The Rashomon Effect for Visualizing High-Dimensional Data

Yiyang Sun[*] Duke University
Haiyang Huang[*+] Duke University
Gaurav Rajesh Parikh[*] Duke University
Cynthia Rudin Duke University

Abstract

Dimension reduction (DR) is inherently non-unique: multiple embeddings can preserve the structure of high-dimensional data equally well while differing in layout or geometry. In this paper, we formally define the Rashomon set for DR—the collection of 'good' embeddings—and show how embracing this multiplicity leads to more powerful and trustworthy representations. Specifically, we pursue three goals. First, we introduce PCA-informed alignment to steer embeddings toward principal components, making axes interpretable without distorting local neighborhoods. Second, we design concept-alignment regularization that aligns an embedding dimension with external knowledge, such as class labels or user-defined concepts. Third, we propose a method to extract common knowledge across the Rashomon set by identifying trustworthy and persistent nearest-neighbor relationships, which we use to construct refined embeddings with improved local structure while preserving global relationships. By moving beyond a single embedding and leveraging the Rashomon set, we provide a flexible framework for building interpretable, robust, and goal-aligned visualizations.

1 INTRODUCTION

Dimension reduction (DR) is the cornerstone of modern data analysis, visualization, and representation learning, enabling researchers to explore and interpret complex high-dimensional datasets in low-dimensional spaces. A wide range of DR techniques have been developed, from linear methods such as PCA (Pearson, 1901) to non-linear techniques such as t-SNE (Van der Maaten and Hinton, 2008), UMAP (Healy and McInnes, 2024; Sainburg et al., 2020), and PaCMAP (Wang et al., 2021; Huang et al., 2024) — that aim to preserve various aspects of the original data structure, such as local and global neighborhoods. A critical limitation of most DR methods is that they return a *single* embedding, often driven by stochastic optimization, heuristic design choices, or algorithmic randomness.

These single output embeddings mask a fundamental fact: there is rarely a unique "correct" embedding of high-dimensional data. Instead, many diverse low-dimensional representations can preserve the data structure nearly equally well. Different runs of the same DR algorithm – with variations in random seed, initialization, bootstrapped samples, or pairwise constraints – can yield embeddings that differ substantially in global layout or local relationships, even when they achieve similar objective scores or visual quality (Kobak and Linderman, 2021; Wang et al., 2021). Such variability introduces epistemic uncertainty that is seldom quantified or used in downstream analysis.

Importantly, this variability also manifests in the *mobility of clusters* within the embedding space. A cluster that appears compact and well-separated in one embedding might shift position, rotate, or stretch in another – while still preserving its internal neighborhood structure. These transformations do not necessarily indicate a failure of the DR method to optimize its objective but rather reflect the geometric flexibility inherent to the mapping. In other words, as long as the cluster itself is maintained, it can translate, rotate, or reorient in the embedding space while still potentially yielding a valid outcome of dimension reduction.

In this work, we *extend the notion of the Rashomon effect from supervised learning (Breiman, 2001) to dimension reduction.* Here, the *Rashomon effect* is the phenomenon that many structurally valid embeddings can be found for the same high-dimensional dataset. Even without labels, most DR methods are guided by well-defined objectives – such as preserving local neigh-

[*]These authors contributed equally to this work
[+]Now at Google.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

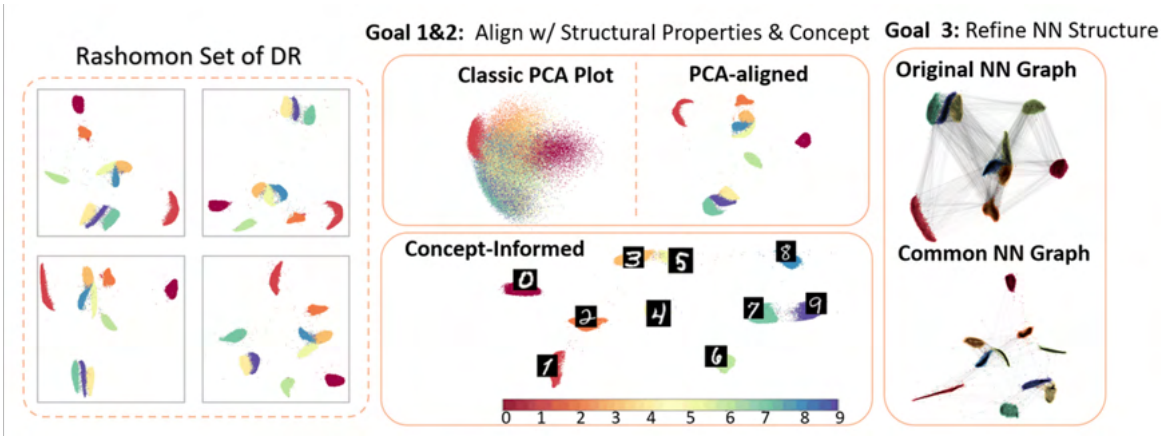

Figure 1: Three goals for generating and exploring the Rashomon set for dimension reduction

borhoods or global geometry – and thus allow us to identify a set of embeddings that perform about equally well. Rather than seeing the resulting variability as a nuisance, we instead treat it as a source of opportunity and insight, as we show in Fig. 1. First, we can steer embeddings toward desirable properties, such as **alignment with principal components** (Goal 1) or **prior knowledge** (Goal 2), to enhance interpretability without compromising structural fidelity. Second, by comparing many good embeddings, we can uncover structural relationships – such as cluster boundaries or neighborhood graphs – that are stable **across alignment goals, random seeds, or bootstrapped samples** and that appear in the results of many DR methods (Goal 3). These recurring patterns represent a form of unsupervised consensus, capturing the underlying structure in the data that is robust to algorithmic or sampling choices.

2 RELATED WORK

Dimension Reduction Early dimension reduction methods focused on preserving global geometric structure. MDS (Torgerson, 1952) preserves pairwise distances, while PCA (Pearson, 1901) maintains variance along principal directions, and NMF (Lee and Seung, 1999) preserves part-based representations. While effective at maintaining the overall data structure, these techniques often fail to capture local neighborhoods and clusters. To address this limitation, *local* DR methods were developed. Examples include Isomap (Tenenbaum et al., 2000), LLE (Roweis and Saul, 2000), Laplacian Eigenmap (Belkin and Niyogi, 2001), and more recent Neighborhood Embedding algorithms such as t-SNE (Van der Maaten and Hinton, 2008), UMAP (Healy and McInnes, 2024), PaCMAP (Wang et al., 2021), LocalMAP (Wang et al., 2024), and many others (Tang et al., 2016; Amid and Warmuth, 2019; Artemenkov and Panov, 2020; Sarfraz et al., 2022; Damrich et al., 2023; Van Assel et al., 2022; Zu and Tao, 2022). To support online and continual learning scenarios, *parametric* DR methods learn a mapping function that approximates the DR objective. Notable examples include Paramt-SNE (van der Maaten, 2009), ParamUMAP (Sainburg et al., 2020), InfoNC-t-SNE (Damrich et al., 2023), ParamRepulsor (Huang et al., 2024) and others (Böhm et al., 2023; Moor et al., 2020; Nazari et al., 2023).

We note that all of these algorithms are designed to produce a **single** embedding that captures the underlying structure of the data. However, in this work, we observe that multiple distinct embeddings can satisfy the same DR objectives, and we show that constructing a diverse set of such valid embeddings can be beneficial for improving the robustness and alignment of the embedding.

Embedding Improvement Numerous studies have sought to improve DR results by modifying algorithmic behavior and examining the influence of various components. Some focus on parameter tuning (Wattenberg et al., 2016; Cao and Wang, 2017; Nguyen and Holmes, 2019; Belkina et al., 2019; Kobak and Linderman, 2021), while others explore the impact of loss functions (Wang et al., 2021; Böhm et al., 2022) and graph reweight/update strategies (Colange et al., 2020; Wang et al., 2021; Dalmia and Sia, 2021; Wang et al., 2024). In contrast, our work introduces a novel perspective: we investigate the *invariant* components of embeddings across multiple runs and analyze which elements contribute to preserving DR performance. This insight enables us to create embeddings that extract shared structural information, leading to more robust and reliable representations.

Consensus Embeddings A consensus embedding combines multiple embeddings of the same dataset into a single, unified embedding that captures shared structural information (Viswanath and Madabhushi, 2012). Our work on Rashomon DR can help facilitate consensus embeddings by finding many valid embeddings to combine, and we study this in Goal 3.

Existing consensus embedding methods, such as Median Consensus Embedding (Tomo and Yoneoka, 2025) and C-LLE (Tiwari et al., 2008), predominantly rely on assumptions of linear relationships between input embeddings. We know these assumptions are too strong, and most of them require storing a pairwise $n \times n$ matrix, which is extremely large and difficult to work with. Other approaches—often applied in image segmentation (Viswanath and Madabhushi, 2012) or network alignment (Li et al., 2022)—typically perform fusion through subspace projections or global alignment, implicitly assuming a shared global structure. These assumptions are incompatible with scenarios where embeddings disagree, e.g., where individual clusters may move independently or where local geometries vary across embeddings—common traits within the Rashomon set. Most critically, prior work in consensus embeddings has largely ignored the *trustworthiness of neighborhood graphs*, assuming they are always trustworthy when we know they are not (Wang et al., 2024). In this work, we aim to extract trustworthy nearest neighbor (NN) relationships from the Rashomon set by identifying the local structure that remains stable across multiple high-quality embeddings. Hence, our method reinterprets consensus not as coordinate fusion but as the extraction of consistent local relationships across embeddings. By doing so, we move toward a structure-aware, graph-level consensus that is better aligned with the goals of both interpretability and robust representation learning.

3 DEFINING THE RASHOMON SET FOR DIMENSION REDUCTION

Most DR algorithms lack robustness, often producing different embeddings for the same dataset in different runs (Wang et al., 2021; Kobak and Linderman, 2021; Healy and McInnes, 2024). This variability poses challenges for interpretation. While prior work has treated such inconsistency as a drawback, we observe that it parallels the concept of the Rashomon set in supervised learning (Breiman, 2001), which captures the existence of multiple models that perform equally well on the same data. Motivated by this connection, we extend the Rashomon set framework to the context of DR. We begin with the conventional definition based on the loss threshold and subsequently adapt it to a DR-specific formulation that considers the preservation of pairwise relationships between data points. In this paper, we primarily use parametric dimensionality reduction (DR) methods as examples since they are easier to implement on large datasets through stochastic gradient descent. However, all of the definitions introduced here are equally applicable to non-parametric DR methods.

Let $\mathcal{X} = x_1, \ldots, x_n \subset \mathbb{R}^p$ be a high-dimensional dataset, and let $\mathcal{F}_\theta : \mathbb{R}^p \to \mathbb{R}^d$ be a parametric dimension reduction method with learnable parameters θ, such that $\mathcal{F}_\theta(\mathcal{X}) = y \in \mathbb{R}^{n \times d}$. The set of all $\mathcal{F}_\theta$ functions is denoted by $\mathcal{F}$. Let $\mathcal{L}_{\text{DR}}(\mathcal{X}, \mathcal{F}_\theta)$ denote the loss function that measures the quality of the embedding, specifically measuring how much information $\mathcal{F}_\theta(\mathcal{X})$ preserves about $\mathcal{X}$. Let Θ be the parameter space for $\mathcal{F}_\theta$, and let θ^* be an optimal or reference solution, i.e.,

$$\theta^* \in \arg\min_{\theta \in \Theta} \mathcal{L}_{\text{DR}}(\mathcal{X}, \mathcal{F}_\theta).$$

Definition 3.1 (Rashomon set of Dimension Reduction from a Loss Perspective)**.** We define the Rashomon set of Dimension Reduction from a Loss Perspective, denoted $\mathcal{R}_{\text{loss}}(\mathcal{X}, \mathcal{F}_\theta, \delta, \mathcal{L}_{\text{DR}})$, as the set of all parameter values θ such that the corresponding embedding $y = \mathcal{F}_\theta(\mathcal{X})$ achieves a loss close to the reference: $\mathcal{R}_{\text{loss}}(\mathcal{X}, \mathcal{F}, \delta, \mathcal{L}_{\text{DR}}) :=$

$$\{\theta \in \Theta | \mathcal{L}_{\text{DR}}(\mathcal{X}, \mathcal{F}_\theta) \leq \mathcal{L}_{\text{DR}}(\mathcal{X}, \mathcal{F}_{\theta^*}) + \delta\}.$$

Embeddings within this set are all produced by models whose loss values are within a tolerance δ of the minimum.

While most widely used DR methods are framed as loss minimization problems (Healy and McInnes, 2024; Van der Maaten and Hinton, 2008; Wang et al., 2021; Artemenkov and Panov, 2020), their training objectives can also be interpreted through the lens of nearest neighbor graph preservation. Specifically, these methods often define their loss functions based on the high-dimensional nearest neighbor graph, aiming to minimize discrepancies between pairwise affinities in the original and embedded spaces. Consequently, the optimization process can be viewed as implicitly selecting a graph structure. In this view, a Rashomon set based on loss, which comprises embeddings with low loss, can be interpreted as preserving approximately the same high-dimensional neighbor graph. Embeddings within the same Rashomon set are likely to induce mostly similar weighted graphs over a fixed edge set. This insight motivates a graph-theoretic reinterpretation of the Rashomon set in the context of DR, introduced below.

Suppose we have a candidate set of embeddings, denoted as $\mathcal{S} = \{y^{(1)}, ..., y^{(|\mathcal{S}|)}\}$, which is of size $|\mathcal{S}|$. $\mathcal{S}$ can consist of embeddings from the Rashomon set from

a loss perspective from Definition 3.1. Define the k-nearest neighbors of a point i in $\mathcal{X}$ by $k\text{NN}(i)$; these are asymmetric, i.e., $i \in k\text{NN}(j)$ does not imply $j \in k\text{NN}(i)$. We note that high-dimensional Euclidean nearest neighbors may not correspond to neighbors along the data manifold. However, a sufficiently large initial k ensures that true manifold neighbors are included within the candidate set, even if some false positives are also present. The reduction of false positives is handled subsequently through our scoring-based selection in Section 5, which identifies the most consistently close pairs across multiple embeddings.

To evaluate how an embedding y reproduces the graph structure, we define $W^y \in \mathbb{R}^{n\times n}$ as a weight matrix that shows the strength between all pairs of points. The construction of W^y is a hybrid of both the high-dimensional and low-dimensional spaces: the adjacency structure (which pairs (i, j) are considered neighbors) is determined by kNN in the high-dimensional space $\mathcal{X}$, while the edge weights are computed from distances in the low-dimensional embedding y. If $i \notin k\text{NN}(j)$, then the entry of W^y_{ij} is 1. Otherwise, entry W^y_{ij} is

$$W^y_{ij} = \frac{(\|y_i - y_j\|_2^2 + \gamma)}{(\|y_i - y_j\|_2^2 + \gamma) + 1},$$

where γ is the scaling constant for numerical stability. Nearer points have lower weight, and farther points have higher weight. This function is designed to saturate as points get farther from each other. Thus, W^y depends on the specific embedding y being evaluated: while the adjacency list is fixed by the high-dimensional kNN graph of $\mathcal{X}$, the weights vary with the candidate low-dimensional embedding.

Definition 3.2 (Rashomon set of Dimension Reduction from Graph Perspective)**.** We define the Rashomon set of Dimension Reduction from a Graph Perspective, denoted $\mathcal{R}_{\text{graph}}(\mathcal{Y}; k\text{NN}; \mathcal{S})$, as the set of embeddings in $\mathcal{Y}$ that induce approximately the same W among embeddings in $\mathcal{S}$, that is:

$$\mathcal{R}_{\text{graph}}(\mathcal{Y}; k\text{NN}; S) := \left\{ y \in \mathcal{Y} \;\middle|\; \frac{1}{|S|}\sum_{i=1}^{|S|} d(W^y, W^{y^{(i)}}) \le \epsilon \right\},$$

where W^y is the induced low-dimensional edge weight matrix constructed using kNN, $W^{y^{(i)}}$ is the edge weight matrix of embedding $y^{(i)} \in \mathcal{S}$ constructed using kNN, and $d(\cdot, \cdot)$ is a distance between two weight matrices defined below.

Definition 3.3 (Soft Jaccard Distance Between Weighted Matrices)**.** Let $W^{y^1}, W^{y^2} \in \mathbb{R}^{n\times n}$ be two weighted matrices corresponding to embeddings $y^1, y^2 \in \mathcal{Y}$. We define the *Soft Jaccard Distance* as:

$$d(W^{y^1}, W^{y^2}) := 1 - \frac{\sum_{i,j} \min\left(W^{y^1}_{ij}, W^{y^2}_{ij}\right)/(W^{y^1}_{ij} + W^{y^2}_{ij})}{\sum_{i,j} \max\left(W^{y^1}_{ij}, W^{y^2}_{ij}\right)/(W^{y^1}_{ij} + W^{y^2}_{ij})}.$$

This value lies within the range of $[0, 1]$, where it would be 0 if the two weight matrices are identical and 1 if they are completely different. To ensure scale invariance and comparability across embeddings, each embedding $y \in \mathcal{Y}$ is first standardized such that $\frac{1}{n}\sum_{i=1}^n y_i = 0$, and $\frac{1}{n}\sum_{i=1}^n \|y_i\|_2^2 = 1$, which ensures that distance magnitudes are comparable across embeddings when computing edge weights.

In Section 6.1 and Appendix H, we show that in practice, the two Rashomon definitions exhibit similar behavior when S consists of low-loss embeddings. It is also important to note that several evaluation metrics (mentioned in Section 6 and Appendix C) remain largely consistent within the Rashomon set. This consistency suggests that our Rashomon sets are stable across both definitions and evaluation criteria.

4 KNOWLEDGE ALIGNMENT FOR DR WITHIN THE RASHOMON SET

To investigate how domain knowledge can guide the selection of embeddings within the Rashomon set, we introduce alignment-based regularization terms that nudge a base DR method toward interpretable structure without significantly altering DR loss ($\mathcal{L}_{\text{DR}}$). Our goal is to identify embeddings that remain within the Rashomon set—i.e., that preserve $\mathcal{L}_{\text{DR}}$—while also aligning with external knowledge or structure. Here we develop two useful kinds of alignment.

4.1 PCA-Informed Alignment

In this setup, we choose the DR embedding to align with PCA embeddings while preserving local structure, which means that the embeddings' axes should be as similar as possible to the first two principal components. To formalize this, we introduce a term that encourages the directions between point pairs in the DR embedding to align with those in the PCA embedding.

Consider a pair of points (i, j) that are not identified as nearest neighbor pairs, $i \notin k\text{NN}(j)$ (i.e., these could be further pairs in the PaCMAP algorithm (Wang et al., 2021), non-nearest neighbor pairs in other dimension reduction methods, or contrastive neighbor methods (Huang et al., 2024; Artemenkov and Panov, 2020)). We exclude nearest-neighbor pairs from this alignment term because $\mathcal{L}_{\text{DR}}$ already governs the local structure through NN relationships. Including NN pairs in the PCA alignment would create a competing objective that could distort local neighborhoods. By restricting alignment to non-NN (more distant) pairs, we steer only the global layout toward the principal components, so that the local structure will not be impacted. This separation of concerns is what allows the embedding to remain within the Rashomon set while gaining interpretable axes. For non-neighbor points i and j, let

y_1, y_2 be their coordinates in the DR embedding y, and let $y_{\text{PCA},1}, y_{\text{PCA},2}$ be their coordinates in the PCA embedding y_{PCA}. The goal of this regularization term is to align the directions of $y_1 - y_2$ and $y_{\text{PCA},1} - y_{\text{PCA},2}$ by minimizing the cosine similarity between those two vectors. Therefore, the total loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{DR}} + \lambda_{\text{PCA}} \cdot \mathbb{E}_{i \notin k\text{NN}(j)} \left[\left(1 - \frac{\langle y_1 - y_2, y_{\text{PCA},1} - y_{\text{PCA},2} \rangle}{\|y_1 - y_2\|_2 \cdot \|y_{\text{PCA},1} - y_{\text{PCA},2}\|_2} \right)^2 \right]$$

where λ_{PCA} controls the strength of the alignment constraint. This is studied in Section 4.1 and illustrated in Figure 1. An example of aligning PaCMAP with PCA is shown in Figure 2 for the USPS dataset (Hull, 1994), which shows that the USPS embedding is flipped and rotated toward the principal components without harming the original local structure.

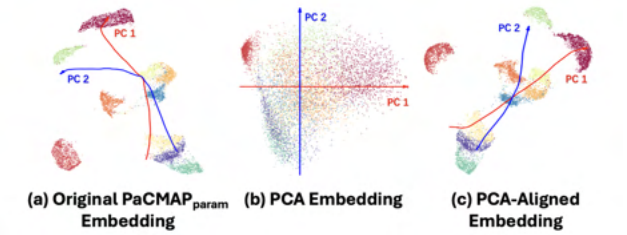

Figure 2: PaCMAP$_{\text{param}}$ embedding with and without PCA-Informed Alignment. The colored curves overlaid on the embeddings are generated by applying the learned parametric DR mapping to points sampled along the first two principal component directions in the original high-dimensional space, thereby visualizing how the DR mapping transforms the PCA axes.

4.2 Concept-Informed Alignment

Here, we encourage the DR embedding to align with a given concept along a designated axis, which is typically the horizontal axis of the embedding space. To do this, we define an axis-informed objective term that encourages this alignment. The concept can arise from any given function, including labels from supervised or semi-supervised problems. Specifically, for each non-nearest-neighbor pair (i, j) where both concept labels l_i, l_j are available, we compute the standardized difference along the horizontal axis of the embedding, denoted by $\tilde{y}_{i,1} - \tilde{y}_{j,1}$, and compare it to the difference between their concept labels, $\tilde{l}_i - \tilde{l}_j$. The axis-informed loss penalizes squared deviations between these two quantities:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{DR}} + \lambda_{\text{Axis}} \mathbb{E}_{i \notin k\text{NN}(j)} \left[\left(\tilde{y}_{i,1} - \tilde{y}_{j,1} - (\tilde{l}_i - \tilde{l}_j) \right)^2 \right]$$

λ_{Axis} controls the strength of the alignment with concept label differences. This formulation encourages the embedding to preserve meaningful variation along an interpretable direction while remaining flexible in other dimensions. This technique is studied in Section 6.1 and illustrated in Figure 1. An example aligning PaCMAP with a label is shown in Figure 4 for the Fashion-MNIST dataset (Xiao et al., 2017), where labels were arranged into a pre-defined order: fashion items are arranged from head to toe.

