers in the original word space, and then identifying

two effective neighbor clusters for each targeted cluster. Parameter α is set to 0.6 for COVA embeddings and 0.9 for projections, and η set to 0.9 for eigen-COVA. The word vectors \mathbf{X} are used as the input for the embedding methods and the compacted features \mathbf{SYM}^{-1} are used as input for the projection methods. Fig.H.15 illustrates the cluster neighbor preservation performance by COVA-E2. Compared to Table 2 that shows the same cluster neighbor preservation performance for existing methods, a much better match to the original cluster neighbor structure is obtained by COVA in Fig.H.15. Fig.H.16 collects the visualization output produced by different versions of COVA, which all succeed in drawing data cohorts close to their corresponding cohort prototypes, indicated by the matching positions between the learned cohort centers (marked by “○”) and the input cohort prototypes (marked by “□”).

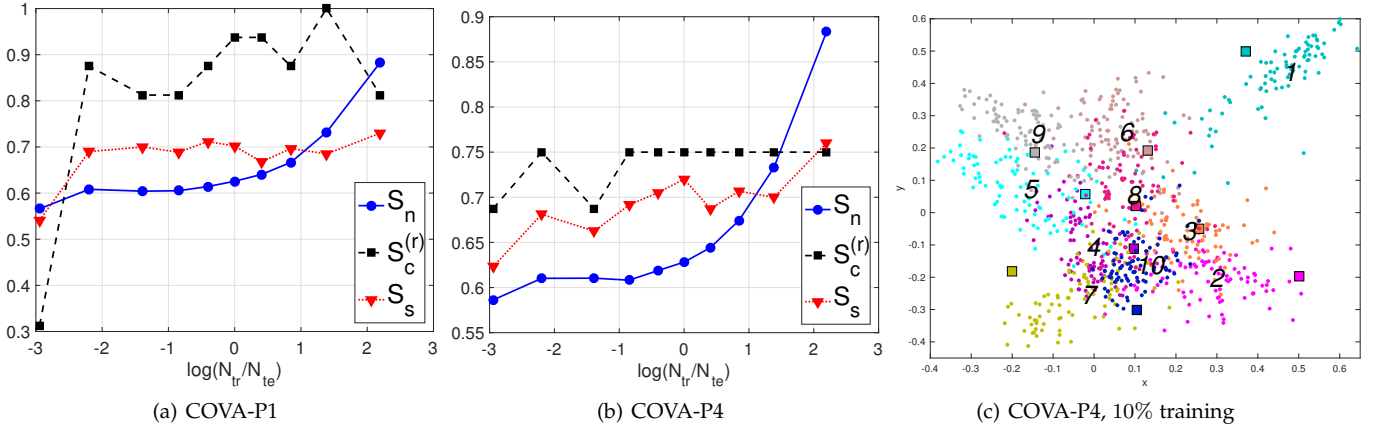


Figure H.10. (a)-(b): Performance change for COVA-P1 and COVA-P4 by varying the amount of training samples using Places2 images, where x-axis represents $\log\left(\frac{N_{tr}}{N_{te}}\right)$ with N_{tr} , N_{te} being the number of training and test samples, and $\frac{N_{tr}}{N_{tr}+N_{te}}$ varying from 5% to 90%. (c) COVA-P4 output where the prototypes are shown as “□”.

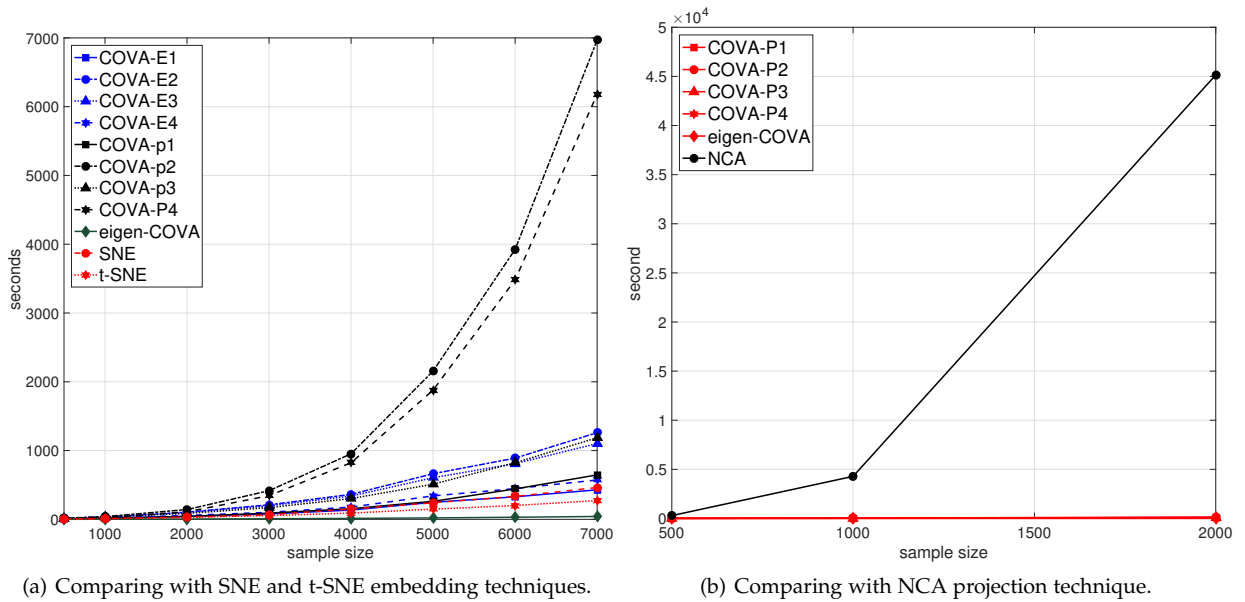


Figure H.11. Computational time comparison between COVA and existing methods.

H.4 CLINICAL TRIALS VISUALIZATION

This document set is retrieved from a clinical trials collection [9], based on nine disease queries of “asthma”, “breast cancer”, “lung cancer”, “prostate cancer”, “leukemia”, “depression”, “schizophrenia”, “cardiovascular” and “HIV”. The top 200 trials, most relevant to each disease are included in the study. Each trial is characterized by the normalized occurrence counts (tf-idf values) of a set of 23,275 words and noun phrases. This results in a $1,800 \times 23,275$ real-valued matrix \tilde{X} , and a $1,800 \times 9$ binary disease cohort indicator matrix Y . The aim for this visualization is to depict the information within \tilde{X} and Y , and highlight the connections between trials and between diseases.

For the clinical trials dataset, to preprocess \tilde{X} and Y , we examine the disease relevance by measuring the word and phrase co-occurrences between the clinical trials. This, for example, can be computed as the cosine similarity between the centroids of two disease cohorts in the word feature

space, forming a 9×9 between-disease similarity matrix with

the ij th element calculated as
$$\frac{\left(\frac{1}{n_i} \sum_{y_p=i} \tilde{x}_p\right)^T \left(\frac{1}{n_j} \sum_{y_q=j} \tilde{x}_q\right)}{\left\| \frac{1}{n_i} \sum_{y_p=i} \tilde{x}_p \right\|_2 \left\| \frac{1}{n_j} \sum_{y_q=j} \tilde{x}_q \right\|_2}.$$

This implies that more relevant diseases share more co-occurred words and phrases in their clinical trials. The corresponding cohort prototypes generated from the similarity matrix are displayed in Fig.17(a). Similar to the Cora setup, the $1,800 \times 9$ compacted feature matrix $X = \tilde{S}Y M^{-1}$ is used as the input to all algorithms, where $\tilde{S} = S \circ N(S, 30)$ and S is computed from \tilde{X} using cosine similarities.

In all experiments, around 15% of the data samples are randomly chosen to be the \tilde{k} sample prototypes to compute the relation features for the projection models. COVA results are demonstrated with the setting of $\alpha=0.9$ and $\eta=0.92$. The cohort membership matrix is computed by Eq.(12), and the local weight matrix W is computed using Euclidean distances with 50 effective neighbors. Different COVA models and existing methods are visually compared in Fig.H.18 and

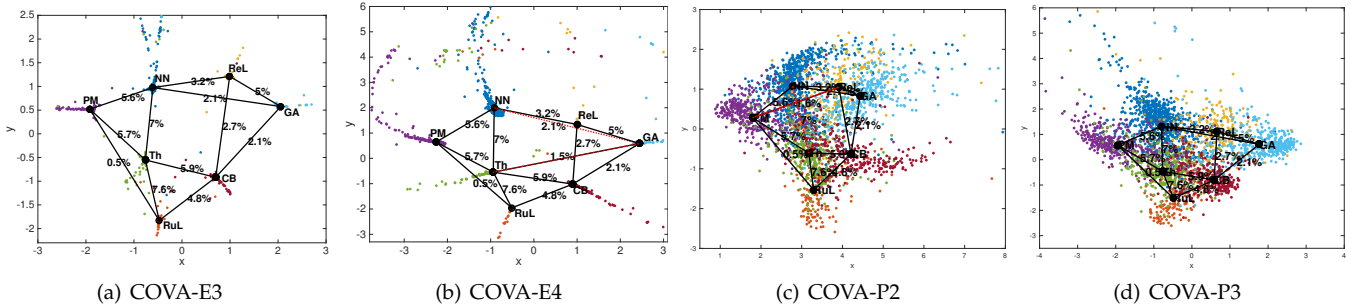


Figure H.12. COVA visualization for the Cora publications by E3, E4, P2 and P3. Cohort locations are controlled by the citation information. Different cohorts correspond to different shadings. Centroids of the neighboring classes are connected with solid lines. Connection mismatches are shown with dotted lines in different shadings.

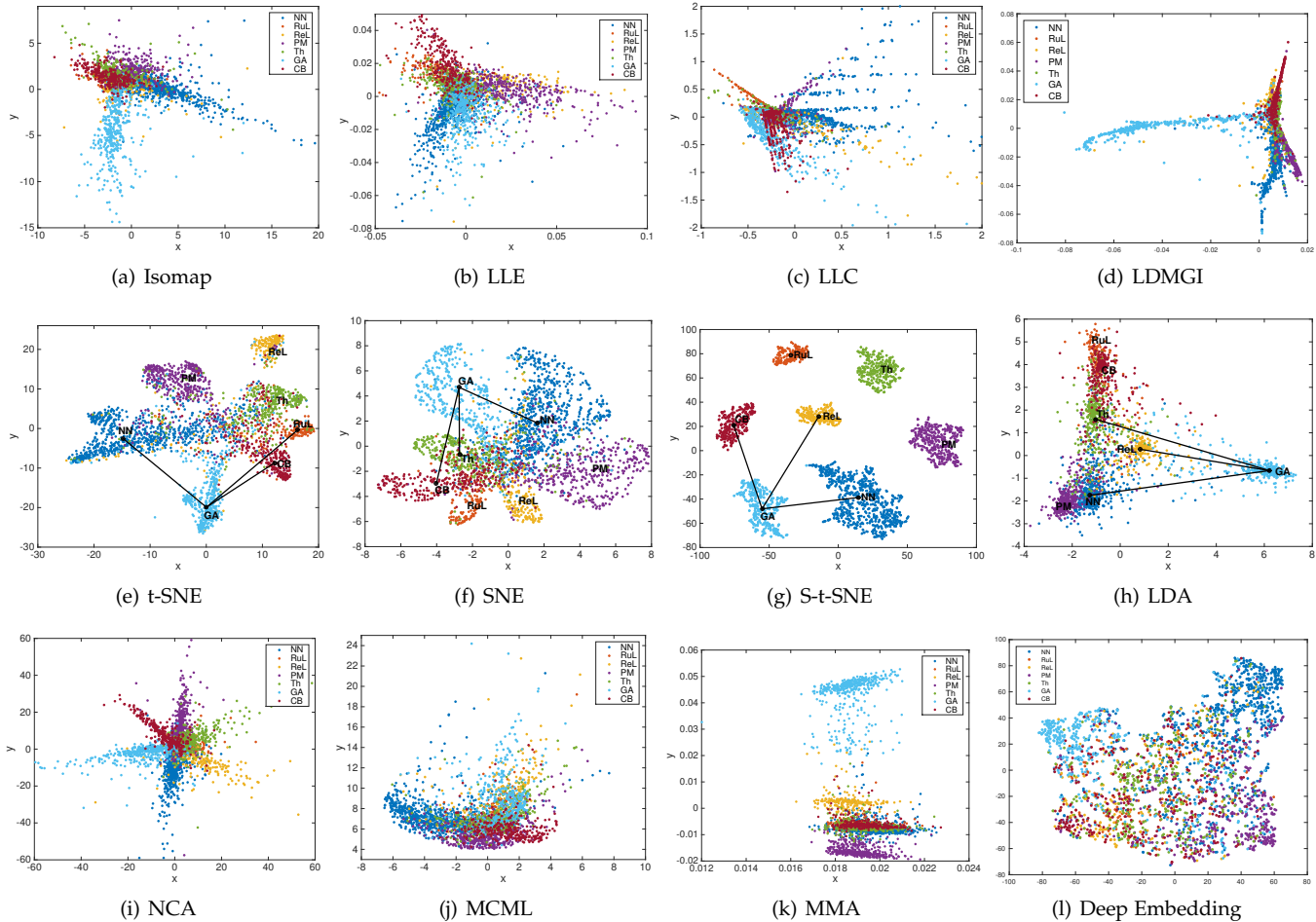


Figure H.13. Visualization output generated by existing algorithms for Cora publications. Different cohorts correspond to different shadings.

Fig.H.7(b). The four neighboring classes of “leukemia” are highlighted by examining the class centroid distances in the visualization output of COVA and some representative existing methods. All COVA models identified the same neighbors (HIV and the three cancer classes for breast, lung and prostate), which match the closeness information indicated by the class prototypes. It can be seen that the cohort locations and the closeness levels differ substantially amongst the existing methods. Additionally, “depression” and “schizophrenia” are two diseases naturally more related to each other than to other ones, and such a relation can be directly inferred from the information content of their clin-

ical trials. However, for some methods, such as t-SNE and SNE, these two classes are mapped quite apart from each other. The third row of Table H.3 quantitatively compares the COVA methods with the existing ones, indicating similar patterns as observed for other datasets (see the quantitative performance analysis in Appendix H.1 for Places2 images).

H.5 FLICKR IMAGE VISUALIZATION

In this experiment, we use the NUS-WIDE collection [10] of 269,648 Flickr images related to 81 concepts and we randomly select 200 images from each of the eight concepts “snow”, “mountain”, “garden”, “flowers”, “tiger”, “bear”,

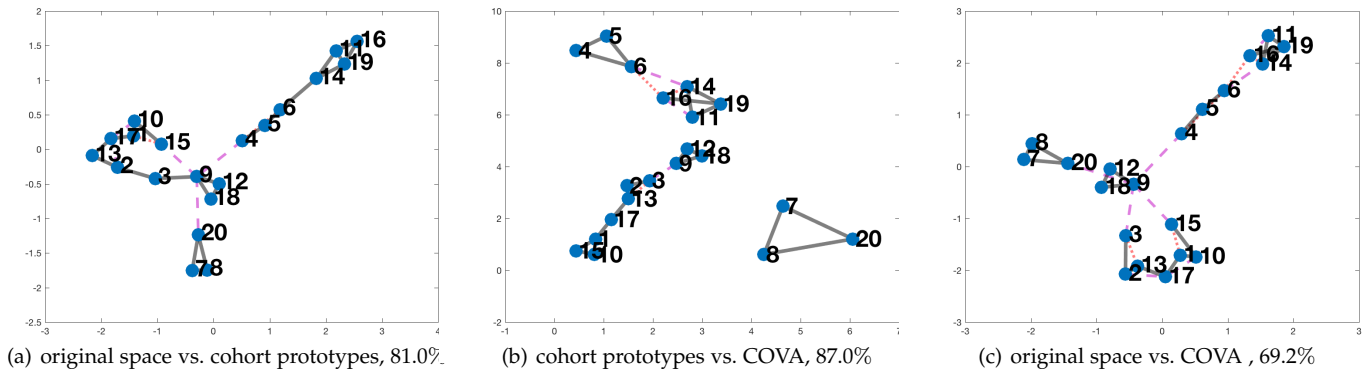


Figure H.15. Illustration of cluster neighbor preservation (two effective neighbors are identified) for COVA-E2 using Cora publications. Given a method pair of “X vs. Y”, edges in solid, dotted and dashed lines indicate true positive, false positive and false negative neighbor links of X compared to Y. Link preservation accuracies are shown in percentages.