5 COMMON KNOWLEDGE EXTRACTION FROM THE RASHOMON SET

While each embedding in the Rashomon set satisfies structural constraints based on its loss function, not all local relationships within its embeddings are equally reliable. Some nearest-neighbor (NN) pairs are consistently close across embeddings, while others vary significantly in distance or may not be close in any good embedding. This observation motivates a deeper question: can we extract trustworthy structural information from the Rashomon set by identifying NN relationships that are stable and reproducible? Taking this idea one step further, can we leverage the set of trustworthy pairs to construct an embedding that better captures the essential structure of the data?

Building on the findings of Wang et al. (2024), which identified the presence of false negative NN pairs in embedding neighborhoods that can negatively affect cluster identification, we aim to refine the selection of NN pairs for use in DR methods. Specifically, we leverage embeddings from the Rashomon set to assess the stability of these pairs across reasonable embeddings. We aim to identify pairs that *consistently* appear as neighbors.

Here, we provide theoretical proof to show that stable neighbors are more likely to be trustworthy. Let $\mathcal{D}$ be the distribution of valid embeddings within the Rashomon set. Let $y \sim \mathcal{D}$ be a random embedding drawn from this set, and let $d_{ij}^y = \|y_i - y_j\|_2$ be the pairwise distance in embedding y. We define fixed *population* parameters for each nearest-neighbor pair (i, j), $j \in k\text{NN}(i)$:

$$\mu_{ij}^* = \text{Median}_{y \sim \mathcal{D}}[d_{ij}^y], \tag{1}$$

$$\sigma_{ij}^* = \sqrt{\mathbb{E}_{y \sim \mathcal{D}}[(d_{ij}^y - \mu_{ij}^*)^2]}, \tag{2}$$

$$m_{ij}^* = \sup_{y \in \mathcal{D}} d_{ij}^y, \tag{3}$$

$$M^* = \text{Median}_{y \sim \mathcal{D},\, j \in k\text{NN}(i)}[d_{ij}^y]. \tag{4}$$

Using these fixed population constants, we define the scoring function:

$$\Psi_{ij}(y) = \begin{cases} d_{ij}^y + \lambda \cdot \sigma_{ij}^*, & \text{if } d_{ij}^y \le M^* \\ m_{ij}^* + \lambda \cdot \sigma_{ij}^*, & \text{otherwise} \end{cases} \tag{5}$$

where $\lambda \ge 0$ is a penalty coefficient. This quantity acts as an upper confidence bound for the dis-

tance, penalizing pairs with high variance across embeddings. Since $\Psi_{ij}(y)$ depends on the random sample y through d_{ij}^{y} while using only fixed population constants $(\mu_{ij}^*, \sigma_{ij}^*, m_{ij}^*, M^*)$, it provides i.i.d. scores across independent embeddings. If Ψ is low, the pair is most likely a trustworthy (stable and close) pair within the kNN of i. In practice, we do not have access to the infinite distribution $\mathcal{D}$. Therefore, we use a *calibration-test split*: population parameters $(\mu_{ij}^*, \sigma_{ij}^*, m_{ij}^*, M^*)$ are estimated from a separate calibration set of embeddings, and the scoring is then applied to an independent test set of embeddings. By the law of large numbers, the empirical estimates converge to their population counterparts as the calibration set size increases. To operationalize this approach, we propose two ways of finding the k most trustworthy neighbors for each point i (details are in Appendix B): **Average Rank Approach** and **Mean Score Approach**. The first one ranks the neighbors j by their scores in ascending order and takes the average rank, whereas the second one uses the average raw score directly. In the experiments and theorem, we have used the **Mean Score Approach** as the default, as it directly connects to the theorem and behaves similarly to the rank variant.

We emphasize that the notion of trustworthiness here depends on *both* the high-dimensional space and the embeddings, not on the embeddings alone. The candidate neighbor pairs are defined by the kNN graph in the original high-dimensional space $\mathcal{X}$, and the scoring function Ψ then evaluates whether these high-dimensional neighbors are *consistently* close across multiple low-dimensional embeddings. If an embedding method maps true high-dimensional neighbors to distant locations, those pairs will exhibit high variance and large Ψ scores, and will therefore *not* be deemed trustworthy.

Theorem 5.1. *Let (i, j^*, j') be a triplet of points, and let $\{y^{(1)}, \ldots, y^{(T)}\}$ be T i.i.d. low-dimensional embeddings sampled from $\mathcal{D}$. Define the margin variable for the t-th embedding based on the population scoring function Ψ:*

$$Z_{i,j^*,j'}^{(t)} := \Psi_{ij^*}(y^{(t)}) - \Psi_{ij'}(y^{(t)}).$$

Since Ψ depends on the random sample $y^{(t)}$ through $d_{ij}^{y^{(t)}}$ while using only fixed population constants, and the $y^{(t)}$ are i.i.d., the $Z_{i,j^*,j'}^{(t)}$ are strictly i.i.d. real-valued random variables. Let $\Delta = \mathbb{E}[Z_{i,j^*,j'}^{(t)}]$ be the true expected margin, $V = Var(Z_{i,j^*,j'}^{(t)})$, and B a constant such that $Z_{i,j^*,j'}^{(t)} \leq B$ almost surely. Define the empirical margin:

$$\hat{\Delta}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} Z_{i,j^*,j'}^{(t)}.$$

Then with probability at least $1-\delta$:

$$\Delta \leq \hat{\Delta}^{(T)} + \frac{B\log(1/\delta)}{3T} + \sqrt{\frac{2V\log(1/\delta)}{T}}.$$

We say that j^* is strictly more trustworthy than j' if $\hat{\Delta}^{(T)} + \frac{B\log(1/\delta)}{3T} + \sqrt{\frac{2V\log(1/\delta)}{T}} < 0$, which certifies $\Delta < 0$ with high probability.

Theorem 5.1 shows that as we sample nearest-neighbor candidates from more embeddings, trustworthy pairs with smaller $\Psi_{ij}^{(T)}$ are likely to be sampled. Based on this finding, we propose Algorithm 1 and 2 (see details in App. B) to create a refined kNN graph. In Section 6.2, we will show the refined graph generates better embeddings, accomplishing our Goal 3. It is worth noting that only a small subset of the Rashomon set (as few as five embeddings) is sufficient to achieve a better consensus embedding.

6 EXPERIMENTS

Datasets We conduct experiments across a diverse set of datasets to assess the consistency and reliability of embeddings created from the Rashomon set. For image data, we include MNIST (LeCun et al., 2010), F-MNIST (Xiao et al., 2017) and COIL-20 (Nene et al., 1996). From computational biology, we use several single-cell RNA-sequencing (scRNA-seq) datasets from studies (Kang et al., 2018; Stuart et al., 2019; Zhu et al., 2023; Stoeckius et al., 2017), preprocessed following the protocol in (Townes et al., 2019). Finally, to study embedding stability under controlled structural variations, we include 3D point cloud datasets with known geometry, such as Mammoth (The Smithsonian Institute, 2020) and Airplane (Wu et al., 2015). See Appendix D for more details.

Algorithms We conduct experiments using five widely-used parametric DR methods: ParamUMAP ($\text{UMAP}_{\text{param}}$) (Sainburg et al., 2020), Parametric InfoNC-t-SNE (`InfoNCE`) (Oord et al., 2018), Parametric Neg-t-SNE (`NegtSNE`) (Damrich et al., 2023), Parametric NCVis (`NCVis`) (Damrich et al., 2023; Gutmann and Hyvärinen, 2010) and Parametric PaCMAP ($\text{PaCMAP}_{\text{param}}$) (Huang et al., 2024). Most implementations are from the *contrastive-ne* (Lab, 2022), with modifications to incorporate the additional terms introduced in Sections 4.1 and 4.2. Qualitative results for $\text{PaCMAP}_{\text{param}}$ are presented in the main text, while visualizations for the remaining methods are provided in the appendix. For all parametric methods, we use their default hyperparameters. Since no method-to-method comparison is conducted, keeping the default settings provides a fair basis for evaluation.

Rashomon set construction and evaluation For each DR algorithm, we construct a Rashomon set comprised of 235 embeddings per task. This set is generated by producing 5 embeddings using distinct random seeds across 47 different label weight configurations (λ_{PCA} and λ_{Axis} from Section 4). For concept-informed DR, we used the dataset's class labels as the concept that the embedding aligns along. We then analyze the loss curves corresponding to varying λ values and establish a cutoff threshold that identifies embeddings with comparable performance. Following Breiman (2001), embeddings falling within this threshold are considered part of the Rashomon set and others are excluded. Subsequently, we utilize these selected embeddings to derive a common nearest-neighbor (NN) graph, as detailed in Section 5. This approach identifies structural patterns that are consistently preserved across high-performing embeddings. A detailed process for the experiment is shown in Appendix E. In the experiment setup, we set the missingness of the concept labels (see Appendix D) as 90%, which limits the amount of prior knowledge used. Additional experiments in Appendix H show how the embedding quality is influenced by the concept label weight λ_{Axis} and missingness ratio.

The performance of each embedding within the Rashomon set is evaluated using a suite of supervised and unsupervised metrics that assess both local and global structure preservation that has been used in previous work (Healy and McInnes, 2024; Tang et al., 2016; Huang et al., 2024; Linderman et al., 2019). To further assess the alignment between the DR embeddings and the principal components obtained via Principal Component Analysis (PCA), we introduce the Triplet PCA Score. This metric evaluates the consistency of triplet relationships between the DR embeddings and their corresponding PCA representations, providing insights into the preservation of global structure. A more detailed description of the calculation for each metric is in Appendix C.

6.1 Embedding Alignment Enhances Global Structure without Compromising Local Structure

PCA-informed alignment. Here, we examine how our PCA alignment term can align clusters globally without disrupting local structure, as quantified by the soft Jaccard index (see Definition 3.3). Figure 3 shows that the MNIST embeddings have been effectively aligned to reflect the relative positions observed in the standard PCA embeddings visually and quantitatively. For all the datasets, the experimental results are shown in Table 1.

Concept-aware alignment. Here, we evaluate the concept-aware embedding method of Section 4.2. Qualitatively, on the Fashion MNIST dataset, where the labels correspond to clothing items arranged from head to toe, Figure 4 shows that the embeddings exhibit a clear alignment along the horizontal axis going from feet to head.

Importantly, if we examine the metrics evaluated before and after alignment in Figure 4(b), we observe that for all methods, the metric values remain largely consistent. This indicates that the alignment process does not significantly disrupt the structural properties of the embeddings. For both types of alignment, neither the PaCMAP loss nor the soft Jaccard similarity showed significant degradation. This is consistent with having high-quality embeddings throughout the Rashomon set. More results can be observed in Appendix F and G.

6.2 Trustworthy Common Knowledge Graphs Improve DR Performance

Here we examine how extracted common knowledge can enhance the performance of dimension reduction based on Algorithm in Appendix B. As shown in Figure 5, the embedding constructed using only the trustworthy nearest neighbor pairs (b) results in much clearer separation between different digits in USPS (Hull, 1994), compared to the original embedding (a). Furthermore, the quantitative metrics in (c) show that the combined DR approach (hatched bars) consistently outperforms or matches the original embeddings across multiple algorithms. This demonstrates that leveraging stable relationships across the Rashomon set improves both local detail and global organization, leading to more robust and interpretable representations. More examples are in Appendix I.

6.3 Case Study: Common Knowledge Extraction Filters False Positives

Here we use the MNIST dataset as an example to show how the scoring selection function optimizes the nearest neighbor graph using the Rashomon set, enabling it to identify more trustworthy neighbor pairs. Figure 6 illustrates the difference between the original $\text{PaCMAP}_{\text{param}}$ embedding and the refined embedding after incorporating stable nearest neighbors identified through our Rashomon set-based selection.

Figure 6 shows that refining the nearest neighbor (NN) pairs leads to an embedding that achieves better class separability while still preserving meaningful global relationships among digits. In Figure 6 (a), the left panel shows the original PaCMAP embedding, where some digit clusters are entangled and boundaries between similar classes – such as 3, 5, and 8 – are less clear,

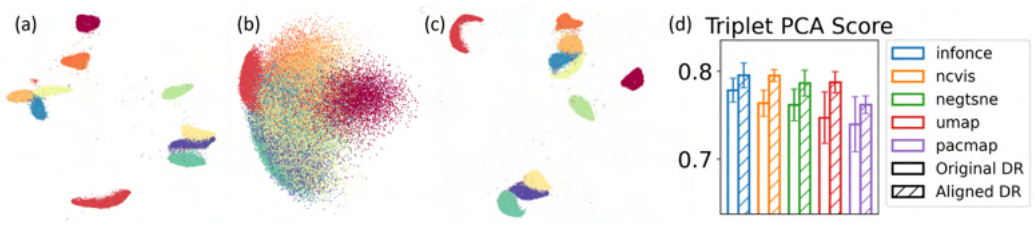

Figure 3: (a) MNIST PaCMAP param embedding, (b) PCA embedding, (c) PCA-informed embedding with λ_{PCA} set to be 0.1. It is nicely aligned with the first two principal components while capturing the detailed cluster structure. (d) Triplet PCA score has improved after PCA-informed alignment.

Table 1: Triplet PCA Score for embeddings before and after PCA-alignment within different datasets, orange are the scores that slightly decrease and the teal are the scores that improve. The almost uniform improvement indicates better alignment with the principal components; the fact that the improvements are small is consistent with preservation of local structure, i.e., staying within the Rashomon set.

	$\text{UMAP}_{\text{param}}$			`NCVis`			`InfoNCE`			`NegtSNE`			$\text{PaCMAP}_{\text{param}}$		
Dataset	Diff	Before	After	Diff	Before	After	Diff	Before	After	Diff	Before	After	Diff	Before	After
AirPlane	0.07	0.67	0.73	0.23	0.71	0.93	0.28	0.65	0.93	0.19	0.73	0.92	0.12	0.77	0.89
COIL20	0.12	0.68	0.8	0.11	0.68	0.79	0.10	0.73	0.83	0.12	0.68	0.8	0.05	0.74	0.79
FMNIST	0.00	0.87	0.87	0.00	0.87	0.88	0.01	0.87	0.88	0.00	0.87	0.87	0.02	0.83	0.85
MNIST	0.04	0.75	0.79	0.03	0.76	0.79	0.02	0.78	0.8	0.02	0.76	0.79	0.02	0.74	0.76
Mammoth	0.02	0.91	0.93	0.02	0.92	0.93	0.01	0.95	0.96	-0.01	0.91	0.9	-0.03	0.92	0.89
USPS	0.03	0.82	0.85	-0.01	0.87	0.86	0.04	0.83	0.87	0.00	0.85	0.85	0.05	0.82	0.87
Cortx	0.01	0.85	0.86	0.02	0.84	0.87	0.06	0.86	0.92	0.01	0.86	0.88	0.02	0.86	0.88
Kang	0.01	0.61	0.62	0.01	0.63	0.64	0.01	0.64	0.65	0.01	0.63	0.64	-0.01	0.66	0.65
CBMC	0.02	0.61	0.64	0.13	0.61	0.74	0.04	0.69	0.73	0.07	0.61	0.69	0.02	0.59	0.61
Stuart	0.02	0.59	0.6	0.04	0.56	0.6	0.10	0.57	0.68	0.02	0.58	0.6	-0.01	0.64	0.63

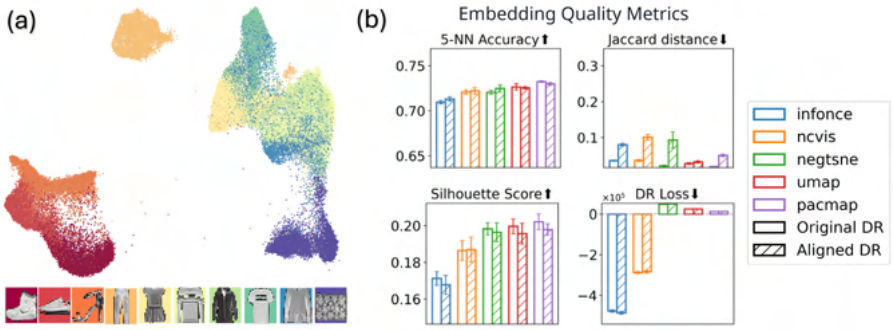

Figure 4: (a) Concept-informed aligned $\text{PaCMAP}_{\text{param}}$ embedding. Alignment is along the horizontal axis from feet (left) to head (right). Footwear is labeled in shades of red to orange, trousers in yellow, dresses in light yellow, pullovers and coats in green, shirts and t-shirts in blue, handbags in purple. (b) Evaluation metrics and losses for FMNIST before and after concept alignment, which remain generally unchanged.

due to false neighbor relationships. After selecting only the consistent NN pairs across the Rashomon set, the right panel reveals a cleaner embedding: digit clusters are more compact and well-separated, yet their relative positions remain consistent with semantic similarities (e.g., curved digits remain near each other). This suggests that the global structure is maintained even as local ambiguities are resolved.

Figure 6(b) provides a closer look at several rejected NN pairs. These pairs often span across visually dissimilar digits, such as connecting a 5 with a 3, or a 9 with a 7. Their rejection suggests that they were unreliable connections in the original embedding; neighbors in pixel space are not equivalent to neighbors along the manifold of data. This visual and empirical evidence demonstrates that pruning away inconsistent neighbor relationships enhances the reliability of both local

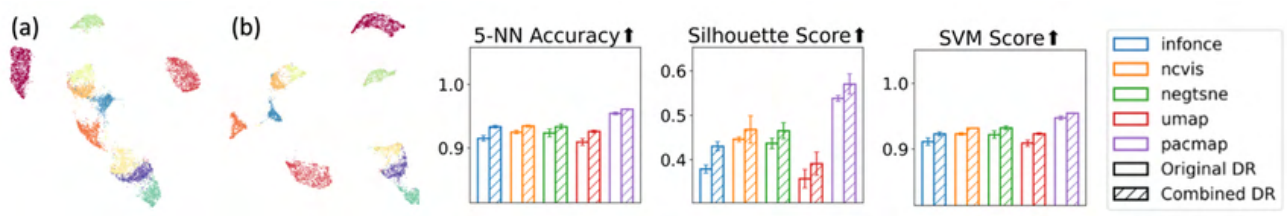

Figure 5: (a) Original $\text{PaCMAP}_{\text{param}}$ embedding of USPS dataset. (b) Common knowledge embedding using only stable neighbor pairs within the Rashomon set. (c) Quantitative comparison of original vs. combined DR embeddings across three evaluation metrics for five methods.

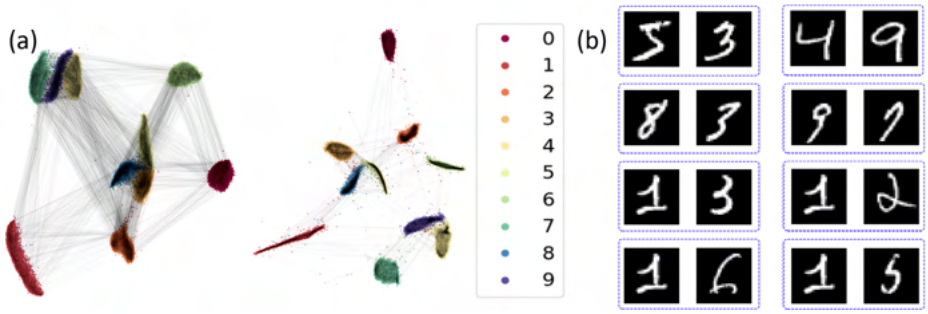

Figure 6: (a) MNIST embedding before (left) and after (right) common NN pairs are selected, (b) Examples of rejected NN pairs when finding common knowledge through the Rashomon set. These pairs are close in pixel space (but not close along the manifold of any digit).

and global structure in the embedding. The result is a more interpretable representation where each digit class forms a tighter, more coherent cluster with fewer spurious links.

7 DISCUSSION AND LIMITATIONS

Our work reframes dimension reduction as a problem with multiple valid solutions, bringing the Rashomon set into the DR paradigm. This perspective enables more interpretable, robust, and customizable embeddings. By aligning embeddings with principal components or user-defined concepts (e.g, developmental trajectories in biology), our approach offers a flexible way to inject interpretable structure into low-dimensional representations. By extracting common information across high-quality embeddings, we enhance the trustworthiness of DR.

One caveat is computation. Similar to random forests or boosting, generating approximations of a Rashomon set requires multiple runs of DR, each with different seeds, sub-samples, or regularization settings, leading to a higher computational cost during training. This overhead is often justified by the robustness and insights gained. Another possible caveat is that the user may try to align the axes with concepts that contradict the natural DR layout. This is not difficult to detect as $\mathcal{L}_{\text{DR}}$ will increase. Formalizing this is a useful direction for future work.

Overall, this work provides a new lens through which to view dimension reduction—one that makes uncertainty visible, interpretable alignment possible, and common structure extractable. As DR continues to play a central role in data analysis pipelines and hypothesis generation for scientific domains, methods that leverage the space of valid embeddings will become increasingly valuable.

References

Amid, E. and Warmuth, M. K. (2019). TriMAP: Large-scale Dimensionality Reduction Using Triplets. *arXiv e-prints*, page arXiv:1910.00204.

Artemenkov, A. and Panov, M. (2020). NCVis: Noise Contrastive Approach for Scalable Visualization. In *Proceedings of The Web Conference*, pages 2941–2947.

Belkin, M. and Niyogi, P. (2001). Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems*, volume 14, pages 585–591. MIT Press.

Belkina, A. C., Ciccolella, C. O., Anno, R., Halpert, R., Spidlen, J., and Snyder-Cappione, J. E. (2019). Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization

and analysis of large datasets. *Nature Communications*, 10(5415).

Bernhardsson, E. (2019). *Annoy: Approximate Nearest Neighbors in C++/Python*. Python package version 1.16.3.

Böhm, J. N., Berens, P., and Kobak, D. (2022). Attraction-Repulsion Spectrum in Neighbor Embeddings. *Journal of Machine Learning Research*, 23(1):4118–4149.

Böhm, J. N., Berens, P., and Kobak, D. (2023). Unsupervised visualization of image datasets using contrastive learning. In *International Conference on Learning Representations*.

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199–231.

Cao, Y. and Wang, L. (2017). Automatic selection of t-SNE perplexity. *arXiv preprint arXiv:1708.03229*.

Coenen, A. and Pearce, A. (2019). Understanding UMAP. `https://pair-code.github.io/understanding-umap/`.

Colange, B., Peltonen, J., Aupetit, M., Dutykh, D., and Lespinats, S. (2020). Steering distortions to preserve classes and neighbors in supervised dimensionality reduction. *Advances in Neural Information Processing Systems*, 33:13214–13225.

Dalmia, A. and Sia, S. (2021). Clustering with UMAP: Why and how connectivity matters. *arXiv preprint arXiv:2108.05525*.