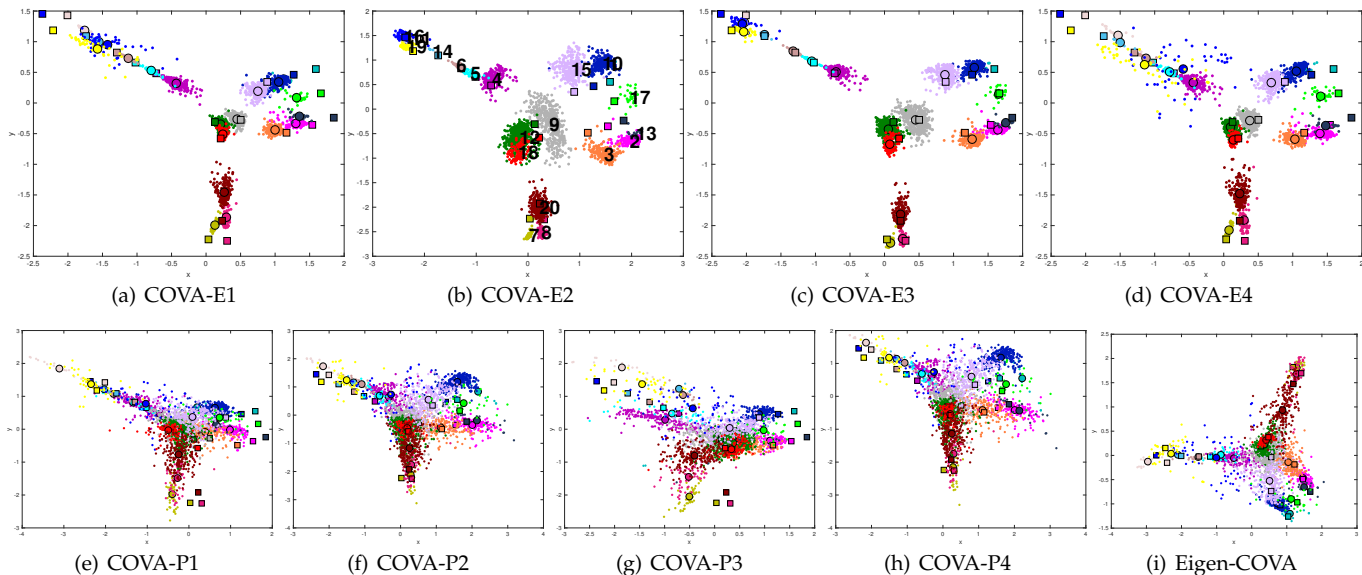


Figure H.16. Visualizing 20 clusters of Cora publications. The used cohort prototypes and learned cluster centers are marked by “□” and “○”, respectively. Cluster IDs are displayed in (b).

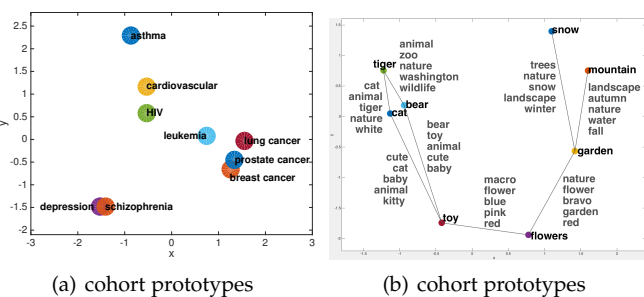


Figure H.17. Illustration of the used prototypes for the clinical trials and Flickr image visualization.

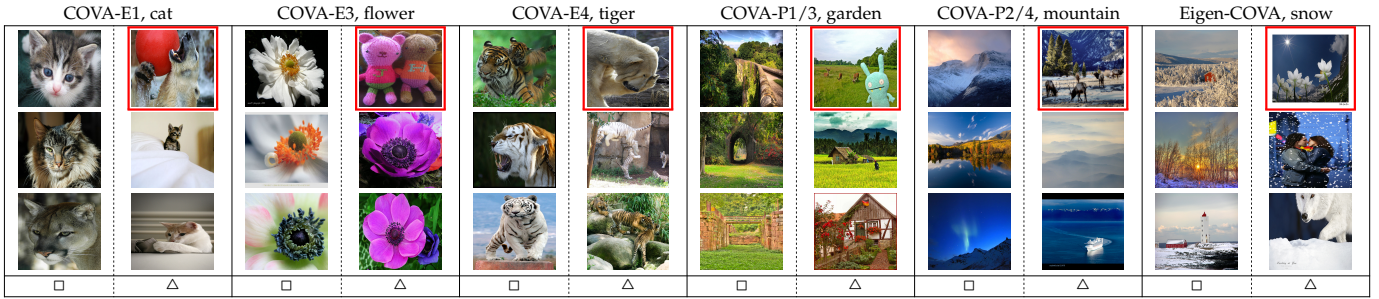
vector represents the distributed semantics of the corresponding word learned from its local context examined over the text collection. Cosine similarities between these vectors are expected to indicate certain semantic relatedness between the corresponding words. We visualize $n=6,500$ (referred to as task 1) and $n=10,000$ (task 2) words assigned to $c=10$ and $c=40$ clusters, respectively. These clusters are obtained by applying k-means clustering over cosine similarities between the word vectors. Given the word vector

matrix \tilde{X} , similar to previous experimental setup, a compacted feature matrix $X = [S \circ N(S, h)] Y M^{-1}$ (where $h=50$ is used for the large word set and $h=30$ for the other) is used as the input of the visualization model, where S is computed from word vectors stored in \tilde{X} by cosine similarity.

We adopt two ways of computing proximity between clusters. For task 1, it is based on the word content of a selected collection of PubMed documents. Letting the ij th element of the matrix Γ denote the presence of the i th word in a j th document, a similarity matrix S_d between words can be derived by computing cosine similarities over the rows of Γ , examining the number of common documents containing each word pair. After applying a 30-nearest neighbor search, the closeness between the word clusters can be computed from $S_d \circ N(S_d, 30)$ using Eq.(6). The generated cohort prototypes are displayed in Fig.H.20(a). For task 2, the proximity between clusters is based on \tilde{X} without utilizing the extra information Γ . It computes the cosine similarity between the cluster centers in the word space. The generated prototypes are displayed in Fig.H.21.

To visualize the 6,500 words grouped into 10 clusters, we run COVA with $\alpha=0.6$ for the embedding models, and

Table H.5
Illustration of sets of neighboring images identified by different COVA models. Enemy images are highlighted in red boxes.



$\alpha=0.95$ for the projection models, and with $\eta=0.3$ for eigen-COVA. The cohort membership matrix is computed by Eq.(12), and the local weight matrix \mathbf{W} is computed using Euclidean distances with 30 effective neighbors. COVA is visually compared with two representative algorithms in Fig.H.20. One is t-SNE that directly computes the embedding coordinates and the other is deep embedding that computes a feature mapping function using a neural network. The S_n and S_s scores are reported in Fig.H.20, and the cohort arrangements can be visually examined by comparing with Fig.H.20(a). E1, E3 and E4 models with $\alpha=0.6$ offer similar output. We demonstrate COVA-E2 output with $\alpha=0.3$ to show that, by reducing the COVA weight α , the generated output can perform similarly good to t-SNE in terms of local neighbor preservation, meanwhile maintaining a good level of control over cohort positioning. It can be seen from Fig.H.20(l) that the deep embedding model shows separation between cohorts, but it performs poorly at local neighbor preservation. Between the COVA projection models, P1-P4 offer very similar performance, and eigen-COVA is better than P1-P4 at local neighbor preservation, while it is worse at cohort separation. Figs.H.20(f)-20(j) show that eigen-COVA follows the cohort arrangement as guided by the cohort prototypes, but not as well as P1-P4.