Damrich, S., Böhm, J. N., Hamprecht, F. A., and Kobak, D. (2023). From t-SNE to UMAP with contrastive learning. In *International Conference on Learning Representations*.

FICO (2018). Explainable machine learning challenge. `https://community.fico.com/s/explainable-machine-learning-challenge`.

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Healy, J. and McInnes, L. (2024). Uniform manifold approximation and projection. *Nature Reviews Methods Primers*, 4(1):82.

Huang, H., Wang, Y., and Rudin, C. (2024). Navigating the effect of parametrization for dimensionality reduction. In *Neural Information Processing Systems*.

Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554.

Kang, H. M., Subramaniam, M., Targ, S., Nguyen, M., Maliskova, L., McCarthy, E., Wan, E., Wong, S., Byrnes, L., Lanata, C. M., et al. (2018). Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology*, 36(1):89.

Kobak, D. and Linderman, G. C. (2021). Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature Biotechnology*, 39(2):156–157.

Lab, B. (2022). Contrastive neighbor embeddings. `https://github.com/berenslab/contrastive-ne`. Accessed: 2025-05-04.

LeCun, Y., Cortes, C., and Burges, C. (2010). MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Li, M., Coşkun, M., and Koyutürk, M. (2022). Consensus embedding for multiple networks: Computation and applications. *Network Science*, 10(2):190–206.

Linderman, G. C., Rachh, M., Hoskins, J. G., Steinerberger, S., and Kluger, Y. (2019). Fast interpolation-based t-SNE for improved visualization of single-cell rna-seq data. *Nature Methods*, 16:243–245.

Moor, M., Horn, M., Rieck, B., and Borgwardt, K. (2020). Topological Autoencoders. In *International Conference on Machine Learning*, pages 7045–7054. PMLR.

Nazari, P., Damrich, S., and Hamprecht, F. A. (2023). Geometric autoencoders–what you see is what you decode. In *Proceedings of International Conference on Machine Learning*, pages 25834–25857. PMLR.

Nene, S. A., Nayar, S. K., and Murase, H. (1996). Columbia Object Image Library (coil-20). Technical report, Technical Report CUCS-005-96.

Nguyen, L. H. and Holmes, S. (2019). Ten quick tips for effective dimensionality reduction. *PLoS computational Biology*, 15(6).

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*.

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

Roweis, S. T. and Saul, L. K. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326.

Sainburg, T., McInnes, L., and Gentner, T. Q. (2020). Parametric UMAP: learning embeddings with deep neural networks for representation and semi-supervised learning. *ArXiv e-prints*.

Sarfraz, S., Koulakis, M., Seibold, C., and Stiefelhagen, R. (2022). Hierarchical nearest neighbor graph embedding for efficient dimensionality reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 336–345.

Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P. K., Swerdlow, H., Satija, R., and Smibert, P. (2017). Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods*, 14(9):865–868.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W. M., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.

Tang, J., Liu, J., Zhang, M., and Mei, Q. (2016). Visualizing Large-Scale and High-Dimensional Data. In *Proceedings of the 25th International Conference on the World Wide Web*, pages 287–297.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323.

The Smithsonian Institute (2020). Mammuthus primigenius (blumbach). `https://3d.si.edu/object/3d/mammuthus-primigenius-blumbach:341c96cd-f967-4540-8ed1-d3fc56d31f12`.

Tiwari, P., Rosen, M., and Madabhushi, A. (2008). Consensus-locally linear embedding (c-lle): application to prostate cancer detection on magnetic resonance spectroscopy. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008: 11th International Conference, New York, NY, USA, September 6-10, 2008, Proceedings, Part II 11*, pages 330–338. Springer.

Tomo, Y. and Yoneoka, D. (2025). Median consensus embedding for dimensionality reduction. *arXiv preprint arXiv:2503.08103*.

Torgerson, W. (1952). Multidimensional scaling: I Theory and method. *Psychometrika*, 17(4):401–419.

Townes, F. W., Hicks, S. C., Aryee, M. J., and Irizarry, R. A. (2019). Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model. *Genome Biology*, 20(1):1–16.

Van Assel, H., Espinasse, T., Chiquet, J., and Picard, F. (2022). A probabilistic graph coupling view of dimension reduction. *Advances in Neural Information Processing Systems*, 35:10696–10708.

van der Maaten, L. (2009). Learning a Parametric Embedding by Preserving Local Structure. In *Artificial Intelligence and Statistics*, pages 384–391. PMLR.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11).

Viswanath, S. and Madabhushi, A. (2012). Consensus embedding: theory, algorithms and application to segmentation and classification of biomedical data. *BMC Bioinformatics*, 13:1–20.

Wang, Y., Huang, H., Rudin, C., and Shaposhnik, Y. (2021). Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMAP, and PaCMAP for Data Visualization. *Journal of Machine Learning Research*, 22.

Wang, Y., Sun, Y., Huang, H., and Rudin, C. (2024). Dimension reduction with locally adjusted graphs. In *Association for the Advancement of Artificial Intelligence (AAAI) Annual Conference on Artificial Intelligence*.

Wattenberg, M., Viégas, F., and Johnson, I. (2016). How to use t-SNE effectively. *Distill*, 1(10):e2.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine Learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zhu, K., Bendl, J., Rahman, S., Vicari, J. M., Coleman, C., Clarence, T., Latouche, O., Tsankova, N. M., Li, A., Brennand, K. J., et al. (2023). Multi-omic profiling of the developing human cerebral cortex at the single-cell level. *Science Advances*, 9(41):eadg3754.

Zu, X. and Tao, Q. (2022). SpaceMAP: Visualizing high-dimensional data by space expansion. In *International Conference on Machine Learning*, pages 27707–27723.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials for The Rashomon Effect for Visualizing High-Dimensional Data

A Proof of Theorem

Theorem A.1. *Let (i, j^*, j') be a triplet of points. Let $\mathcal{D}$ be the distribution of valid embeddings within the Rashomon set. Let $\{y^{(1)}, \ldots, y^{(T)}\}$ be T i.i.d. low-dimensional embeddings sampled from $\mathcal{D}$. Define the population scoring function $\Psi_{ij}(y) = d^y_{ij} + \lambda \cdot \sigma^*_{ij}$ if $d^y_{ij} \leq M^*$, and $\Psi_{ij}(y) = m^*_{ij} + \lambda \cdot \sigma^*_{ij}$ otherwise, where $\sigma^*_{ij}, m^*_{ij}, M^*$ are fixed population constants estimated from a separate calibration set. Define the margin variable for the t-th embedding:*

$$Z^{(t)}_{i,j^*,j'} := \Psi_{ij^*}(y^{(t)}) - \Psi_{ij'}(y^{(t)}).$$

Since Ψ depends on the random sample $y^{(t)}$ through $d^{y^{(t)}}_{ij}$ while using only fixed population constants, and the $y^{(t)}$ are i.i.d., the variables $Z^{(t)}_{i,j^*,j'}$ are strictly i.i.d. real-valued random variables. Let $\Delta = \mathbb{E}[Z^{(t)}_{i,j^*,j'}]$ be the true expected margin. Define the empirical margin:

$$\hat{\Delta}^{(T)} = \frac{1}{T} \sum_{t=1}^{T} Z^{(t)}_{i,j^*,j'}.$$

Let $V = \mathrm{Var}(Z^{(t)}_{i,j^*,j'})$ and B a constant such that $Z^{(t)}_{i,j^*,j'} \leq B$ almost surely. Then with probability at least $1 - \delta$:

$$\Delta \leq \hat{\Delta}^{(T)} + \frac{B \log(1/\delta)}{3T} + \sqrt{\frac{2V \log(1/\delta)}{T}}.$$

We say that the neighbor j^* is strictly more trustworthy than j' if the upper bound of this confidence interval is negative:

$$\hat{\Delta}^{(T)} + \frac{B \log(1/\delta)}{3T} + \sqrt{\frac{2V \log(1/\delta)}{T}} < 0.$$

Proof. The sequence $\{Z^{(t)}_{i,j^*,j'}\}_{t=1}^{T}$ consists of i.i.d. real-valued random variables with mean $\Delta = \mathbb{E}[Z^{(t)}_{i,j^*,j'}]$, variance $V = \mathrm{Var}(Z^{(t)}_{i,j^*,j'})$, and $Z^{(t)}_{i,j^*,j'} \leq B$ almost surely. By the one-sided Bernstein inequality for bounded i.i.d. random variables:

$$\mathbb{P}\left(\frac{1}{T}\sum_{t=1}^{T} Z^{(t)} - \Delta \leq -\epsilon\right) \leq \exp\left(-\frac{T\epsilon^2}{2V + \frac{2B\epsilon}{3}}\right).$$

Setting the right-hand side equal to δ and solving the resulting quadratic in ϵ exactly (without the simplification $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$), we obtain:

$$\Delta \leq \hat{\Delta}^{(T)} + \frac{B \log(1/\delta)}{3T} + \sqrt{\frac{2V \log(1/\delta)}{T} + \frac{B^2 \log^2(1/\delta)}{9T^2}}.$$

Since $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b > 0$, this is bounded by:

$$\Delta \leq \hat{\Delta}^{(T)} + \frac{B \log(1/\delta)}{3T} + \sqrt{\frac{2V \log(1/\delta)}{T}},$$

which is the stated bound. Since $\hat{\Delta}^{(T)} \to \Delta$ by the law of large numbers, for sufficiently large T, the confidence interval will certify the sign of Δ. □

B Generating Embedding through Common Knowledge

In this section, we provide detailed algorithmic procedures for extracting a common knowledge graph from a Rashomon set of embeddings and generating a refined embedding. Both algorithms below use the *calibration-test split* described in Section 5: a separate calibration set of embeddings is used to estimate the population statistics $(\mu_{ij}, \sigma_{ij}, m_{ij}, M_{\text{global}})$, and these fixed estimates are then used to score neighbor pairs in an independent test (Rashomon) set of embeddings. The two approaches differ in how the per-embedding scores are aggregated to produce a final neighbor ranking:

Indicator (Average Rank) Approach (Algorithm 1): For each test embedding, neighbors are ranked by their penalized score $\Psi_{ij}^{(t)}$. The final selection uses the average rank $\bar{r}_{ij}$ across all test embeddings. This ordinal aggregation is robust to outlier scores since it discards magnitude information.

Mean Score Approach (Algorithm 2): The penalized scores $\Psi_{ij}^{(t)}$ are averaged directly across test embeddings, and the k neighbors with the lowest mean score $\bar{\Psi}_{ij}$ are selected. This approach retains the full distance-aware information from Ψ and is directly connected to the Bernstein concentration bound in Theorem 5.1, where the empirical margin $\hat{\Delta}^{(T)}$ is the difference of two such averages. In our experiments, we use the Mean Score Approach as the default.

Algorithm 1 Refined Embedding – Indicator (Average Rank) Approach

Require: Rashomon set of embeddings $\{Y^{(1)}, Y^{(2)}, \ldots, Y^{(T)}\}$, Calibration set of embeddings $\{C^{(1)}, C^{(2)}, \ldots, C^{(M)}\}$, number of trustworthy neighbors k, original dataset $\mathcal{X} \in \mathbb{R}^{n \times d}$, penalty weight λ

Output: Refined k-NN graph $k\text{NN}_{\text{refined}}$, consensus embedding $\tilde{Y}$

Construct Candidate Neighbor Set: Construct the m-NN graph ($m > k$) from the original dataset $\mathcal{X}$ to define candidate neighbors $\mathcal{N}_m(i)$ for each point i.
Estimate Calibration Statistics:
for each point i within the dataset $\mathcal{X}$ and each neighbor $j \in \mathcal{N}_m(i)$ **do**
  Collect distances from Calibration set: $d_{ij}^C = \{\|C_i^{(r)} - C_j^{(r)}\|_2\}_{r=1}^M$
  Estimate mean $\mu_{ij} \leftarrow \text{mean}_C(d_{ij}^C)$, $m_{ij} \leftarrow \max_C(d_{ij}^C)$ and standard deviation $\sigma_{ij} \leftarrow \text{std}_C(d_{ij}^C)$
end for
Compute global threshold $M_{\text{global}} \leftarrow \text{mean}_{ij \text{ pairs}}(\mu_{ij})$
Rank Edges in Rashomon set:
for each embedding $t = 1$ to T in Rashomon set Y **do**
  for each point i **do**
    for each neighbor $j \in \mathcal{N}_m(i)$ **do**
      Compute raw distance: $d_{ij}^{(t)} = \|Y_i^{(t)} - Y_j^{(t)}\|_2$
      Compute penalized score:
      if $\mu_{ij} < M_{\text{global}}$ **then**
        $\Psi_{ij}^{(t)} \leftarrow d_{ij}^{(t)} + \lambda \cdot \sigma_{ij}$
      else
        $\Psi_{ij}^{(t)} \leftarrow m_{ij} + \lambda \cdot \sigma_{ij}$
      end if
    end for
    Compute rank $r_{ij}^{(t)}$: Rank of neighbor j among $\mathcal{N}_m(i)$ based on score $\Psi_{ij}^{(t)}$ (ascending)
  end for
end for
Select Trustworthy Neighbors via Average Ranking:
for each point i **do**
  for each neighbor $j \in \mathcal{N}_m(i)$ **do**
    Compute average rank: $\bar{r}_{ij} \leftarrow \frac{1}{T} \sum_{t=1}^T r_{ij}^{(t)}$
  end for
  Select k neighbors with the lowest $\bar{r}_{ij}$: $\mathcal{N}_k(i) \leftarrow \texttt{BottomK}(\mathcal{N}_m(i), \bar{r}_{ij}, k)$
end for
Reconstruction:
Construct refined k-NN graph $k\text{NN}_{\text{refined}}$ using $\mathcal{N}_k(i)$ for all i
Create consensus embedding $\tilde{Y}$ using the DR method with $k\text{NN}_{\text{refined}}$
return $k\text{NN}_{\text{refined}}, \tilde{Y}$

Algorithm 2 Refined Embedding – Mean Score Approach

Require: Rashomon set of embeddings $\{Y^{(1)}, Y^{(2)}, \ldots, Y^{(T)}\}$, Calibration set of embeddings $\{C^{(1)}, C^{(2)}, \ldots, C^{(M)}\}$, number of trustworthy neighbors k, original dataset $\mathcal{X} \in \mathbb{R}^{n \times d}$, penalty weight λ

Output: Refined k-NN graph $k\text{NN}_{\text{refined}}$, consensus embedding $\tilde{Y}$

Construct Candidate Neighbor Set: Construct the m-NN graph ($m > k$) from the original dataset $\mathcal{X}$ to define candidate neighbors $\mathcal{N}_m(i)$ for each point i.
Estimate Calibration Statistics:
for each point i and each neighbor $j \in \mathcal{N}_m(i)$ **do**
  Collect distances from Calibration set: $d_{ij}^C = \{\|C_i^{(r)} - C_j^{(r)}\|_2\}_{r=1}^M$
  Estimate mean $\mu_{ij} \leftarrow \text{mean}_C(d_{ij}^C)$, $m_{ij} \leftarrow \max_C(d_{ij}^C)$ and standard deviation $\sigma_{ij} \leftarrow \text{std}_C(d_{ij}^C)$
end for
Compute global threshold $M_{\text{global}} \leftarrow \text{mean}_{ij \text{ pairs}}(\mu_{ij})$
Compute Penalized Scores in Rashomon set:
for each embedding $t = 1$ to T in Rashomon set Y **do**
  for each point i **do**
    for each neighbor $j \in \mathcal{N}_m(i)$ **do**
      Compute raw distance: $d_{ij}^{(t)} = \|Y_i^{(t)} - Y_j^{(t)}\|_2$
      Compute penalized score:
      if $\mu_{ij} < M_{\text{global}}$ **then**
        $\Psi_{ij}^{(t)} \leftarrow d_{ij}^{(t)} + \lambda \cdot \sigma_{ij}$
      else
        $\Psi_{ij}^{(t)} \leftarrow m_{ij} + \lambda \cdot \sigma_{ij}$
      end if
    end for
  end for
end for
Select Trustworthy Neighbors via Mean Score:
for each point i **do**
  for each neighbor $j \in \mathcal{N}_m(i)$ **do**
    Compute average score: $\bar{\Psi}_{ij} \leftarrow \frac{1}{T} \sum_{t=1}^T \Psi_{ij}^{(t)}$
  end for
  Select k neighbors with the lowest $\bar{\Psi}_{ij}$: $\mathcal{N}_k(i) \leftarrow \texttt{BottomK}(\mathcal{N}_m(i), \bar{\Psi}_{ij}, k)$
end for
Reconstruction:
Construct refined k-NN graph $k\text{NN}_{\text{refined}}$ using $\mathcal{N}_k(i)$ for all i
Reconstruct consensus embedding $\tilde{Y}$ using the DR method with $k\text{NN}_{\text{refined}}$
return $k\text{NN}_{\text{refined}}, \tilde{Y}$

C Evaluation Metrics

To assess the quality of learned embeddings under both supervised and unsupervised settings, we evaluate them using the following metrics.

C.1 Soft Jaccard distance

This metric measures the distance between two weighted matrices originating from two embeddings, particularly in terms of how consistently they preserve pairwise relationships across a shared large nearest neighbor (NN) graph.

1. Let kNN be a fixed large NN graph derived from high-dimensional data using an NN algorithm (e.g., ANNOY(Bernhardsson, 2019)). In this experiment, we have set up a 50-NN graph.
2. For each data point pair (i, j), define a similarity weight $w_{ij}^{y^1}$ in the baseline embedding and $w_{ij}^{y^2}$ in the compared embedding y^1 and y^2, where $W_{ij} = \frac{(\|y_i - y_j\|_2^2 + \delta)}{(\|y_i - y_j\|_2^2 + \delta) + 1}$, and y is the low-dimensional embedding.
3. Compute the soft Jaccard similarity:

$$d(W^{y^1}, W^{y^2}) := 1 - \frac{\sum_{i,j} \frac{\min\left(W_{ij}^{y^1}, W_{ij}^{y^2}\right)}{W_{ij}^{y^1} + W_{ij}^{y^2}}}{\sum_{i,j} \frac{\max\left(W_{ij}^{y^1}, W_{ij}^{y^2}\right)}{W_{ij}^{y^1} + W_{ij}^{y^2}}}$$

4. Lower values indicate better consistency between neighborhood structures of the two embeddings.

C.2 PCA-aligned Triplet Score (`TripletPCA`)

This metric evaluates whether the embedding preserves the global inter-class relationships revealed by a linear projection (PCA). Specifically, we compare the relative distances between class centroids in PCA space versus those in the embedding.

1. Project the original data $\mathcal{X}$ into PCA space to obtain $\mathbf{y}_{\text{PCA}}$.
2. For each class c, compute its centroid in PCA space and in the evaluated embedding y:

$$\mu_c^{\text{PCA}} = \frac{1}{|C_c|} \sum_{i \in C_c} y_{\text{PCA},i}, \quad \mu_c^y = \frac{1}{|C_c|} \sum_{i \in C_c} y_i$$

3. For all unordered pairs of classes (i, j) with $i < j$, compute the Euclidean distance between their centroids in PCA and in the embedding:

$$D_{ij}^{\text{PCA}} = \|\mu_i^{\text{PCA}} - \mu_j^{\text{PCA}}\|, \quad D_{ij}^y = \|\mu_i^y - \mu_j^y\|$$

4. For all unordered centroid triplets (i, j, k) with $i < j < k$, compare the ordering of distances in PCA and in the embedding:

$$\text{A triplet is } \textit{preserved} \text{ if } \operatorname{sign}(D_{ij}^{\text{PCA}} - D_{ik}^{\text{PCA}}) = \operatorname{sign}(D_{ij}^y - D_{ik}^y)$$

5. The final PCA-guided triplet agreement score is the fraction of triplets with consistent ordering:

$$\text{Score} = \frac{\#\text{ of preserved triplets}}{\text{Total number of triplets}}$$

This metric captures whether the embedding respects the global inter-class structure suggested by a linear reference model (PCA), without relying on individual point-level distances.

C.3 Random Triplet PCA Score (`RandomTripletPCA`)

This metric evaluates whether global relationships between randomly sampled class-level centroids are preserved from PCA space to the embedding.

1. Project the original data $\mathcal{X}$ into PCA space to obtain $\mathbf{y}_{\text{PCA}}$.
2. For each class c, compute its centroid in PCA space and in the evaluated embedding y:
$$\mu_c^{\text{PCA}} = \frac{1}{|C_c|} \sum_{i \in C_c} y_{\text{PCA},i}, \quad \mu_c^y = \frac{1}{|C_c|} \sum_{i \in C_c} y_i$$
3. Randomly sample multiple triplets of distinct class indices (i, j, k).
4. For each triplet, compute the Euclidean distances between class centroids in PCA space and in the embedding:
$$D_{ij}^{\text{PCA}} = \|\mu_i^{\text{PCA}} - \mu_j^{\text{PCA}}\|_2, \quad D_{ij}^y = \|\mu_i^y - \mu_j^y\|_2$$
5. For each triplet, determine the relative ordering of distances:
$$\text{Label}(i,j,k) = \mathbb{I}(D_{ij}^{\text{PCA}} < D_{ik}^{\text{PCA}}), \quad \text{Prediction}(i,j,k) = \mathbb{I}(D_{ij}^y < D_{ik}^y)$$
6. Compute the final agreement score as the proportion of triplets where the ordering is preserved:
$$\text{Score} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}\left(\text{Label}(t) = \text{Prediction}(t)\right)$$

A higher score indicates that the embedding preserves the inter-class distance relationships suggested by PCA. Unlike the full triplet PCA score, this metric uses a randomized subset of triplets to provide a scalable, global evaluation. When preserving the local structure, this metric is slightly worse than the triplet PCA score since the higher score would ruin the local structure to be similar to PCA embedding. Therefore, when evaluating the performance of PCA alignment, we consider both.

C.4 Silhouette Score

The silhouette score was originally used for evaluating cluster quality in unsupervised settings. In this work, we apply it using true class labels as proxy clusters to assess class cohesion and separation.

1. For each data point i in the embedding y:
   - Calculate average distance a_i between i and all other data points within the same class C_i,
   $$a_i = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} \|y_i - y_j\|_2$$
   - Calculate the minimum average distance b_i of i to all points in other classes:
   $$b_i = \min_{C_k \neq C_i} \frac{1}{|C_k|} \sum_{j \in C_k} \|y_i - y_j\|_2$$
2. Compute silhouette score for point i:
$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$
3. The overall silhouette score is the average over all N data points:
$$S = \frac{1}{N} \sum_{i=1}^{N} s_i$$
4. The higher the values, the better quality the embedding is.

C.5 SVM Classification Accuracy

This metric evaluates how well the embedding supports non-linear classification by training a Support Vector Machine (SVM) with an RBF kernel and measuring its prediction accuracy. To improve efficiency, we apply a kernel approximation method.

1. Apply the Nyström method, which approximates the kernel matrix by a low rank matrix, using `sklearn.kernel_approximation.Nystroem` to transform the embedding $\mathbf{Y} \in \mathbb{R}^{n \times d}$ into a higher-dimensional feature space $\Phi(\mathbf{Y}) \in \mathbb{R}^{n \times D}$ such that:

$$K_{\text{RBF}}(\mathbf{y}_i, \mathbf{y}_j) \approx \langle \Phi(\mathbf{y}_i), \Phi(\mathbf{y}_j) \rangle$$

2. Train a linear SVM classifier on the transformed features $\Phi(\mathbf{Y})$ using a one-vs-rest strategy for multi-class problems.

3. Compute the classification accuracy over all n data points:

$$\text{Accuracy} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(\hat{y}_i = y_i)$$

   where $\hat{y}_i$ is the predicted label and $\mathbb{I}(\cdot)$ is the indicator function.

4. Higher accuracy indicates that the embedding supports better class separation under non-linear decision boundaries.

This metric is done under a 5-fold setup in the experiments (each time using 4 folds as the training data for the SVM model and using the remaining fold for the evaluation of accuracy), which captures the global separability of classes in the embedding space using a kernelized classifier.

D Data Description

In our experiments, we evaluated a diverse collection of datasets spanning text, image, and biological domains to examine the effects of different alignment regularizations. Table 2 summarizes each dataset along with the number of samples, features, task types, and label descriptions. Datasets such as MNIST (LeCun et al., 2010), FMNIST (Xiao et al., 2017), USPS (Hull, 1994), and COIL20 (Nene et al., 1996) represent image data, while Human Cortex (Zhu et al., 2023), Kang et al. (Kang et al., 2018), CBMC (Stoeckius et al., 2017), and Stuart et al. (Stuart et al., 2019) focus on single-cell transcriptomic measurements. Tabular datasets such as FICO (FICO, 2018) were also included to assess generalizability. The alignment tasks applied to each dataset fall into two categories: PCA alignment, which enforces global structure consistency with principal component axes, and concept alignment, which aligns the embedding with user-defined or interpretable axes. Most datasets were evaluated under both PCA and concept alignment settings, while others like Airplane and Mammoth only used PCA alignment due to their clear spatial or structural geometry. This comprehensive selection allows us to test the robustness and utility of our methods across a variety of domains and structures.

Table 2: Data Descriptions and Task Types each dataset corresponds to

Dataset	**# Samples**	**# features**	**Tasks Types**	**Labels**
Airplane (Wu et al., 2015)	24,141	3	PCA	Airplane Structure
COIL20 (Nene et al., 1996)	1,440	16,384	Concept, PCA	20 Objects
FICO (FICO, 2018)	9,861	23	Concept	External Risk Score
FMNIST (Xiao et al., 2017)	70,000	784	Concept, PCA	10 Clothes Type
Human Cortex (Zhu et al., 2023)	43,349	100	Concept, PCA	9 Cell Types
Kang et al. (Kang et al., 2018)	13,999	100	Concept, PCA	13 Cell Types
Mammoth (Coenen and Pearce, 2019; The Smithsonian Institute, 2020)	10,000	3	PCA	Mammoth Structure
MNIST (LeCun et al., 2010)	70,000	784	Concept, PCA	Digits 0-9
CBMC (Stoeckius et al., 2017)	67686	100	Concept, PCA	9 Cell Types
Stuart et al. (Stuart et al., 2019)	30,672	100	Concept, PCA	27 Cell Types
USPS (Hull, 1994)	9298	256	Concept, PCA	Digits 0-9

E Details of Experimental Design

To construct the Rashomon set for each DR algorithm, we generate a comprehensive pool of 235 embeddings per alignment task. This is achieved by varying the alignment regularization strength—denoted as the label weight parameters λ_{PCA} and λ_{Axis} (defined in Section 4)—across 47 candidate values and sampling five different random seeds for each configuration. The list of λ candidates spans a wide range of values from near-zero to large magnitudes, specifically $\{0.0, 0.001, 0.002, \ldots, 0.009, 0.01, 0.02, \ldots, 0.1, 0.2, \ldots, 1.0, 2.0, \ldots, 100.0\}$. For concept-aware regularizer, we are using 10% of the data are labeled. For the influence of missingness ratio, label weights and their corresponding loss function, we have shown an example of MNIST in Appendix H using $\text{PaCMAP}_{\text{param}}$.

For all DR methods other than PaCMAP param, we apply a scaling factor of 10,000 to the label weight values during implementation, ensuring consistent influence across algorithms with different objective function scales. In concept-informed DR, the dataset's class labels serve as the alignment concept to guide embedding formation.

To determine which embeddings are included in the Rashomon set, we evaluate the $\mathcal{L}_{\text{DR}}$ across all runs and identify a threshold beyond which embeddings show a significant degradation in performance. This threshold is defined as the point where the loss curve demonstrates a statistically significant increase compared to the minimum observed loss—measured by a consistent gap between the stable region and the rising region of the loss curve. Embeddings whose loss values fall within the low-loss plateau are retained as members of the Rashomon set, while those in the degraded region are excluded.

From the selected Rashomon set of embeddings, we first extract a common 50-nearest neighbor (50-NN) graph following the procedure described in Section 5. For each data point, we then compute edge scores Ψ using Equation 5 to evaluate the reliability of its neighbors. Among the 50 neighbors, we retain the number of edges k (the k is defined as the default setting for each of the methods, e.g. for $\text{PaCMAP}_{\text{param}}$, $k = 10$) with the lowest scores—those deemed most trustworthy—and construct a refined k-NN graph from these pairs. We then reconstruct a new embedding using this filtered neighbor graph. All dimensionality reduction (DR) methods are run with their default settings, as our goal is not to compare across DR methods but rather to assess the improvement in embedding quality before and after incorporating common knowledge graph. A detailed version of the algorithm is shown in Algorithm 1 and 2.Although different datasets and methods may yield Rashomon sets of varying sizes, current experiments have shown that as few as five embeddings are sufficient to demonstrate improvements through common knowledge extraction.

All experiments were conducted on a machine with Intel(R) Xeon(R) Gold 5317 CPU @ 3.00GHz CPU and an NVIDIA A5000 GPU with 128GB Memory.

Common Knowledge Embedding Runtime Breakdown The overall runtime of the common knowledge extraction pipeline can be broken down into three main stages: (i) **Nearest Neighbor Graph Construction**: This step is computed once and reused across all runs. (ii) **Embedding Construction**: This step is performed separately for each DR configuration, but it is highly parallelizable. Multiple embeddings can be processed concurrently per GPU, so the runtime does not scale linearly with the number of embeddings. (iii) **Common Knowledge Extraction**: This step aggregates stable neighbors across embeddings and is relatively lightweight compared to the other stages. In practice, constructing a Rashomon set of several embeddings and performing common knowledge extraction introduces only moderate overhead compared to a single DR run, while yielding significant robustness and interpretability benefits.

F Additional PCA-aligned DR results

Here we provide PCA-aligned and original DR embeddings across five methods—InfoNCE, NCVis, Neg-tSNE, $\text{UMAP}_{\text{param}}$, and $\text{PaCMAP}_{\text{param}}$ —on a set of datasets discussed in Appendix D. Each subplot shows a 2D embedding colored by class labels, with the left column representing original embeddings and the middle column showing the PCA embedding and the right column showing aligned counterparts. The alignment enforces PCA-aware positioning while preserving local structure. The bar plots (bottom) quantify performance using multiple metrics: soft Jaccard distance, triplet satisfaction, random triplet discrimination, and $\mathcal{L}_{\text{DR}}$. Aligned DR embeddings consistently maintain structure quality across all metrics, indicating enhanced trustworthiness and interpretability.

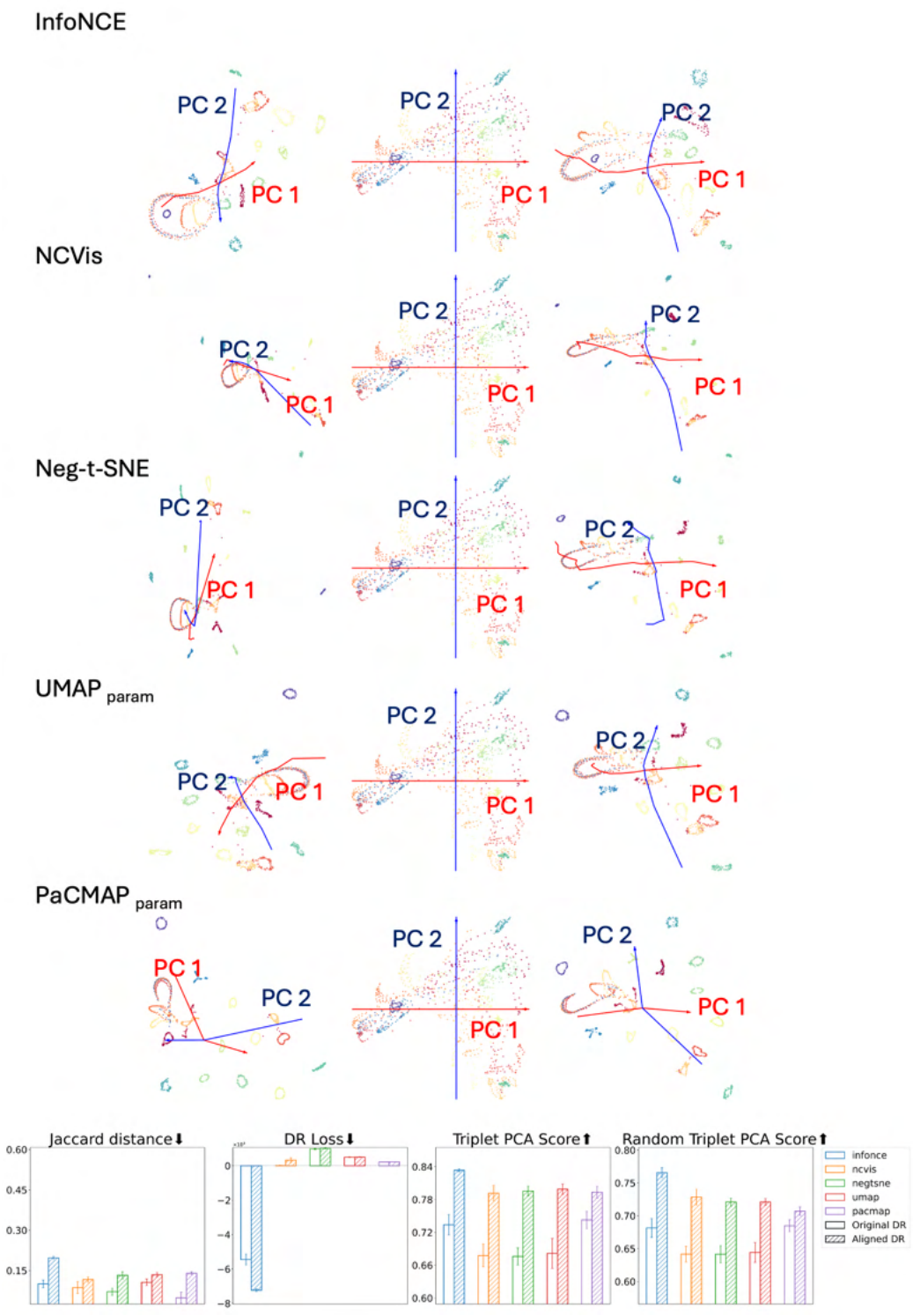

Figure 7: Comparison of original COIL20 embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure. We show that soft Jaccard distance, triplet PCA score, random triplet PCA score, and $\mathcal{L}_{\text{DR}}$ (bottom) remain mostly unchanged and that aligned embeddings consistently maintain structure.

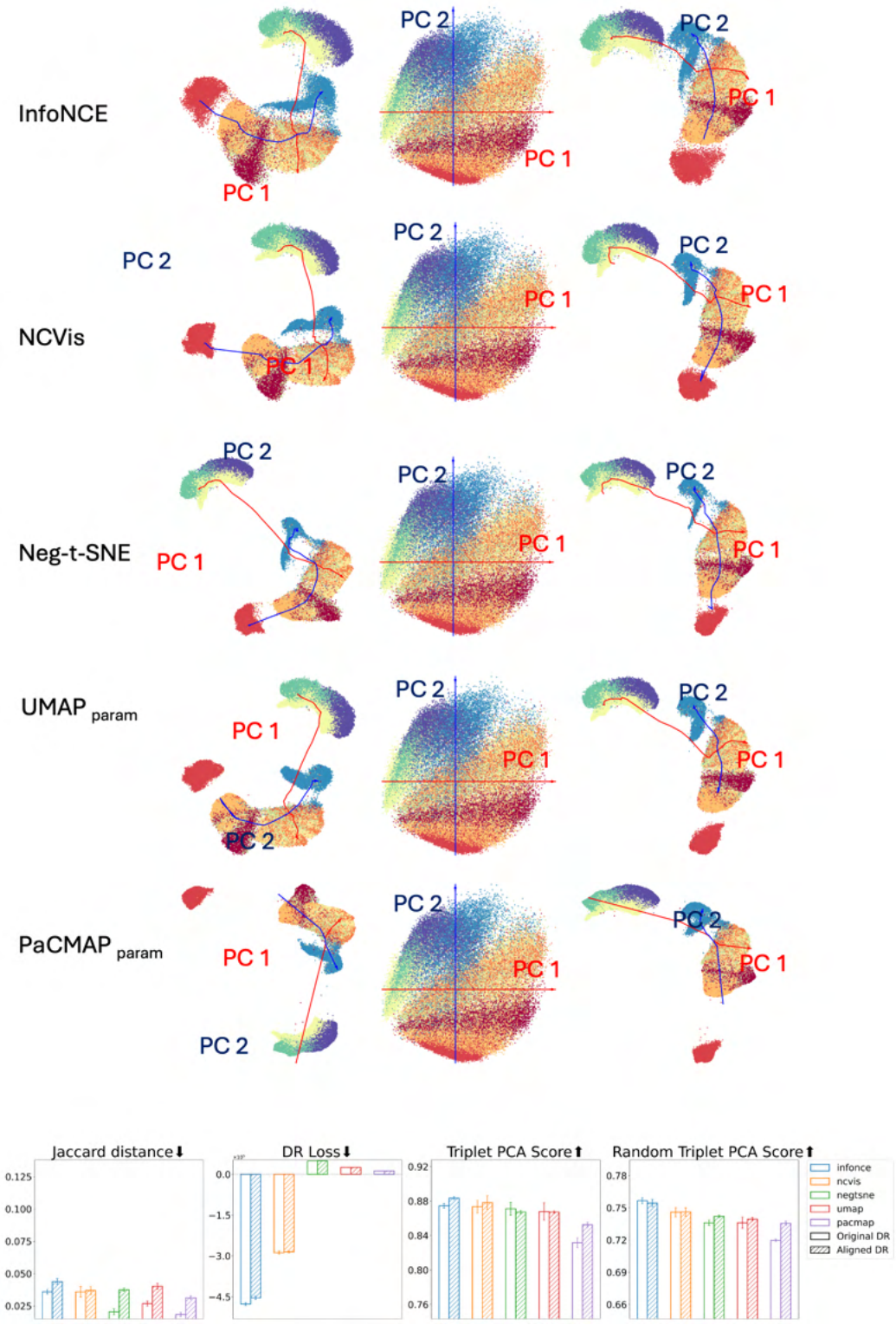

Figure 8: Comparison of original FMNIST embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure. We show that soft Jaccard distance and $\mathcal{L}_{\text{DR}}$ (bottom) remain mostly unchanged and that aligned embeddings consistently maintain structure. Random Triplet PCA score and Triplet PCA score have shown an improvement after the alignment.

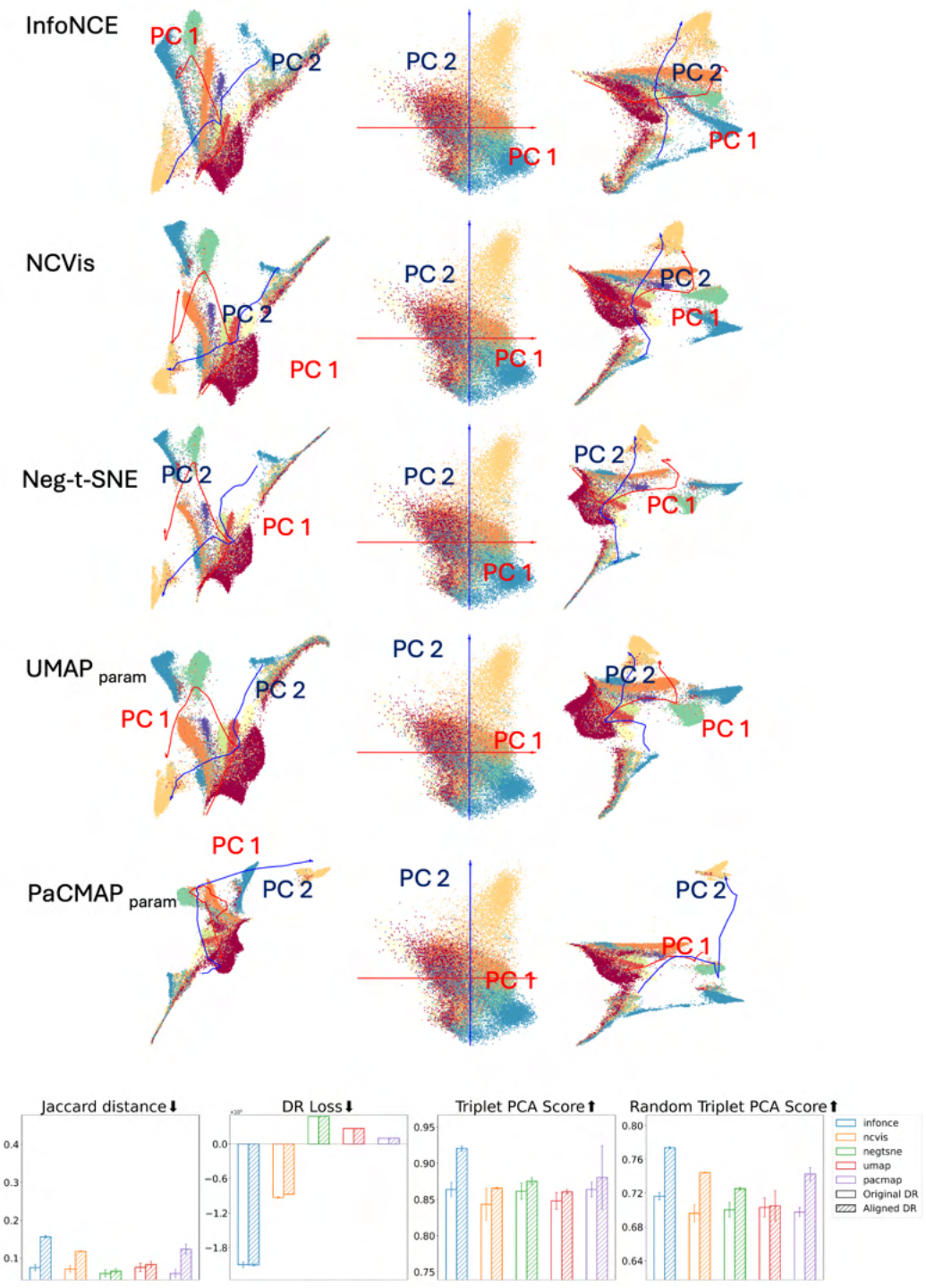

Figure 9: Comparison of original Human Cortex embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure. We show that soft Jaccard distance and $\mathcal{L}_{\text{DR}}$ (bottom) remain mostly unchanged and that aligned embeddings consistently maintain structure. Random Triplet PCA score and Triplet PCA score have shown an improvement after the alignment.

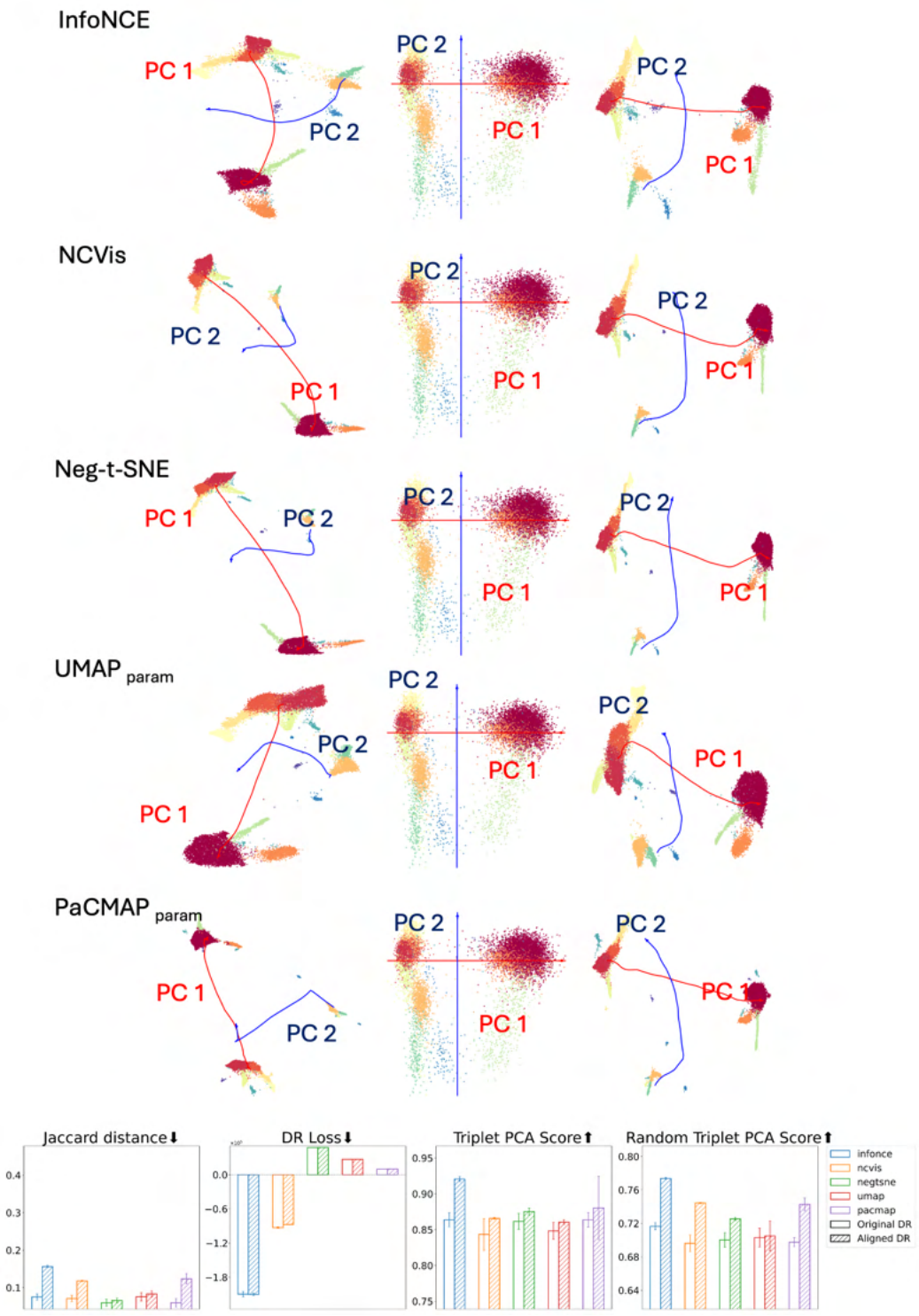

Figure 10: Comparison of original Kang et al. embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure. We show that soft Jaccard distance and $\mathcal{L}_{\text{DR}}$ (bottom) remain mostly unchanged and that aligned embeddings consistently maintain structure. Random Triplet PCA score and Triplet PCA score have shown an improvement after the alignment.