To visualize the 10,000 words grouped into 40 clusters, we use COVA-E1,2 ($\alpha=0.3$), COVA-P1 ($\alpha=0.95$) and eigen-COVA ($\alpha,\eta=0.9$) to demonstrate the results, and compare with t-SNE in Fig.H.22. All the COVA models exhibit cohort arrangement matching the prototypes. The cohort arrangement of eigen-COVA is a rotated version of the given prototypes, due to the use of linear projection to create auxiliary representations of the cohort prototypes. COVA-E2 and t-SNE show some similarity in the arrangement of some cohorts (e.g., 10 and 37 are close in both plots), but also significant difference (e.g., the proximities between 26 and 17, and between 25 and 15 are in disagreement).

H.7 COMPARISON OF COHORT PROTOTYPES

In previous experiments, t-SNE is used to generate cohort prototypes that carry the desired structural information of the cohort arrangement. Here, we compare several representative alternative approaches for cohort prototype generation, including the classical manifold embedding method LE, SNE that uses Gaussian distribution to estimate the prototype relevance instead of the Student t-distribution as used by t-SNE, as well as several more recent approaches than t-SNE, such as structure preserving embedding (SPE)

[12], LDMGI, local ordinal embedding (LOE) [13] and kernel manifold alignment⁴ (KMA) [14]. The cohort neighborhood structure approximated by prototypes is compared with the original input structure. The correct and mistaken neighbor links are highlighted in Fig.H.24 for 30 classes of scene images and in Fig.H.23 for 20 clusters of Cora publications. These are compared with the prototypes shown in Fig.3(a) and Fig.H.15(a), which are generated by t-SNE with the same input. It can be seen that for the two examined datasets, SNE, t-SNE, LDMGI and LOE are the top performing approaches for preserving the desired neighbor proximity structure between cohorts, where SNE performs best. In Fig.H.25, we also demonstrate the COVA visualization output obtained with SNE prototypes, using COVA-E1 and COVA-E2 as example models.

H.8 SUMMARY

We firstly comment on the performance of COVA based on all the conducted experiments of Section 4 and Appendix H, and provide some general guidelines to users. Overall, the embedding models possess stronger data shaping power than the projection ones. E1 is the fastest embedding version of COVA and it has an analytic solution. It can also be optimized using gradient descent to avoid computing the inverse of a matrix of sample size when processing a large amount of samples. Due to its distance error based objective function, E1 provides the strongest cohort arrangement control but with reduced local neighbor preservation for individual samples. With the KL divergence based objective function, E2 is usually the best at local neighbor preservation for individual samples, but demonstrates less control at cohort arrangement and it is also the slowest embedding COVA model. The users can choose between E1-E4 based on data shaping and computing time preferences. Between P1-P4, the four versions of these COVA projection models often produce similar results, and P1 is the fastest. Eigen-COVA is an extremely fast projection model (e.g., around a minute in our implementation to process 10,000 samples). Comparing against P1-P4, its performance varies depending on the used data and also which score is deemed more important.

According to Eq.(13), the COVA weight α can be viewed as a trade off switch for E1-E4 and P1-P4 which, when

4. The KMA method is developed for domain adaptation to combine information from multiple information sources. To apply it to the generation of cohort prototypes, we randomly divide the prototypes into different groups to be used as different domains and seek a common space to embed them.

increased from 0 to 1, allows them to shift focus from local samples to global cohorts. For eigen-COVA, according to Eq.(23), the interaction between the two weights α and η dictates the balance between the local and global aspects. Empirical parameter analysis shows that P1-P4 are much less sensitive to α than E1-E4, and the user can just fix it (e.g., to a typical value of 0.9) for P1-P4. Eigen-COVA is less sensitive to α than E1-E4, but more than P1-P4, and it can be more sensitive to η than to α . In general, higher values of α and η would result in stronger cohort arrangement control and reduced local neighbor control.

We determine the COVA weights α, η by observing the S_n, S_c (or $S_c^{(r)}$) and S_s scores. In this way, the process of achieving a desired balance between the three objectives of local neighbor preservation, cohort positioning and separation control becomes a score-based selection procedure for the weights. In the following, we show that this is equivalent to searching amongst the Pareto optimal solutions for a multi-objective optimization setup based on a preference function.

A multi-objective optimization problem can be defined as $\min_{x \in X} (O_1(x), O_2(x), \dots, O_{n_o}(x))$, where X denotes the feasible set of the problem and n_o the number of objectives. Typically, there does not exist a feasible solution that minimizes all the objectives simultaneously. Instead, Pareto optimal solutions are examined which are those that cannot be improved in any of the objectives without degrading at least one of the remaining ones. Specifically, a feasible point $x^* \in X$ is said to be Pareto optimal for a multi-objective problem if there is no other point $x \in X$ such that $O_i(x) < O_i(x^*) \forall i \in \{1, \dots, n_o\}$. For simplicity, the main COVA model in Eq.(13) is defined by scalarizing the objectives as a weighted sum problem, but the model could also be expressed as

$$\min_{\mathbf{Z} \in \mathbb{R}^{l \times k}, \text{rank}(\mathbf{Z})=k} (O^{(C)}, O^{(L)}). \quad (\text{H.1})$$

Theorem. Given a fixed weight $0 < \alpha < 1$, the optimal solution \mathbf{Z}^* of the COVA embedding model $\min_{\mathbf{Z} \in \mathbb{R}^{l \times k}, \text{rank}(\mathbf{Z})=k} \alpha O^{(C)} + (1 - \alpha)O^{(L)}$ is a Pareto optimal solution of the two-objective optimization in Eq.(H.1).

Proof. Suppose that \mathbf{Z}^* is not a Pareto optimal solution of Eq.(H.1). Consequently, there exists a full-rank embedding matrix $\mathbf{Z} \in \mathbb{R}^{l \times k}$, such that $O^{(C)}(\mathbf{Z}) < O^{(C)}(\mathbf{Z}^*)$ and $O^{(L)}(\mathbf{Z}) < O^{(L)}(\mathbf{Z}^*)$. Given $\alpha > 0, 1 - \alpha > 0$, we have $\alpha O^{(C)}(\mathbf{Z}) + (1 - \alpha)O^{(L)}(\mathbf{Z}) < \alpha O^{(C)}(\mathbf{Z}^*) + (1 - \alpha)O^{(L)}(\mathbf{Z}^*)$, which contradicts the fact that \mathbf{Z}^* is the optimal solution. \square

The above observation can be applied similarly to the COVA projection model $\min_{\mathbf{U} \in \mathbb{R}^{\tilde{k} \times k}, \mathbf{U}^T \mathbf{U} = \mathbf{I}_{\tilde{k} \times \tilde{k}}} \alpha O^{(C)} + (1 - \alpha)O^{(L)}$ expressed as the two-objective problem $\min_{\mathbf{U} \in \mathbb{R}^{\tilde{k} \times k}, \mathbf{U}^T \mathbf{U} = \mathbf{I}_{\tilde{k} \times \tilde{k}}} (O^{(C)}, O^{(L)})$ and the eigen-COVA model $\min_{\mathbf{U} \in \mathbb{R}^{\tilde{k} \times k}, \mathbf{U}^T \mathbf{U} = \mathbf{I}_{\tilde{k} \times \tilde{k}}} \alpha \eta O_{\text{dist}}^{(C)} + (1 - \alpha)O_{\text{dist}}^{(L)} - \alpha(1 - \eta)O_{\text{aux}}$ expressed as the three-objective problem $\min_{\mathbf{U} \in \mathbb{R}^{\tilde{k} \times k}, \mathbf{U}^T \mathbf{U} = \mathbf{I}_{\tilde{k} \times \tilde{k}}} (O_{\text{dist}}^{(C)}, O_{\text{dist}}^{(L)}, -O_{\text{aux}})$.