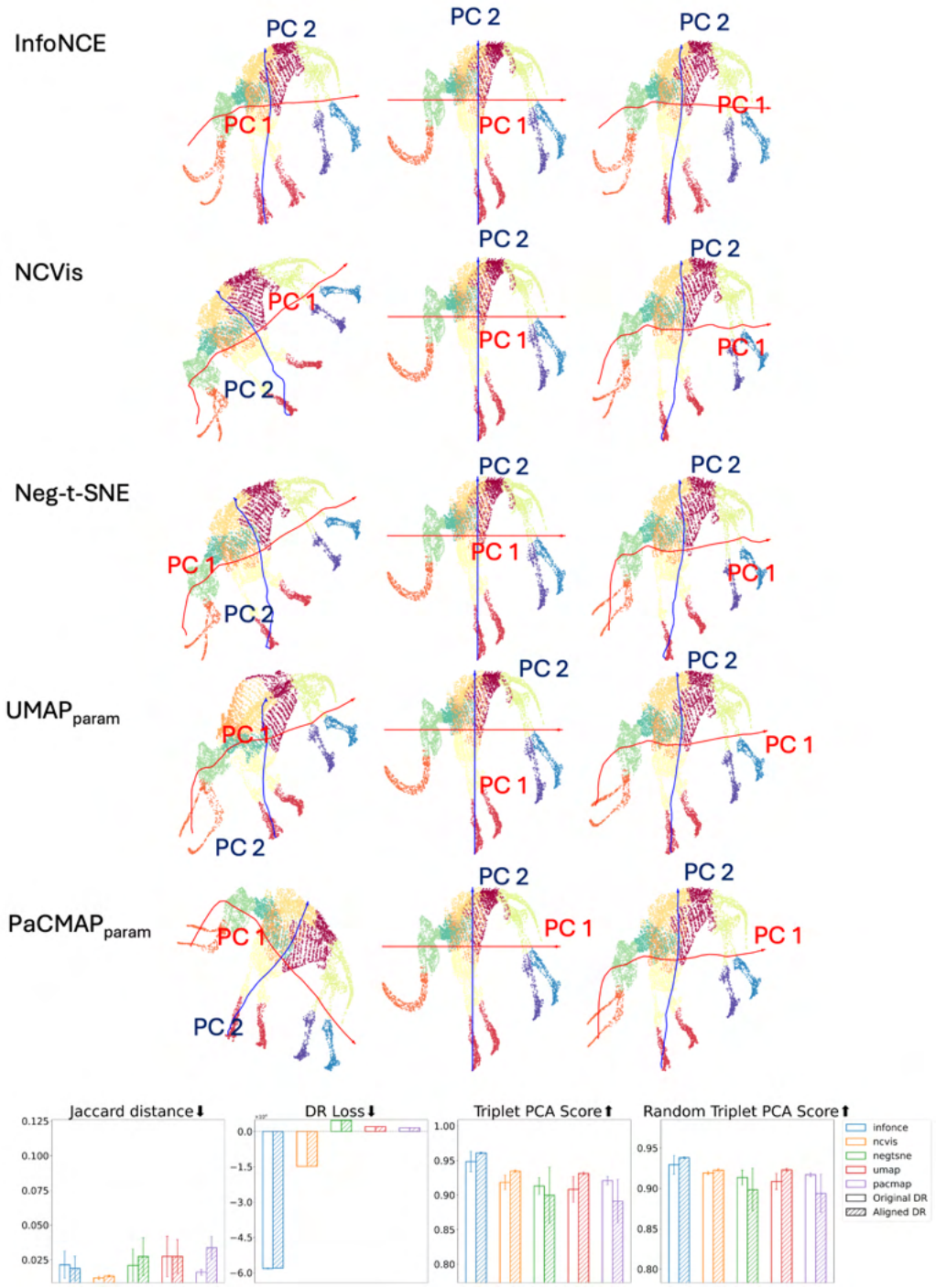

Figure 11: Comparison of original Mammoth embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure. We show that soft Jaccard distance and $\mathcal{L}_{\text{DR}}$ (bottom) remain mostly unchanged and that aligned embeddings consistently maintain structure. Random Triplet PCA score and Triplet PCA score have shown an improvement after the alignment.

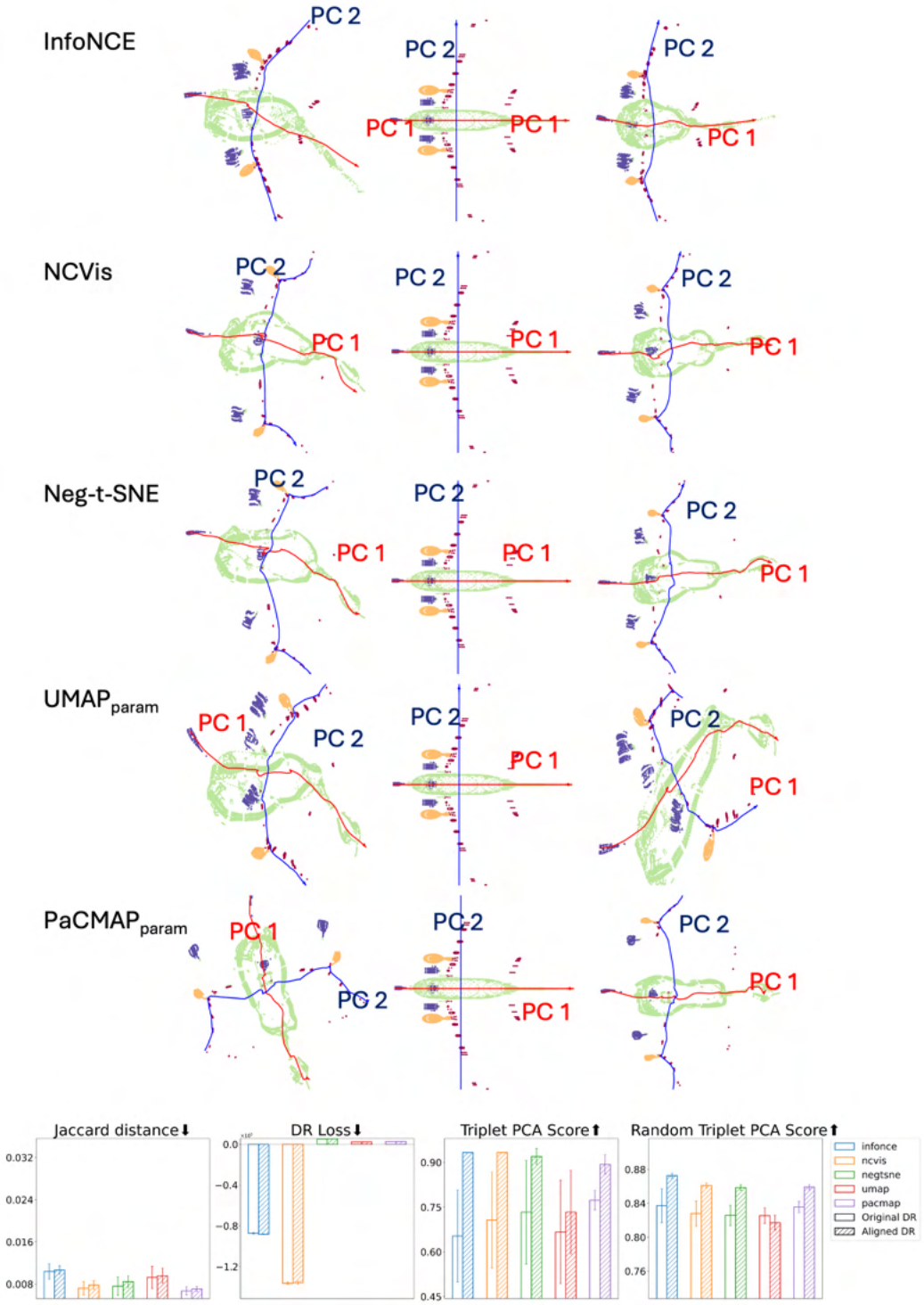

Figure 12: Comparison of original Airplane embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure. We show that soft Jaccard distance and $\mathcal{L}_{\text{DR}}$ (bottom) remain mostly unchanged and that aligned embeddings consistently maintain structure. Random Triplet PCA score and Triplet PCA score have shown an improvement after the alignment.

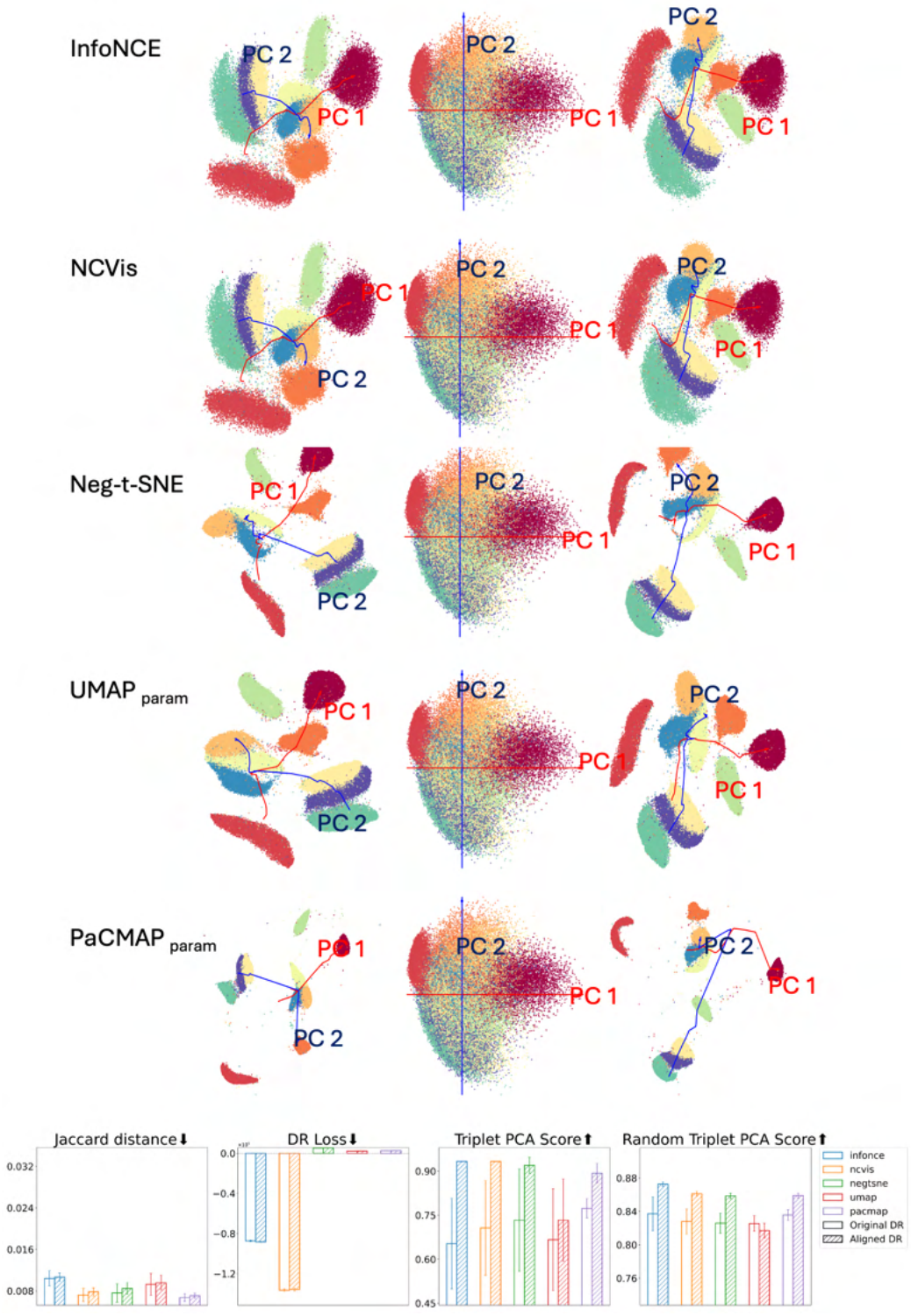

Figure 13: Comparison of original MNIST embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure. We show that soft Jaccard distance, and $\mathcal{L}_{\text{DR}}$ (bottom) remain mostly unchanged and that aligned embeddings consistently maintain structure.

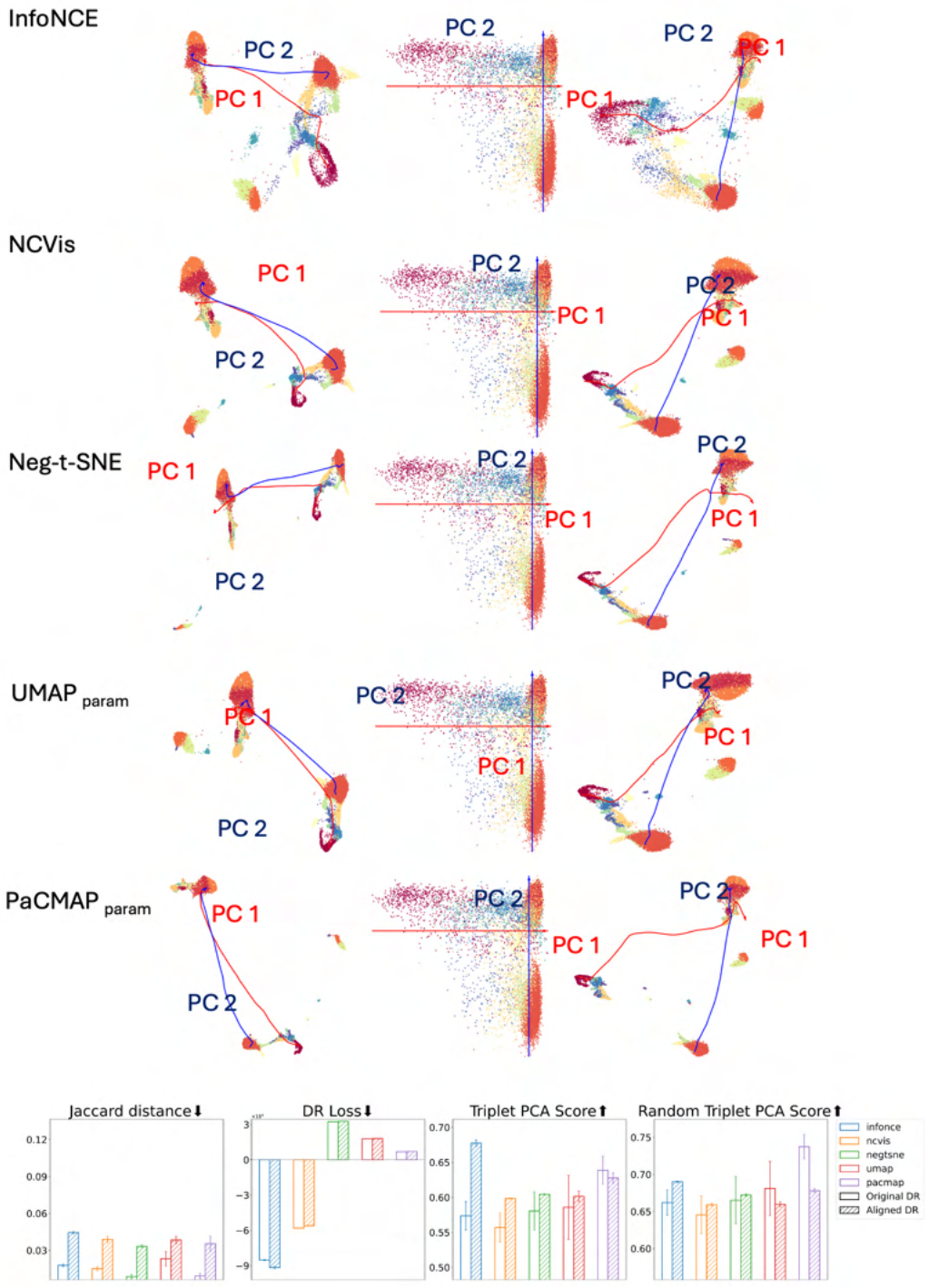

Figure 14: Comparison of original Stuart et al. embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure. We show that soft Jaccard distance and $\mathcal{L}_{\text{DR}}$ (bottom) remain mostly unchanged and that aligned embeddings consistently maintain structure. Random Triplet PCA score and Triplet PCA score have shown an improvement after the alignment.

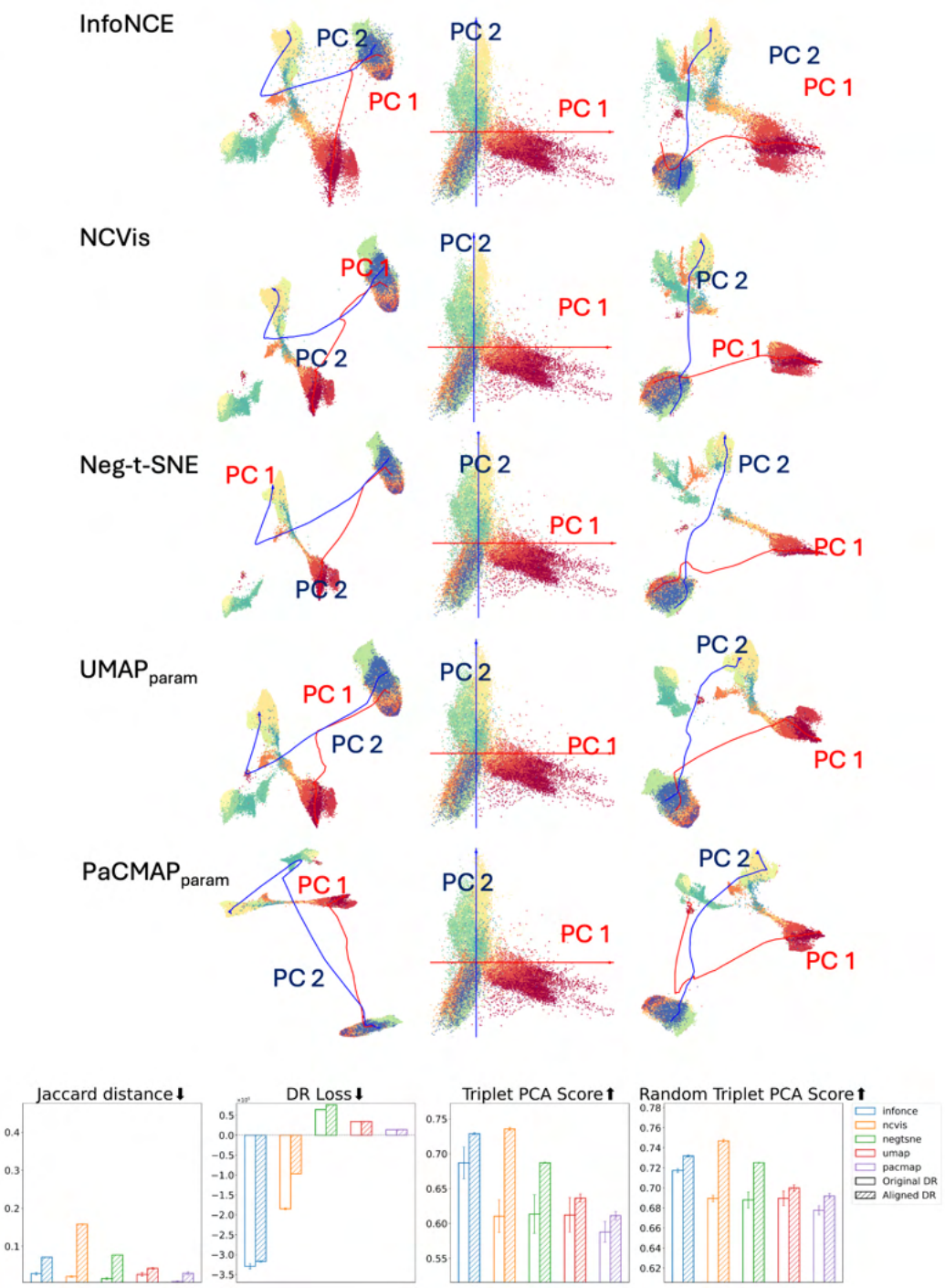

Figure 15: Comparison of original CBMC embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure. We show that soft Jaccard distance, triplet PCA score, random triplet PCA score, and $\mathcal{L}_{\text{DR}}$ (bottom) remain mostly unchanged and that aligned embeddings consistently maintain structure.