Based on the above, we can perform grid search to obtain a set of Pareto optimal solutions associated with different values of the weights, and then choose a setting that offers a reasonable balance between the objectives according to

a preference function that incorporates user's requirements [15]. For COVA, a composition of the three scores of S_n, S_c (or $S_c^{(r)}$) and S_s is a natural choice of this preference function. The Pareto optimal solution that possesses the highest preference can be assumed to offer the best balance between the multiple objectives.

As an alternative to the above preference driven grid search, it is also possible to employ an ϵ -constraint optimization, which converts some objectives to bound constraints [16]. Specifically, for the COVA embedding models, the local neighbor preservation objective can be firstly optimized individually as

$$\min_{\mathbf{Z} \in \mathbb{R}^{l \times k}, \text{rank}(\mathbf{Z})=k} O^{(L)}. \quad (\text{H.2})$$

If we denote by $\hat{O}^{(L)}$ the best objective value found in this step, then, the global cohort objective $O^{(C)}$ can be minimized with the $O^{(L)}$ objective moved to the constraint list, as

$$\min_{\mathbf{Z} \in \mathbb{R}^{l \times k}, \text{rank}(\mathbf{Z})=k} O^{(C)}, \text{ s.t. } O^{(L)} \leq (1 + \epsilon)\hat{O}^{(L)}, \quad (\text{H.3})$$

where ϵ is a small positive relaxation factor or simply a worsening percentage on the already known optimum. The benefit of this strategy is that it may be more natural for the user to specify the relaxation ϵ for achieving the desired balance between objectives. Another alternative is to employ the adaptive weighted sum method [17].

REFERENCES

- [1] P. A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton Uni. Press, 2008.
- [2] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2001.
- [3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 22, pp. 888–905, 2000.
- [4] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [5] N. Boumal, B. Mishra, P. A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *J. Mach. Learn. Res.*, vol. 15, pp. 1455–1459, 2014.
- [6] L. J. P. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
- [7] E. Kokopoulou and Y. Saad, "Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2143–2156, 2007.
- [8] C. T. Zahn, "Graph theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. on Computers*, vol. 100, no. 1, pp. 68–86, 1971.
- [9] I. Korkontzelos, T. Mu, and S. Ananiadou, "ASCOT: a text mining-based web-service for efficient search and assisted creation of clinical trials," *BMC Med. Inform. Decis. Mak.*, vol. 12, 2012.
- [10] T. Chua, J. Tang, R. Hong, and H. Li, "Nus-wide: a real-world web image database from national uni. of singapore," in *Proc. of ACM Int'l Conf. on Image and Video Retrieval*, 2009, p. 48.
- [11] S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, and S. Ananiadou, "Distributional semantics resources for biomedical text processing," in *LBM*, 2013.
- [12] B. Shaw and T. Jebara, in *Proc. of 26th Annual Int'l Conf. on Machine Learning, ICML*, 2009, pp. 937–944.
- [13] Y. Terada and U. von Luxburg, "Local ordinal embedding," in *Proc. of 31th Int'l Conf. on Machine Learning, ICML*, 2014, pp. 847–855.
- [14] D. Tuia and G. Camps-Valls, "Kernel manifold alignment for domain adaptation," *PLOS One*, 2016, dOI:10.1371/journal.pone.0148655.

- [15] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010.
- [16] J. Goulermas, P. Liatsis, and T. Fernando, "A constrained nonlinear energy minimization framework for the regularization of the stereo correspondence problem," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 4, pp. 550–565, 2005.
- [17] I. Y. Kim and O. L. de Weck, "Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation," *Structural and Multidisciplinary Optimization*, vol. 31, no. 2, pp. 105–116, 2006.

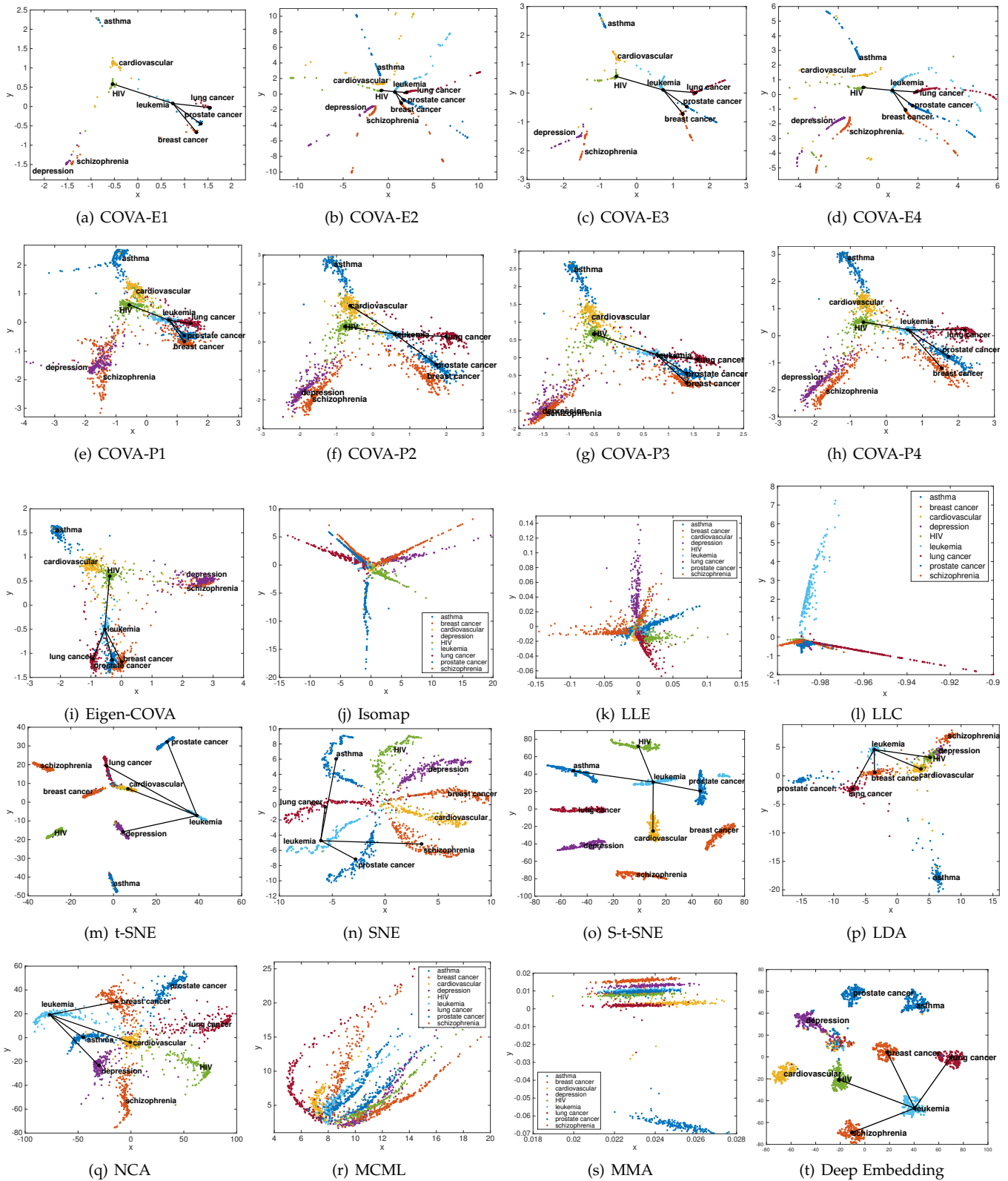


Figure H.18. Visual comparison of different methods using clinical trials. Different cohorts correspond to different shadings.