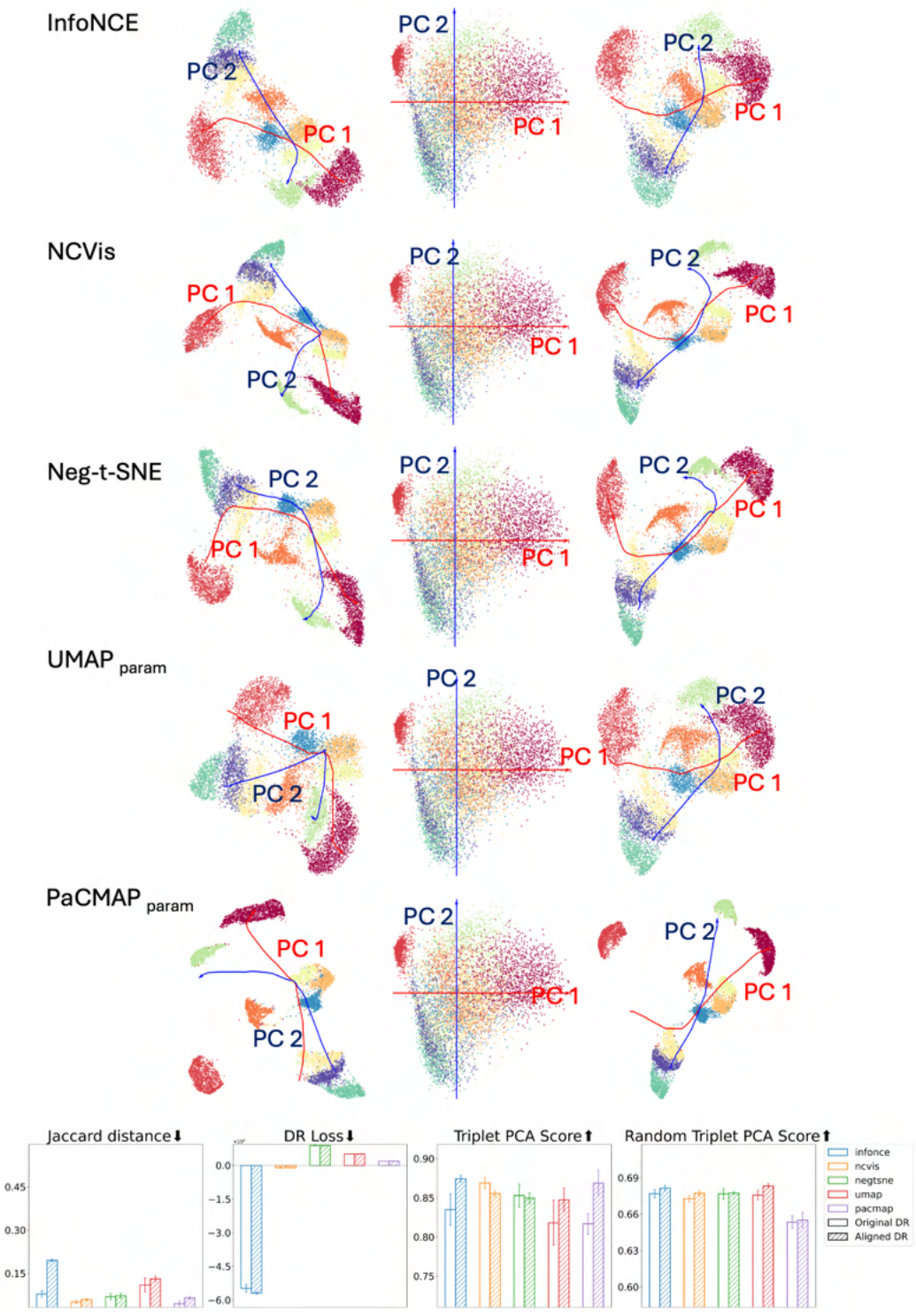

Figure 16: Comparison of original USPS embedding (left) PCA embedding (middle) and PCA informed embeddings (right) across different methods. We see alignment to principal components across all methods while preserving structure and $\mathcal{L}_{\text{DR}}\mathcal{L}_{\text{DR}}$ (bottom) remaining mostly unchanged and that aligned embeddings consistently maintain structure. Random Triplet PCA score and Triplet PCA score have shown an improvement after the alignment.

G Additional Concept-aware DR Results

Here we are showing a list of aligned and original DR embeddings across five methods—InfoNCE, NCVis, Neg-tSNE, UMAP, and PaCMAP – on a set of datasets that have been mentioned in Appendix D. Each subplot shows a 2D embedding colored by class label, with the left column representing original embeddings and the right column showing aligned counterparts. The alignment enforces concept-aware positioning while preserving local structure. The bar plots (bottom right) quantify performance using multiple metrics: 5-NN accuracy, soft Jaccard distance, triplet satisfaction, random triplet discrimination, silhouette score, and $\mathcal{L}_{\text{DR}}$. Aligned DR embeddings consistently maintain structure quality across all metrics, indicating enhanced trustworthiness and interpretability.

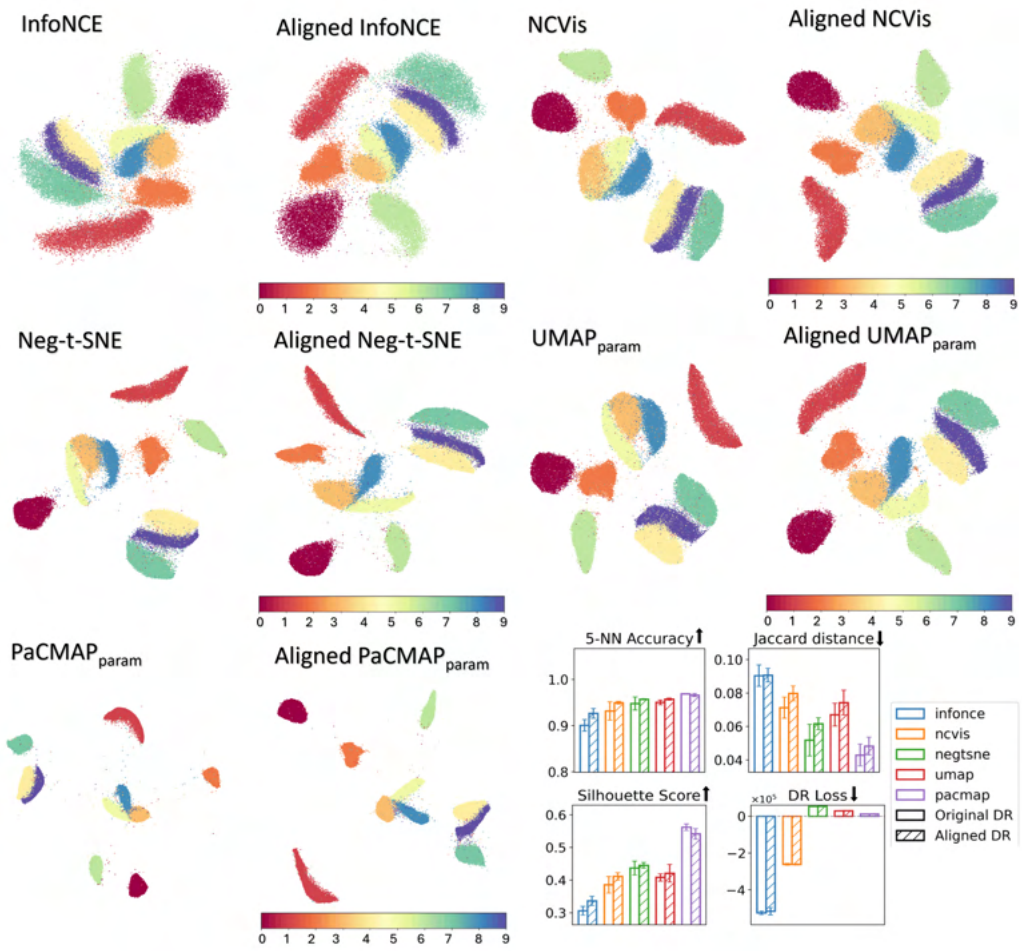

Figure 17: Comparison of original and aligned embeddings for MNIST using a concept-aware regularizer. The embeddings' horizontal axis is aligned with the digit index and embedding evaluation metrics haven't changed significantly after alignment.

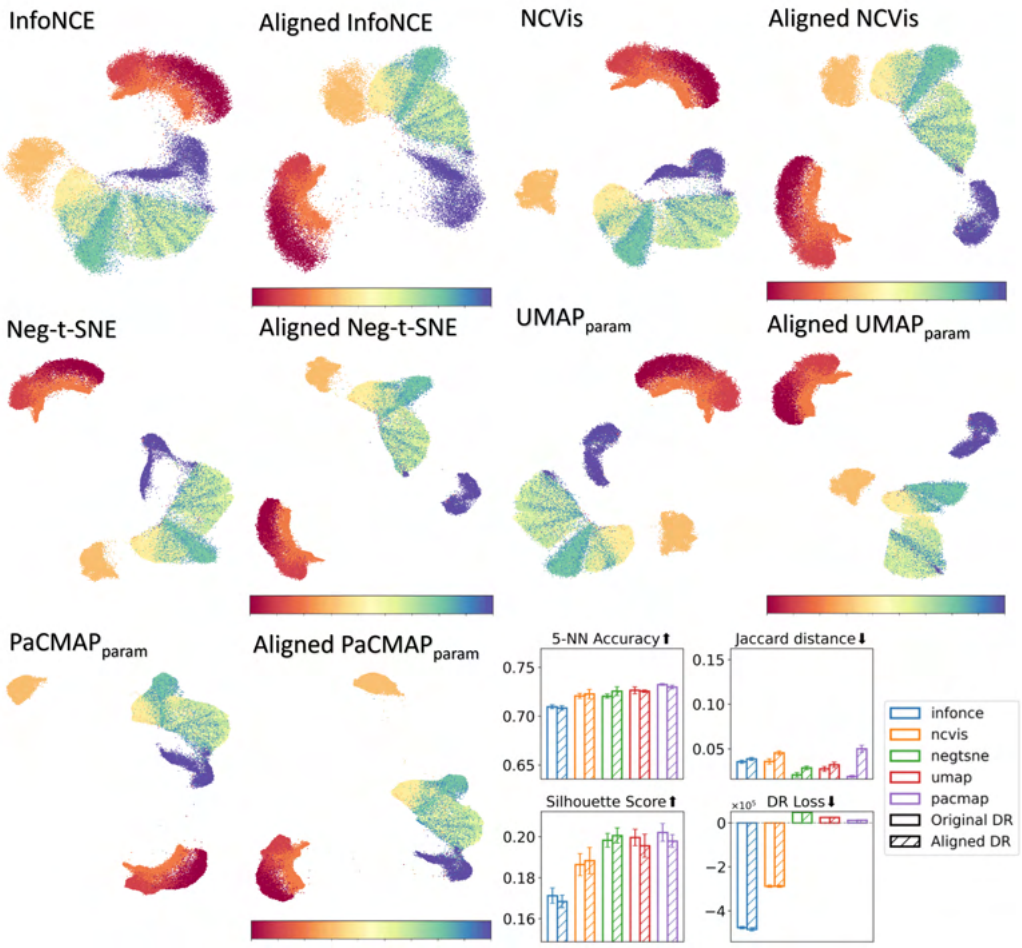

Figure 18: Comparison of original and aligned embeddings for FMNIST using a concept-aware regularizer. The embeddings' horizontal axis is aligned with the label index (from head to toe as mentioned in Section 6.1) and embedding evaluation metrics haven't changed significantly after alignment.

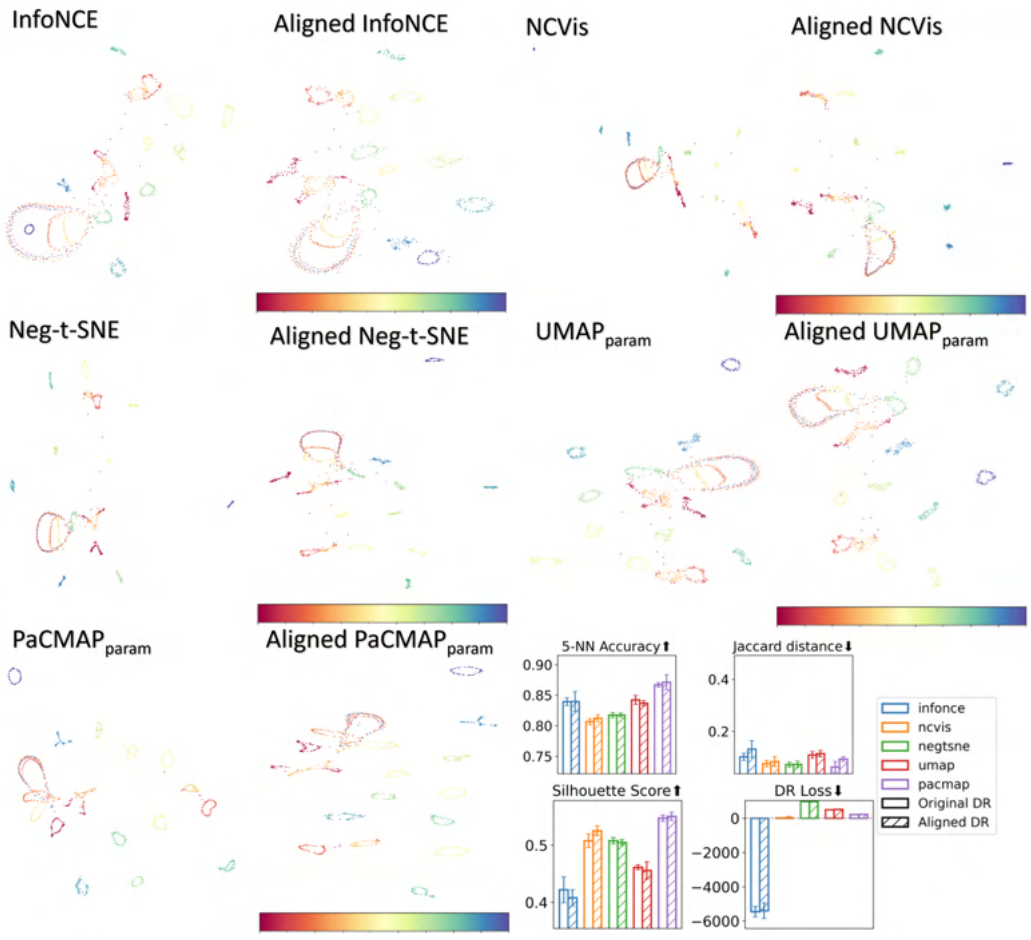

Figure 19: Comparison of original and aligned embeddings for COIL20 using a concept-aware regularizer. The embeddings' horizontal axis is aligned with the label index and embedding evaluation metrics haven't changed significantly after alignment.

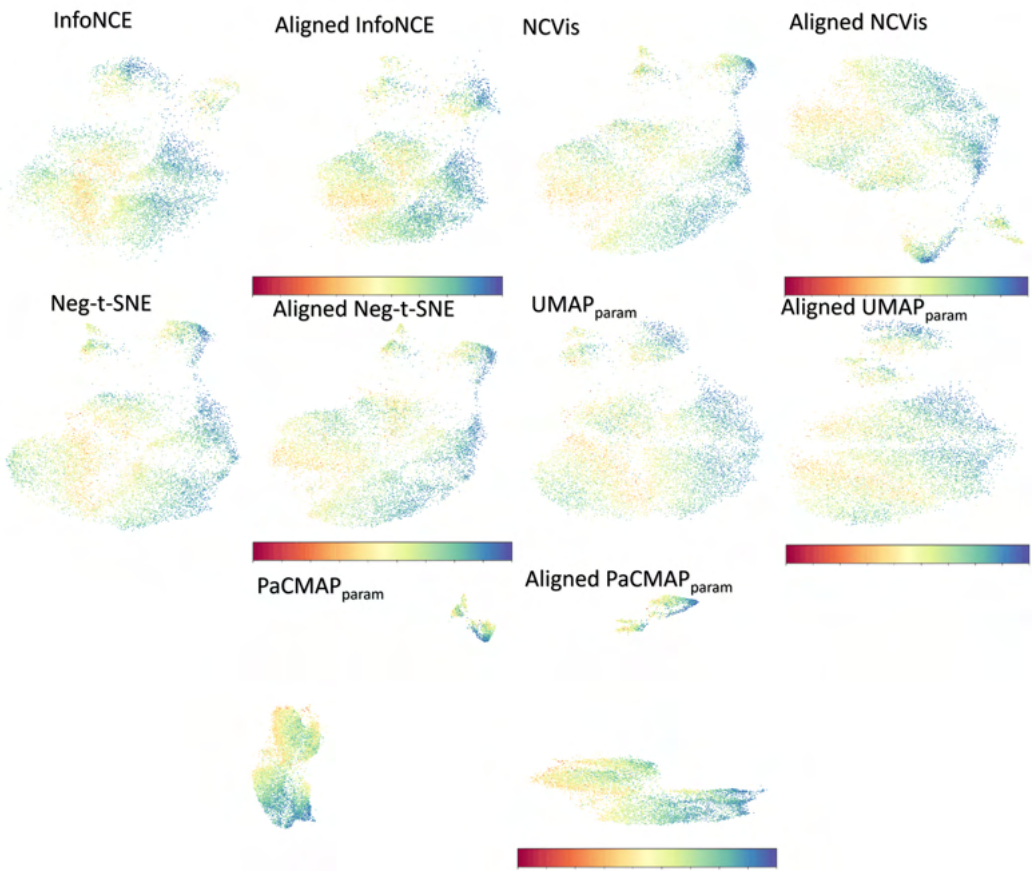

Figure 20: Comparison of original and aligned embeddings for FICO using a concept-aware regularize. The embeddings' horizontal axis is aligned with the external risk score. Since the label is a continuous feature, we are not displaying barplot here.

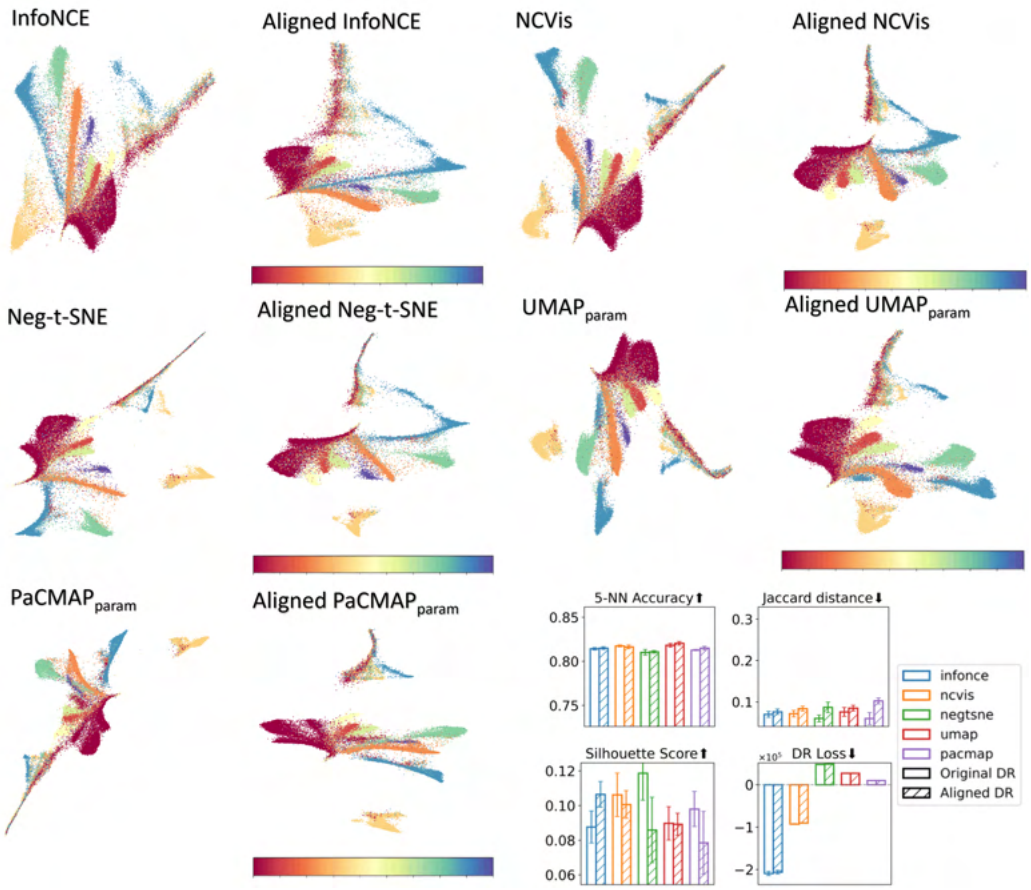

Figure 21: Comparison of original and aligned embeddings for Human Cortex Single Cell dataset using a concept-aware regularizer. The embeddings' horizontal axis is aligned with the cell type index, and the embedding performance hasn't been changed significantly.

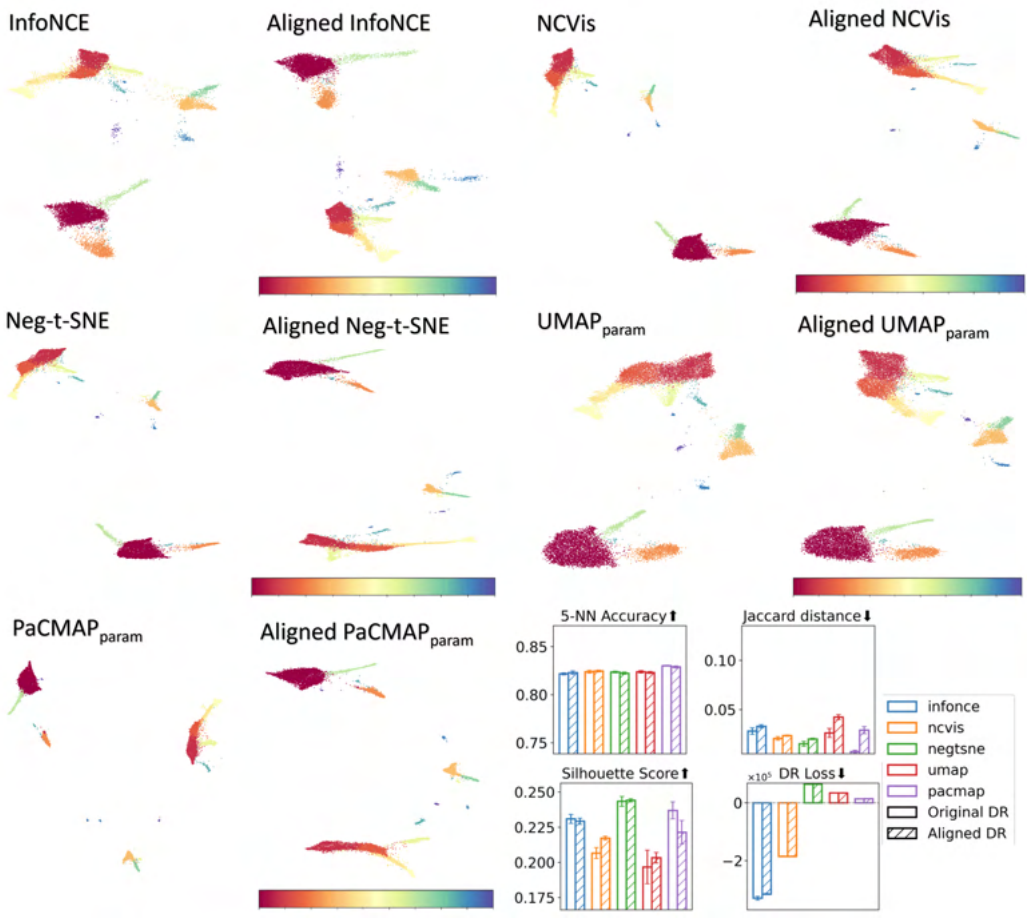

Figure 22: Comparison of original and aligned embeddings for Kang et al. dataset using a concept-aware regularizer. The embeddings' horizontal axis is aligned with the cell type index, and the embedding performance hasn't been changed significantly.