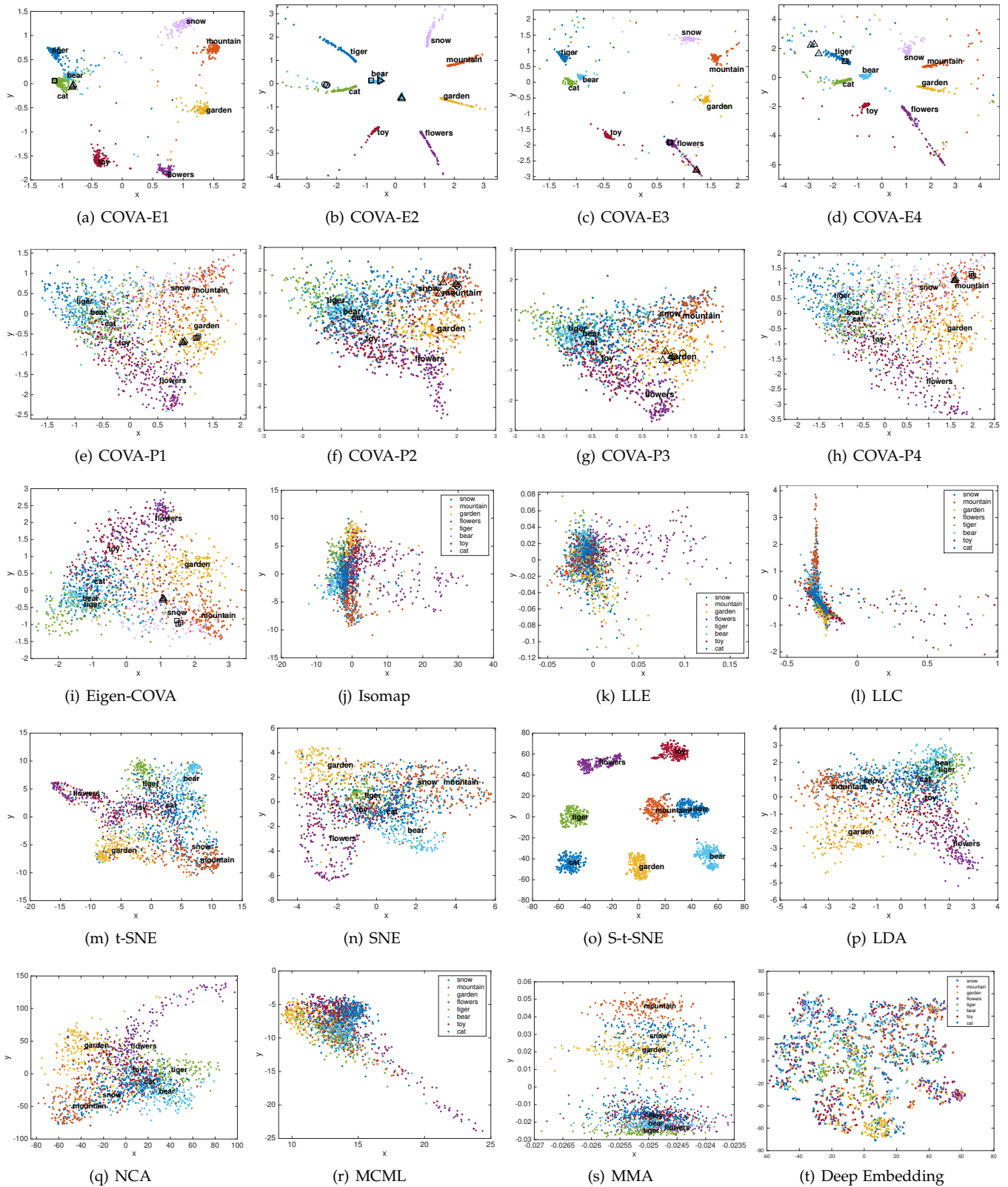


Figure H.19. Visual comparison of different methods using Flickr images. Different cohorts correspond to different shadings.

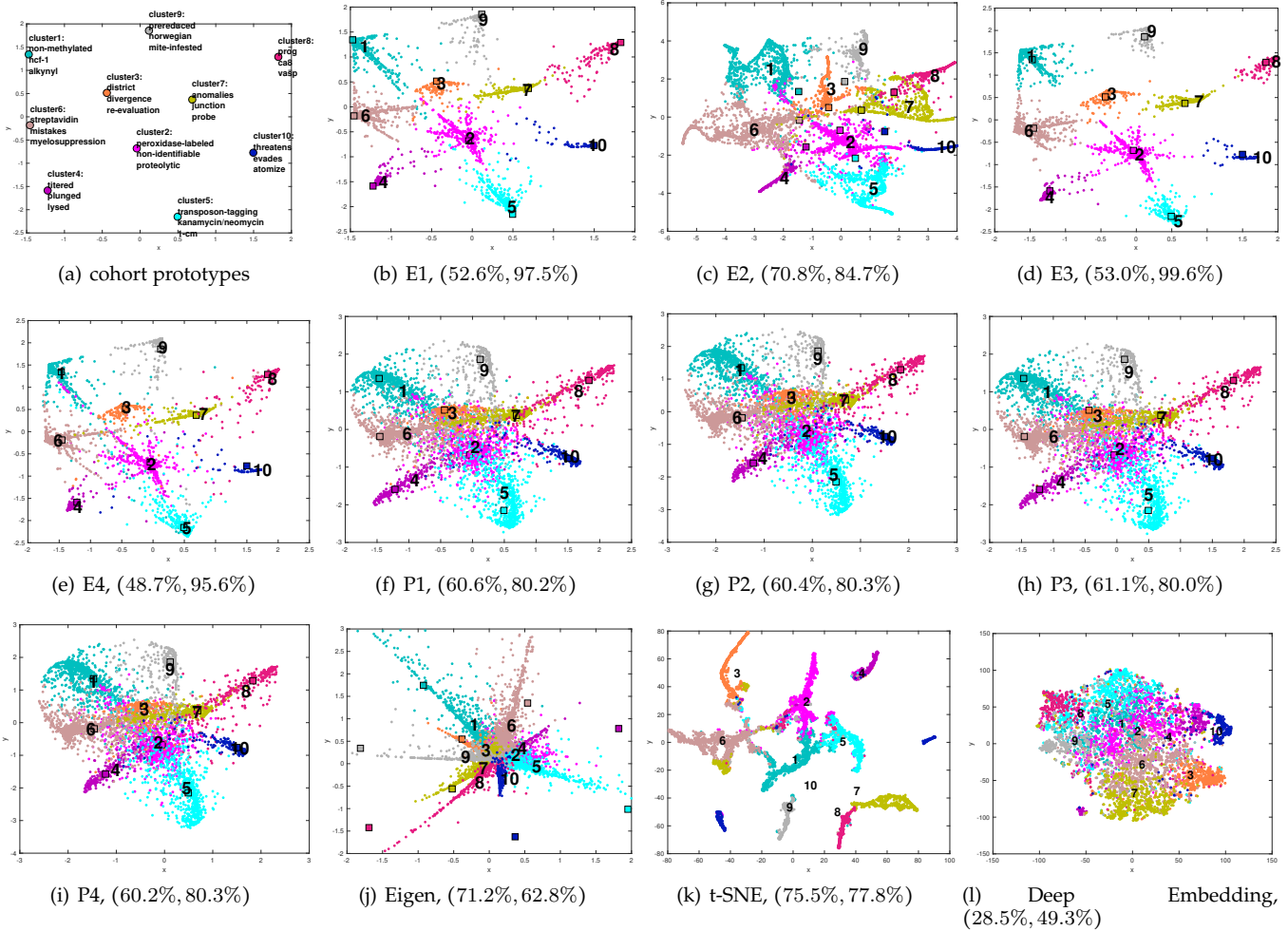


Figure H.20. Visualizing 6,500 word vectors grouped into 10 clusters. Different cohorts correspond to different shadings. The top three words that are closer to the cluster center are listed for each cluster around its prototype in (a). The score pair (S_n, S_s) is shown in percentages.