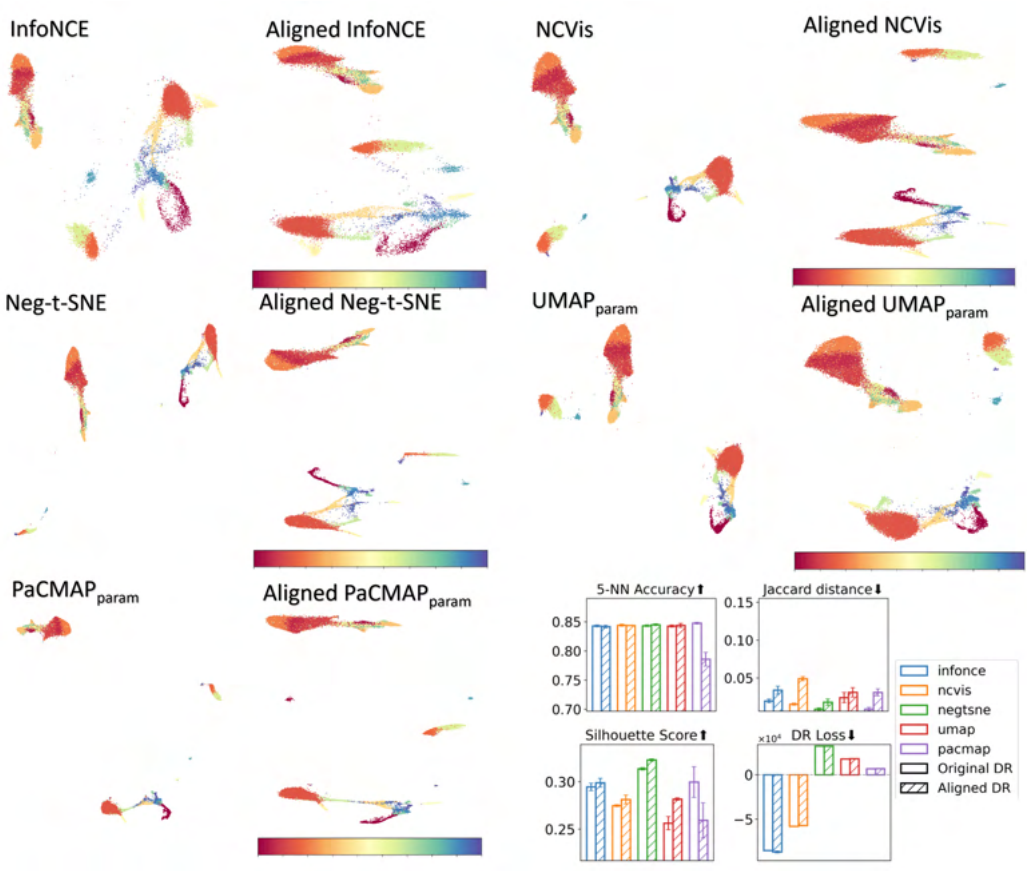

Figure 23: Comparison of original and aligned embeddings for the Stuart dataset using a concept-aware regularizer. The embeddings' horizontal axis is aligned with the cell type index, and the embedding performance hasn't been changed significantly.

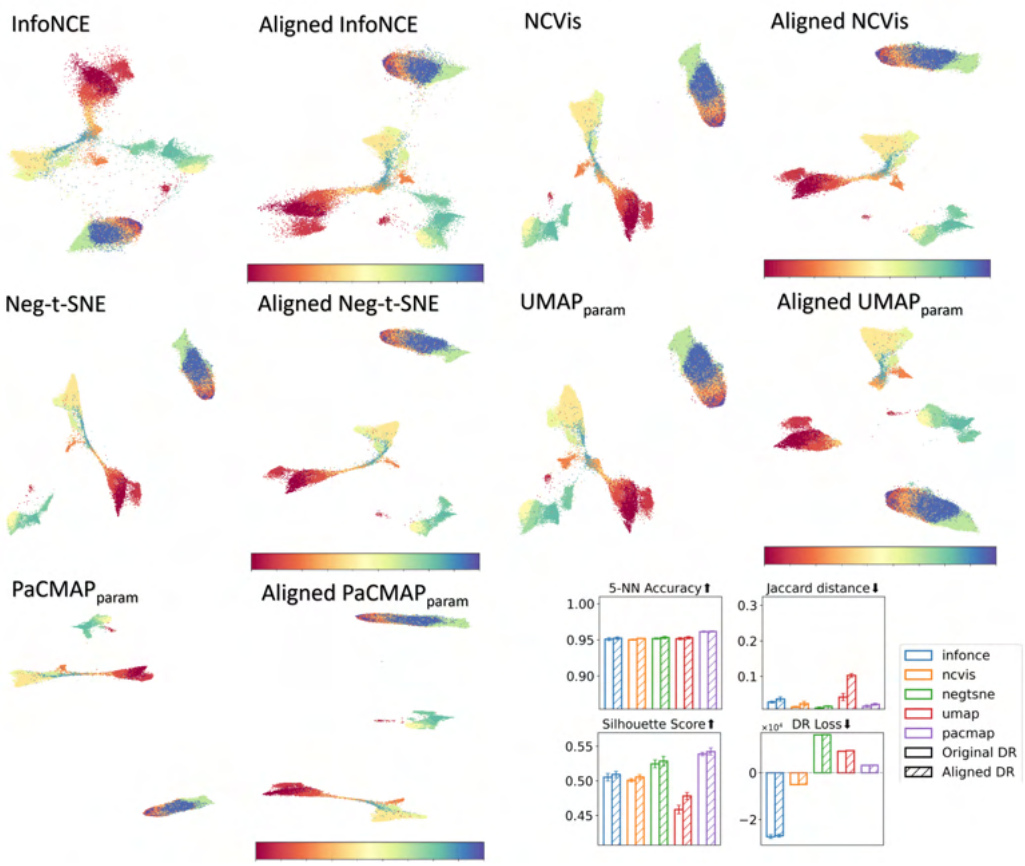

Figure 24: Comparison of original and aligned embeddings for the CMBC dataset using a concept-aware regularizer. The embeddings' horizontal axis is aligned with the cell type index, and the embedding performance hasn't been changed significantly.

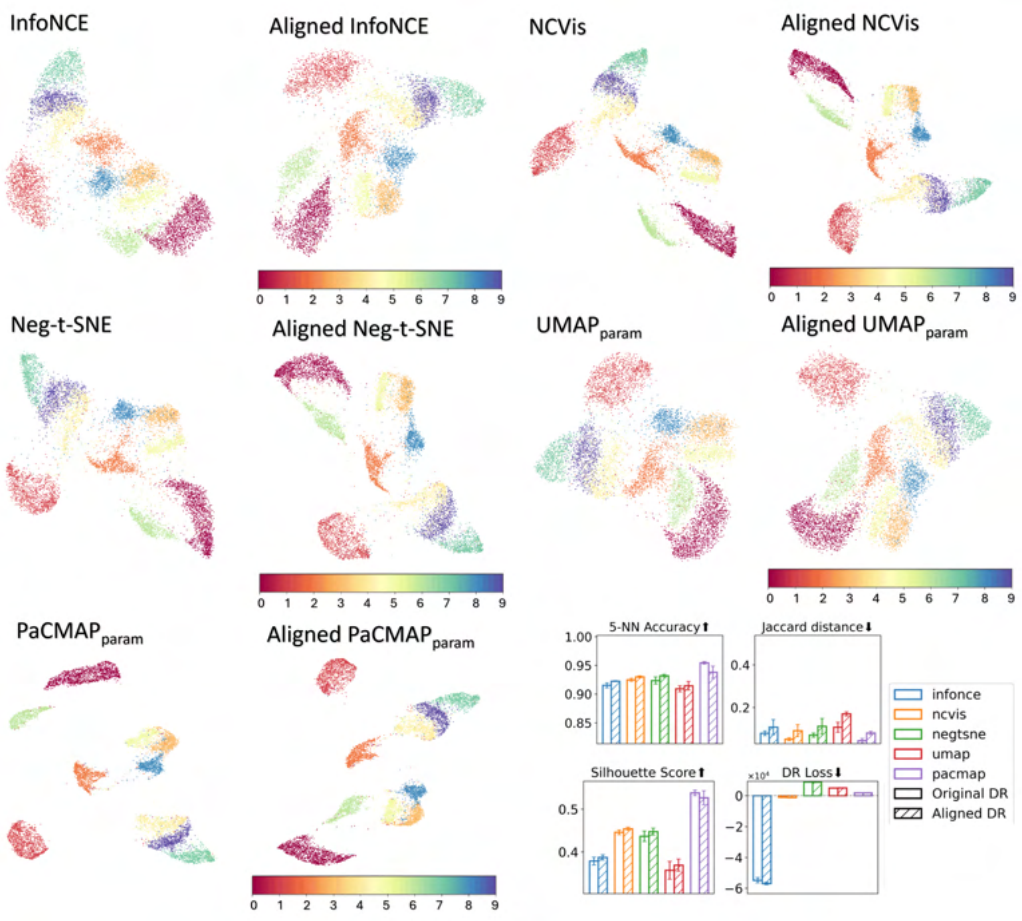

Figure 25: Comparison of original and aligned embeddings for USPS dataset using a concept-aware regularizer. Clusters are shown to align with the digits index and structure preservation across multiple DR methods.

H Influence of Missingness Ratio and Label Weights on Embedding Structure

Figure 26 illustrates how the embedding evolves across varying missingness ratios (rows) and label alignment weights (columns). From left to right, we increase the label weight λ_{Axis}, while from top to bottom, the missingness ratio increases. When the missingness ratio is low (top rows), moderate increases in λ_{Axis} produce embeddings that gradually align with the label order while preserving local structure. However, as the missingness ratio increases (moving down the rows), the embeddings become increasingly sensitive to the label weight. In particular, when both λ_{Axis} and missingness ratio are high (bottom-right corner), the embedding quality sharply degrades—the original structure breaks down, resulting in distorted or fragmented clusters. This shows that under sparse supervision, overly strong label alignment can dominate and disrupt the geometry learned from the intrinsic structure of the data.

Figure 27 illustrates the relationship between $\mathcal{L}_{\text{DR}}$ and the axis alignment weight λ_{Axis} under varying levels of missingness. As the missingness ratio increases, the threshold of λ_{Axis} at which the loss and distance scores begin to rise significantly becomes lower. This trend confirms our earlier qualitative observations: with higher missingness, the concept-based regularizer increasingly dominates the training objective, leading to a degradation of the embedding structure. This curve trend comparison between the$\mathcal{L}_{\text{DR}}$ and the Jaccard distance also shows that they have similar trend.

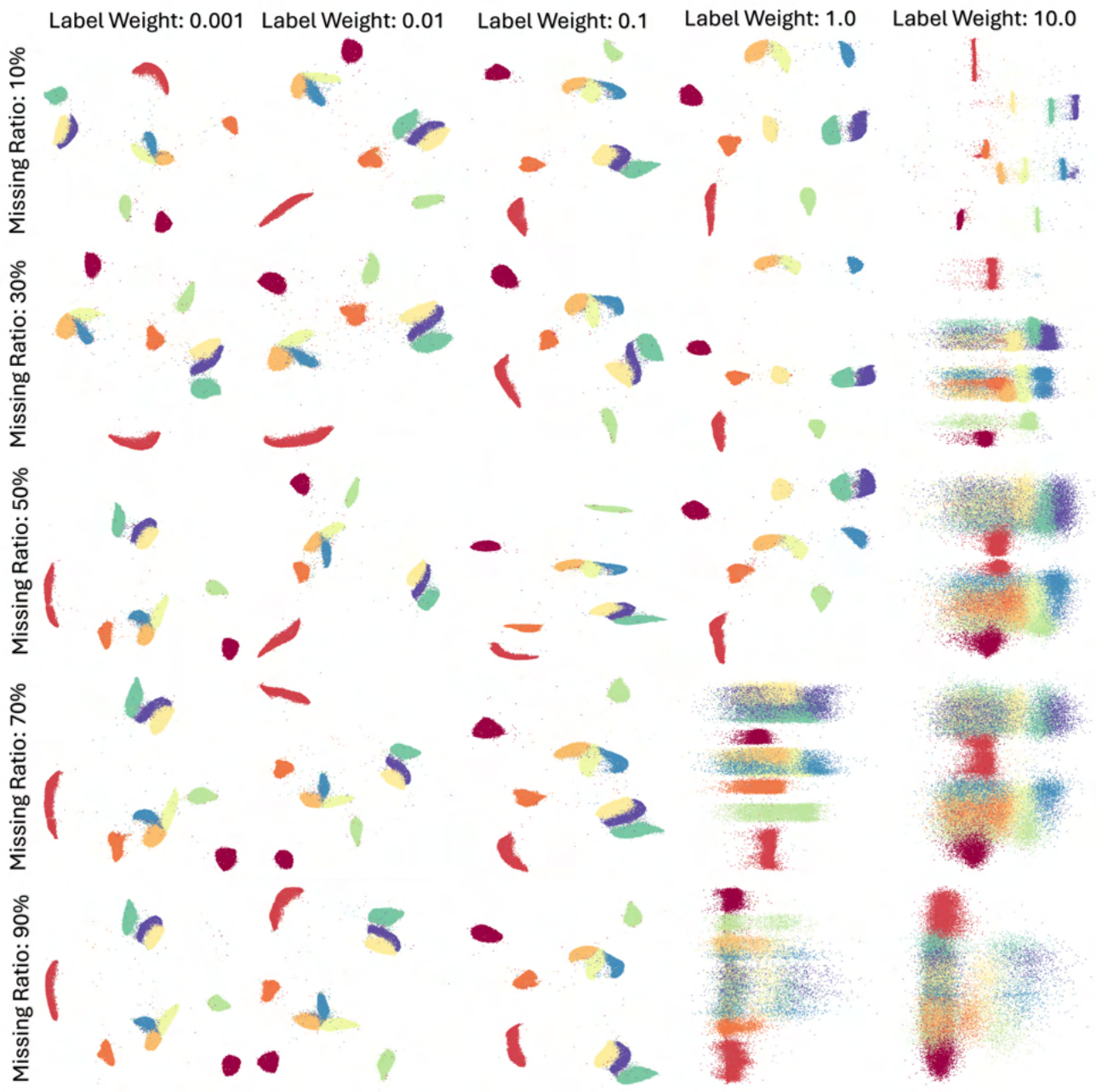

Figure 26: MNIST PaCMAP$_{\text{param}}$ Embedding under different label missingness ratios (rows) and label weights (columns). High label weight with high missingness ratio breaks the original structure.

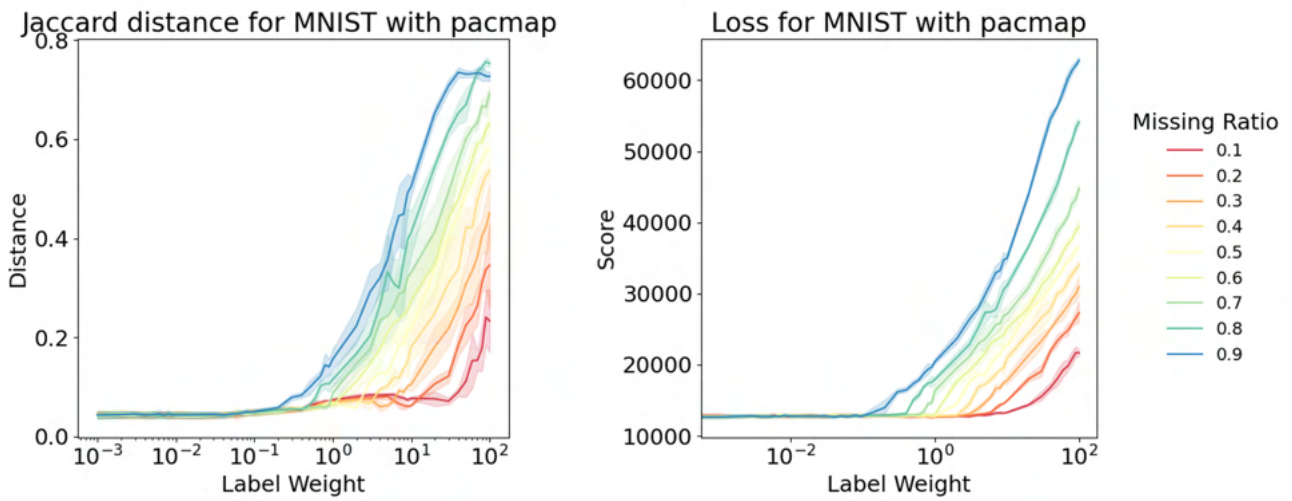

Figure 27: $\mathcal{L}_{\text{DR}}$ and Jaccard Distance under different label missingness ratios.

I Common Knowledge Extraction and Aggregation Results

To evaluate the effectiveness of our common knowledge extraction mentioned in Section 5, we compile results across different datasets, embedding methods, and three evaluation metrics, which are in the same format as they are shown in Section 6.2. The following figures present a comprehensive overview of how the proposed approach performs under varying data characteristics and dimensionality reduction settings. For each dataset, we report one of the embeddings obtained with different DR methods, along with relevant metrics: k-NN Accuracy, Silhouette Score, and SVM Accuracy. These results demonstrate the generalizability and robustness of our method across tasks.

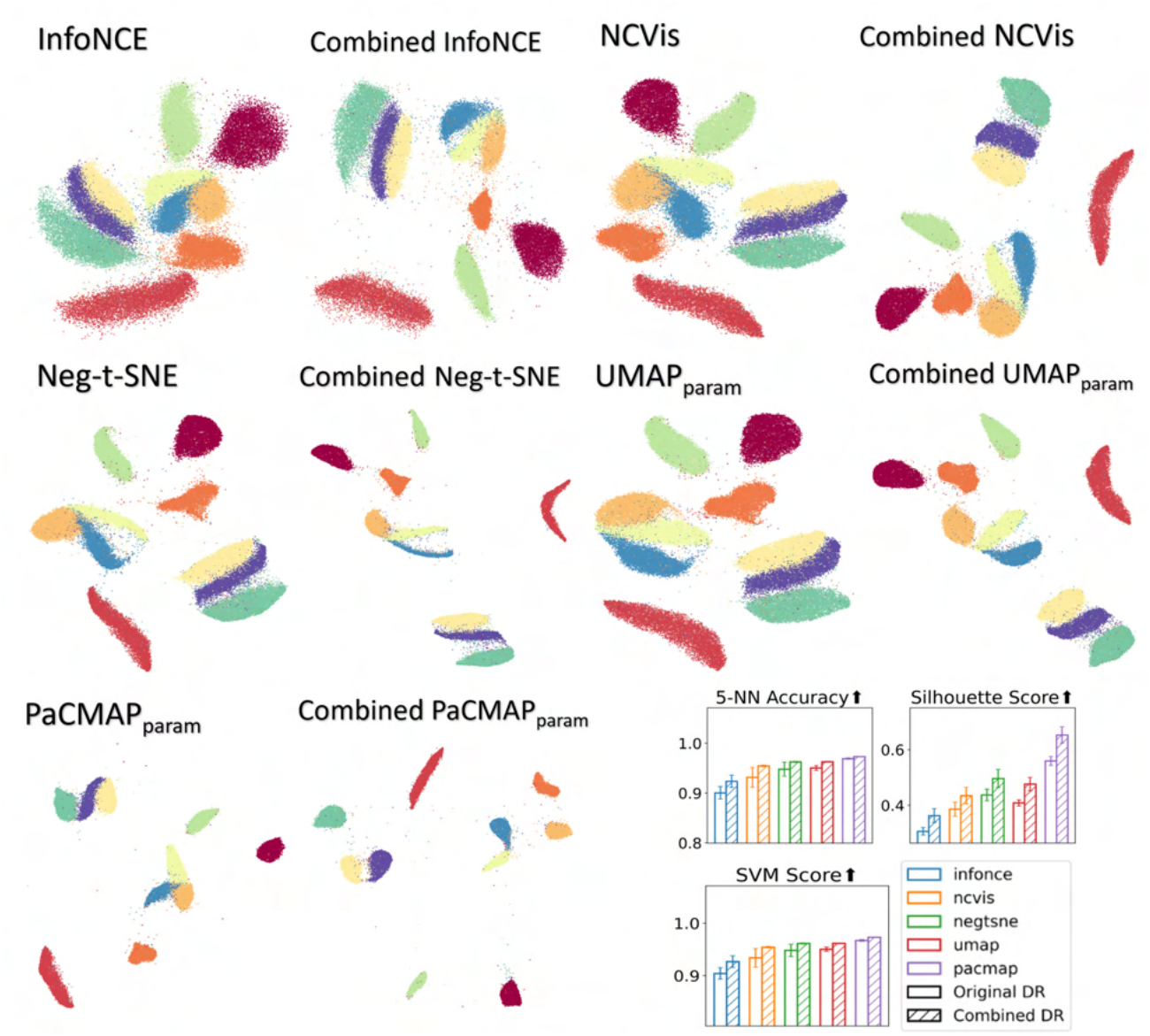

Figure 28: MNIST embeddings and improved embedding using common knowledge. Combined DR improves structure and boosts downstream metrics.