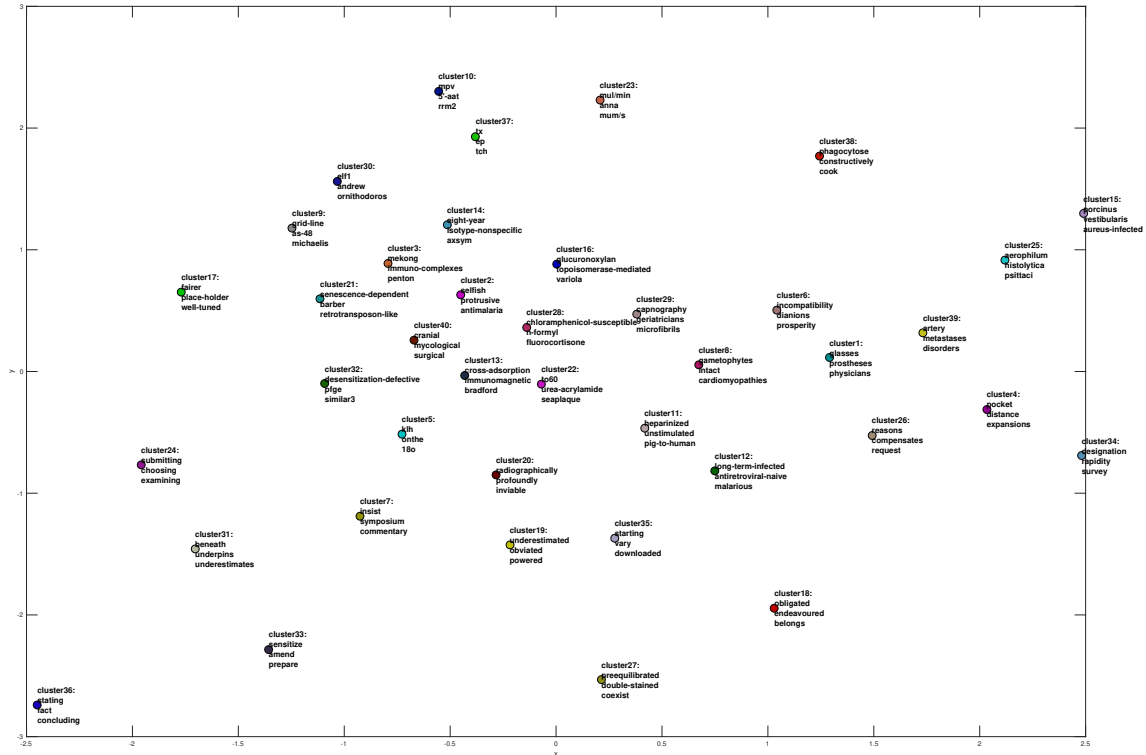


Figure H.21. Illustration of used cohort prototypes. The top three words that are closer to the cluster center are listed for each cluster around its prototype.

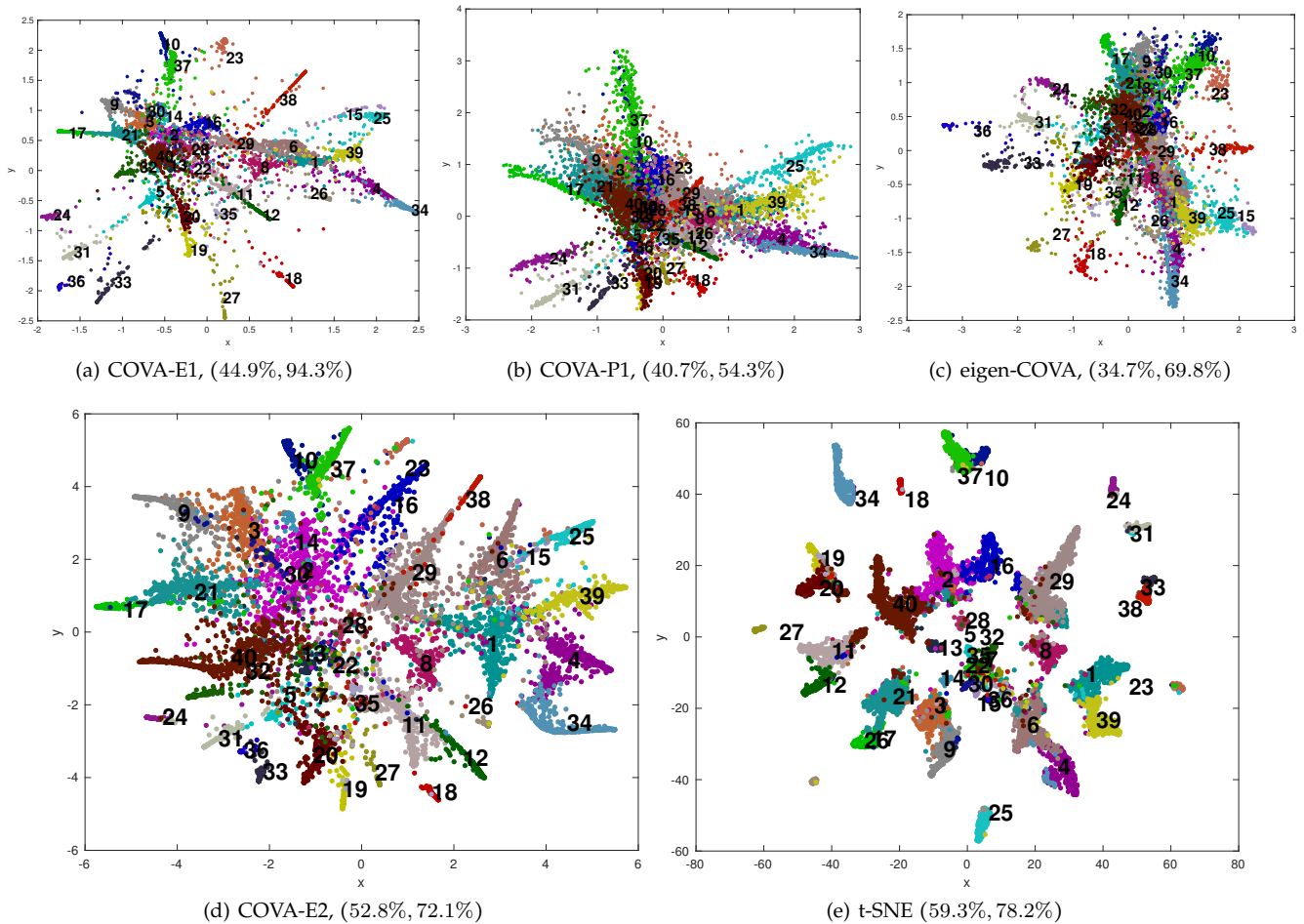


Figure H.22. Visualizing 10,000 word vectors grouped into 40 clusters. Different clusters are numbered and correspond to different shadings. The score pair (S_{n}, S_{s}) is shown in percentages.