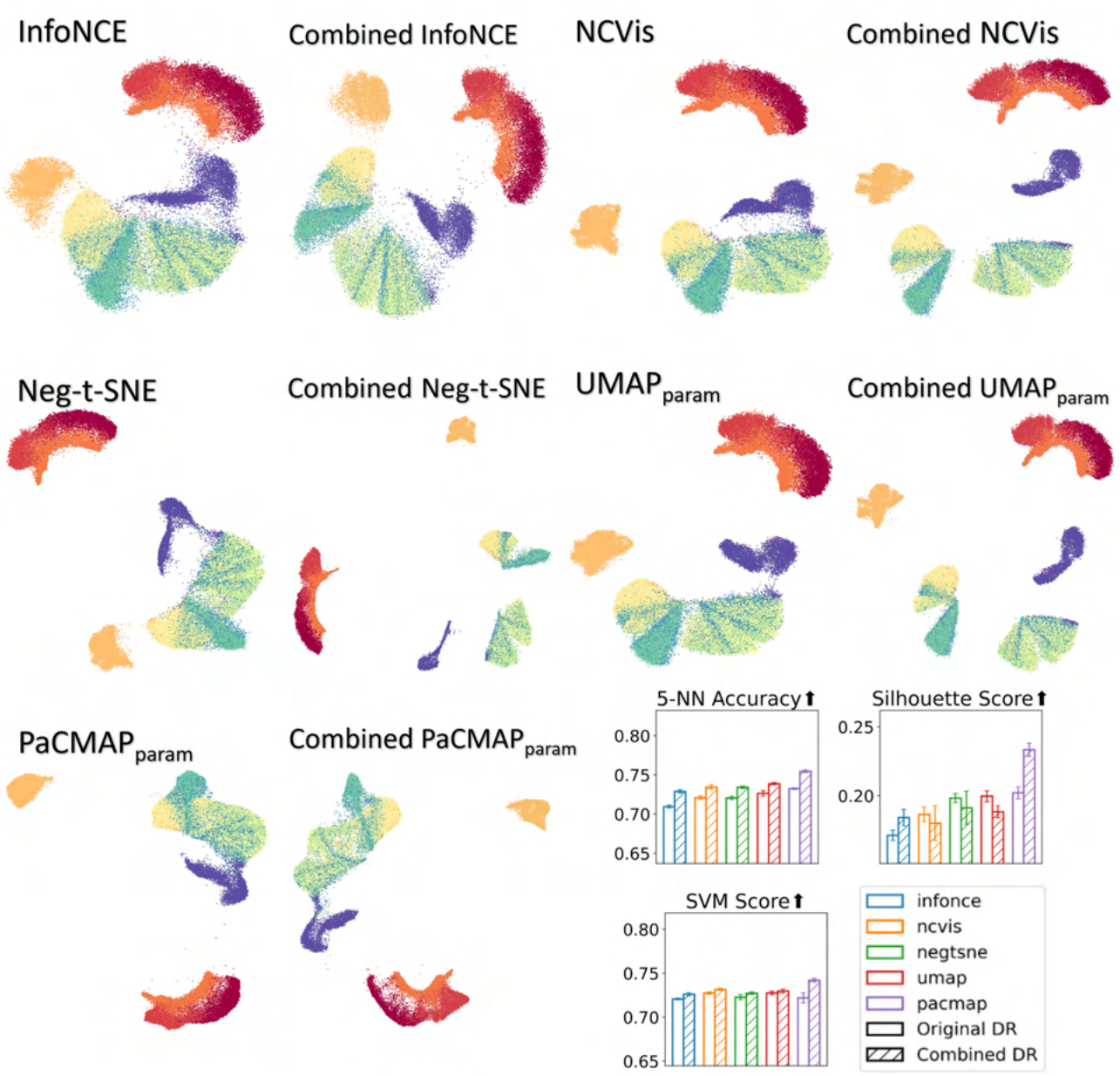

Figure 29: FMNIST embeddings and improved embedding using common knowledge. Combined DR improves structure and boosts downstream metrics. NCVis and Neg-t-SNE and UMAP are showing a bit lower silhouette score due to the mixed area at the bottom right corner

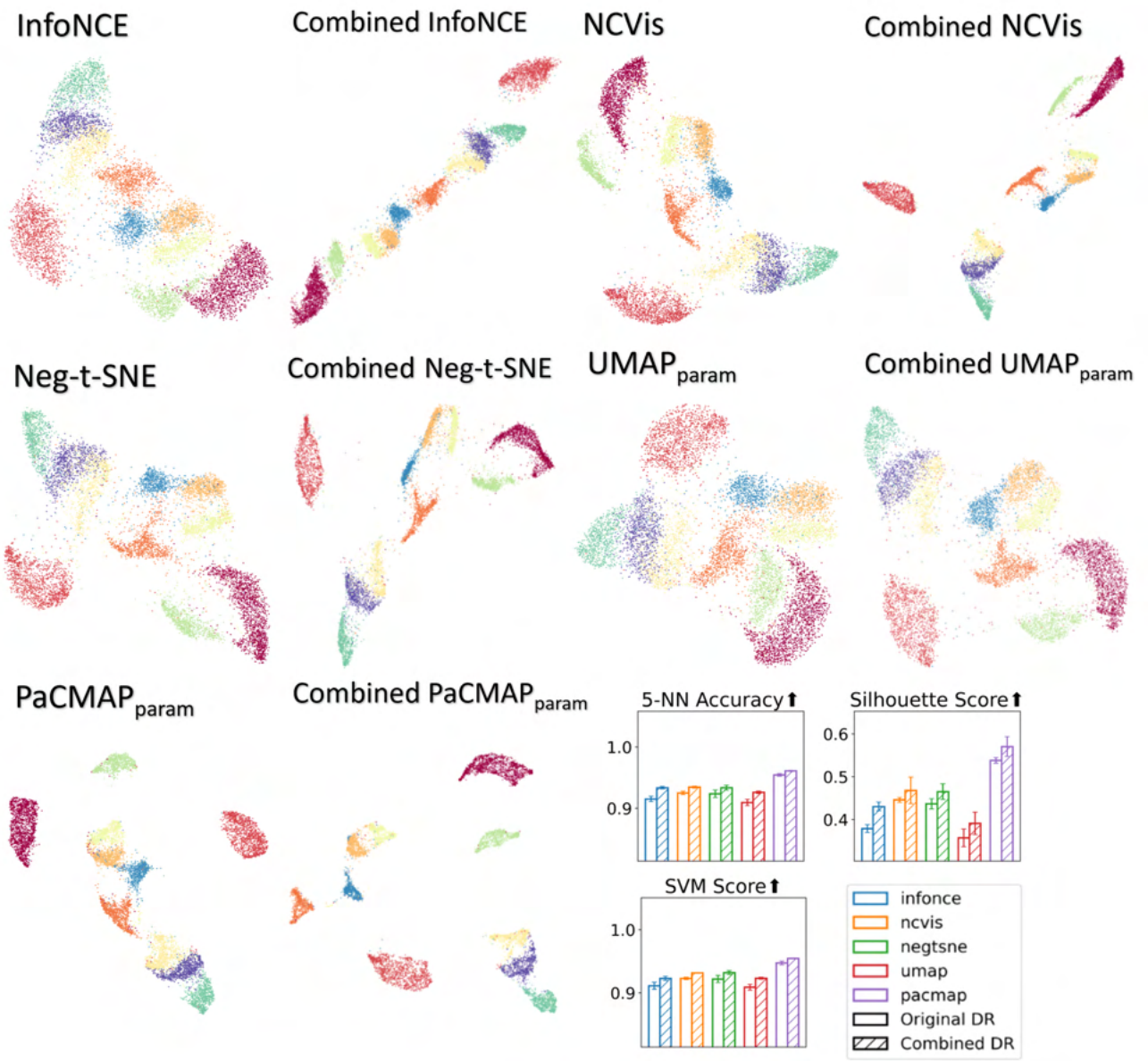

Figure 30: USPS embeddings and improved embedding using common knowledge. Combined DR improves structure and boosts downstream metrics.

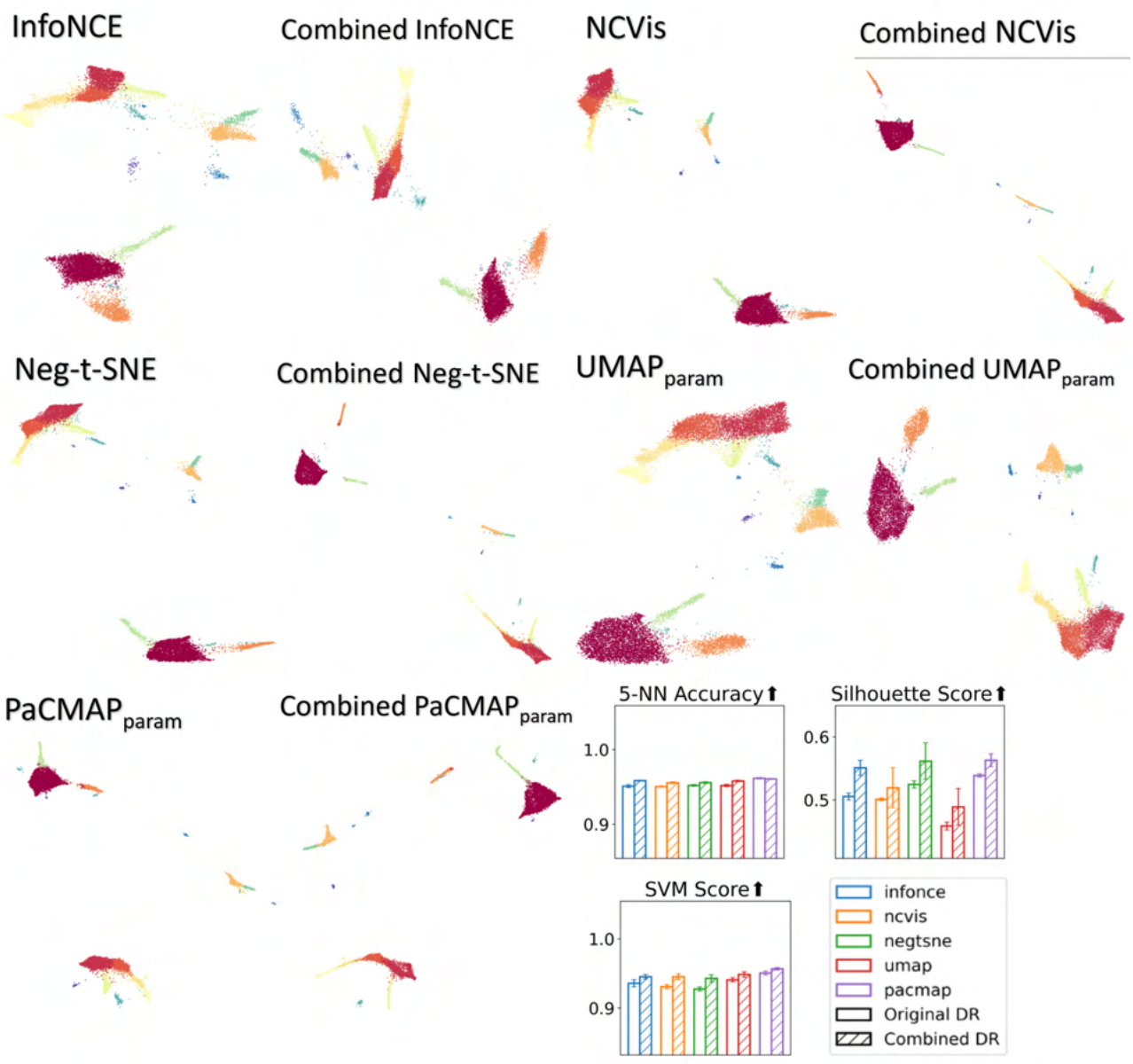

Figure 31: Kang et al. embeddings and improved embedding using common knowledge. Combined DR improves structure and boosts downstream metrics.

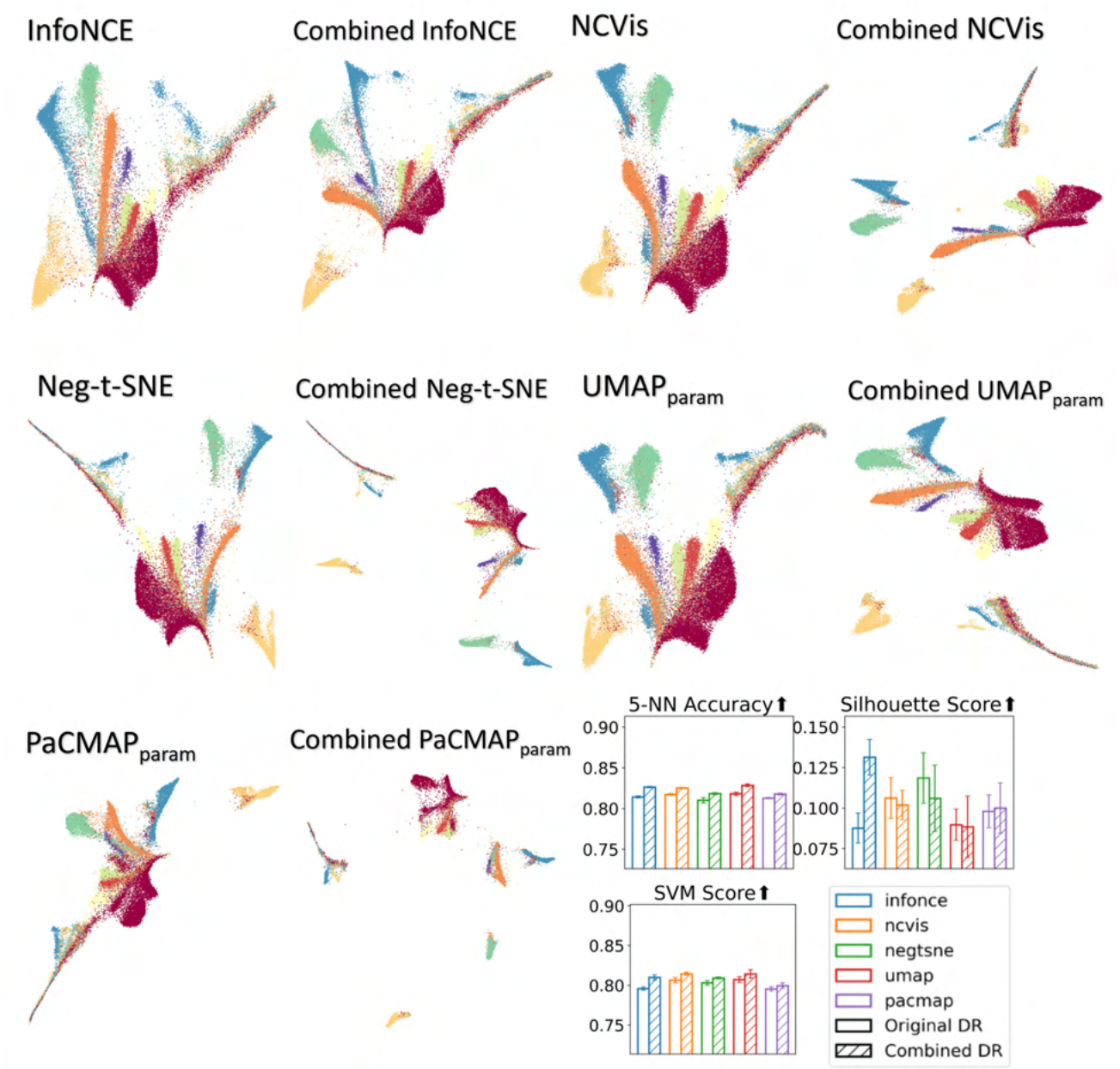

Figure 32: Human Cortex embeddings and improved embedding using common knowledge. Combined DR improves structure and boosts downstream metrics. NCVis and Neg-t-SNE and UMAP are showing a bit lower silhouette score due to the mixed area at the bottom right corner

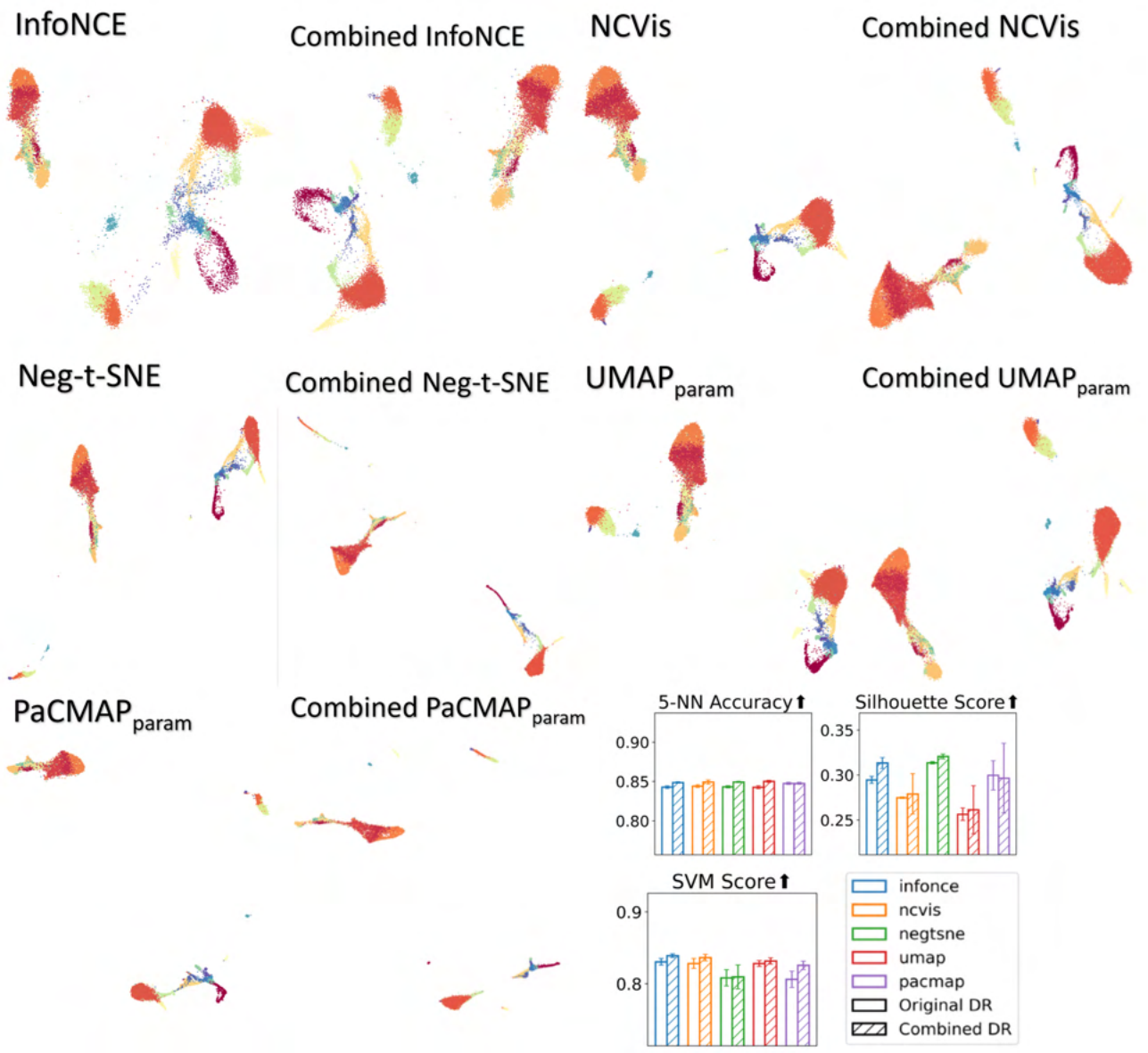

Figure 33: Stuart et. al. embeddings and improved embedding using common knowledge. Combined DR improves structure and boosts downstream metrics.

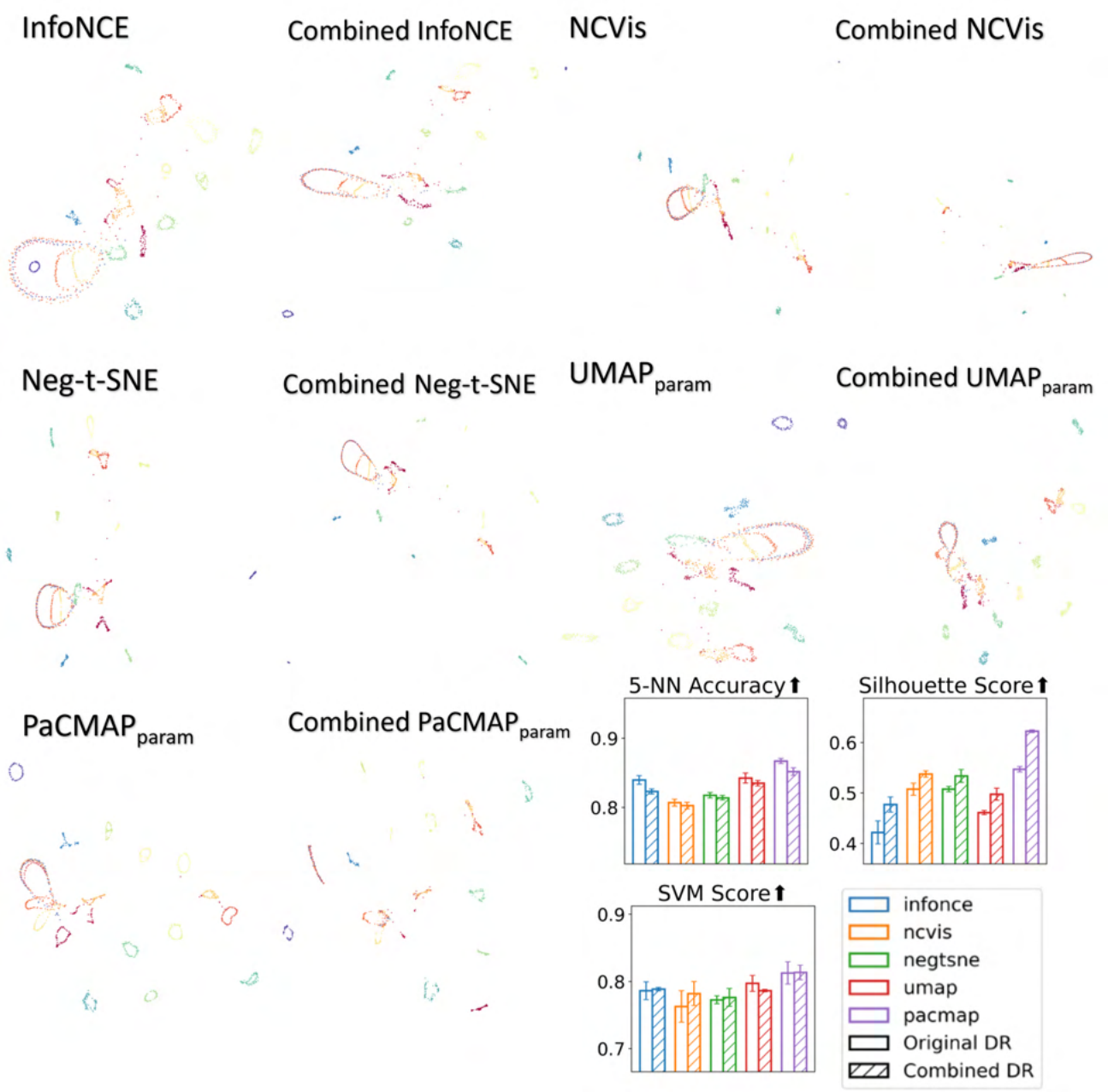

Figure 34: COIL20 embeddings and improved embedding using common knowledge. Combined DR improves structure and boosts downstream metrics.