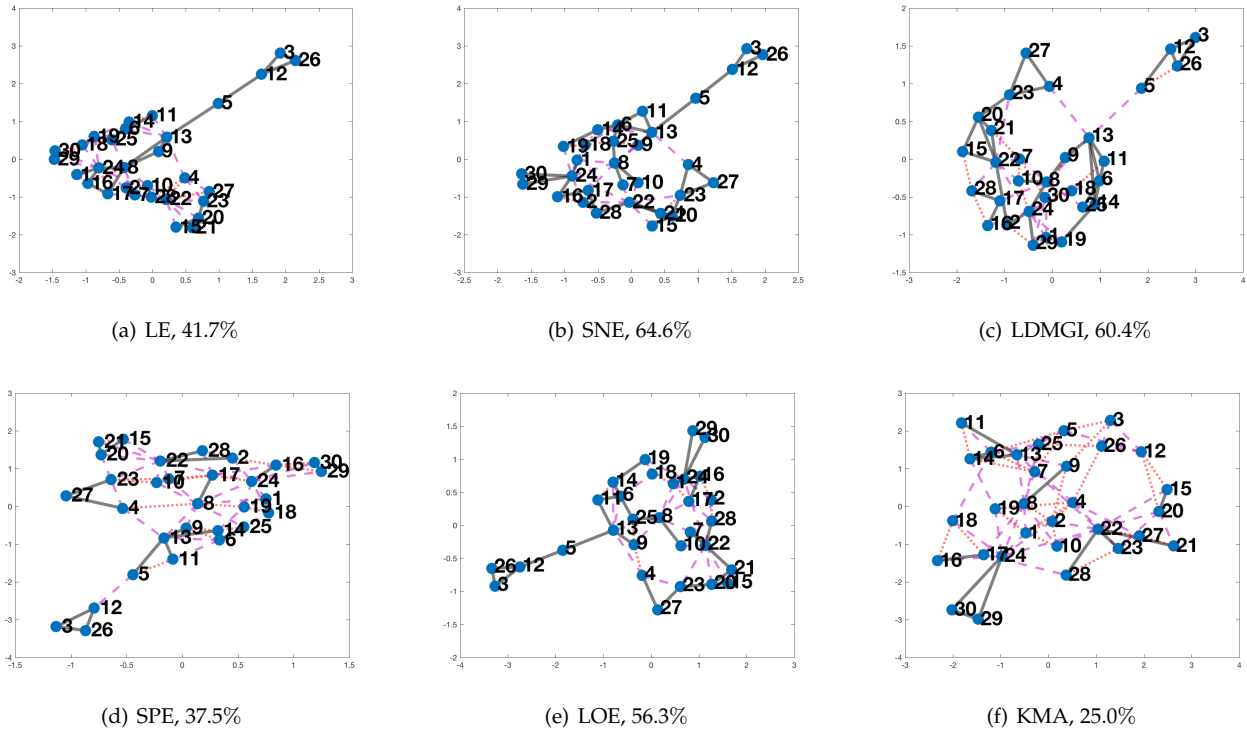


Figure H.23. Comparison of prototype generation approaches in terms of cohort neighbor preservation (two effective neighbors are identified) using 30 classes of scene images. Edges in solid, dotted and dashed lines indicate true positive, false positive and false negative neighbor links approximated by the generated prototypes compared to the desired ones. Link preservation accuracies are shown in percentages.

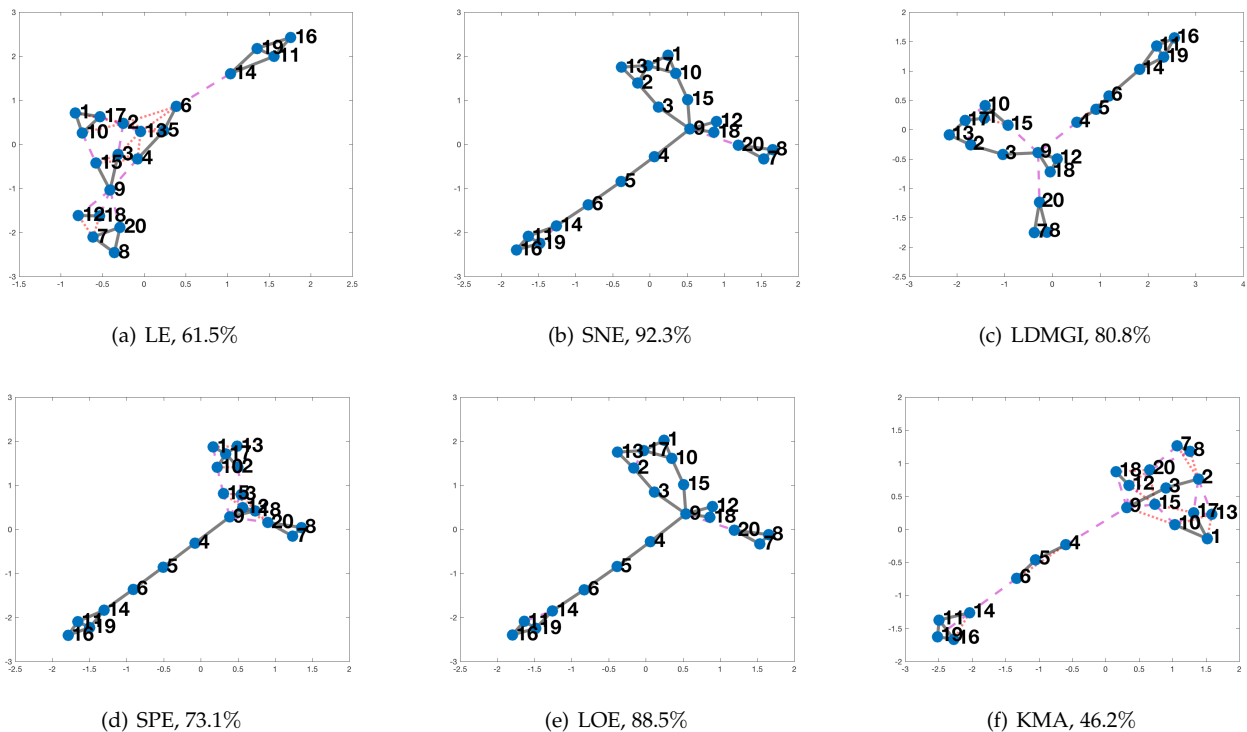


Figure H.24. Comparison of prototype generation approaches in terms of cohort neighbor preservation (two effective neighbors are identified) using 20 clusters of Cora publications. Edges in solid, dotted and dashed lines indicate true positive, false positive and false negative neighbor links approximated by the generated prototypes compared to the desired ones. Link preservation accuracies are shown in percentages.

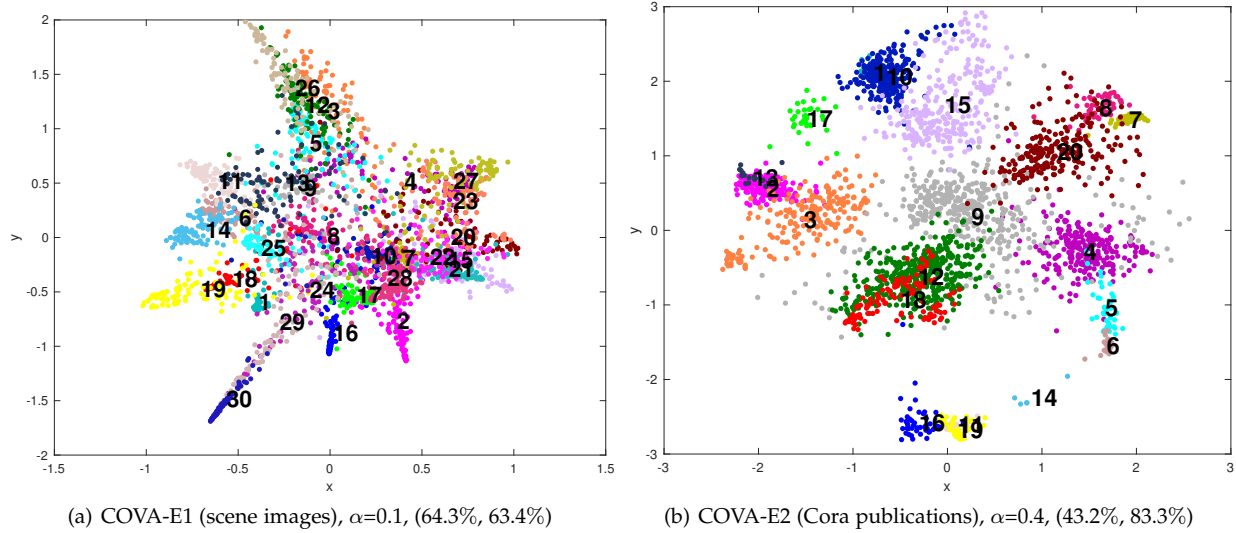


Figure H.25. Examples of COVA embedding output obtained from using cohort prototypes generated by SNE. The score pair of $(S_n, S_c^{(r)})$ is shown in percentages.