The Shape of Attraction in UMAP: Exploring the Embedding Forces in Dimensionality Reduction

Mohammad Tariqul Islam *mhdtariq@mit.edu*
Media Lab, MIT
Electrical and Computer Engineering, Princeton University

Jason W. Fleischer *jasonf@princeton.edu*
Electrical and Computer Engineering, Princeton University

Reviewed on OpenReview: *https://openreview.net/forum?id=fdPNhqav5G*

Abstract

Uniform manifold approximation and projection (UMAP) is among the most popular neighbor embedding methods. The method samples pairs of point indices according to similarities in the high-dimensional space, and applies attractive and repulsive forces to their coordinates in the low-dimensional embedding. In this paper, we analyze the forces to reveal their effects on cluster formations and visualization, and compare UMAP to its contemporaries. Repulsion emphasizes differences, controlling cluster boundaries and inter-cluster distance. Attraction is more subtle, as attractive tension between points can manifest simultaneously as attraction and repulsion in the lower-dimensional mapping. This explains the need for learning rate annealing and motivates the different treatments between attractive and repulsive terms. Moreover, by modifying attraction, we improve the consistency of cluster formation under random initialization. Overall, our analysis provides a mechanistic understanding of UMAP and related embedding methods.

1 Introduction

Modern applications routinely generate high-dimensional data. Dimensionality reduction (DR) techniques have emerged as tools for exploratory analysis of such data by visualizing the underlying structure. The most popular methods, t-distributed stochastic neighbor embedding (Maaten & Hinton, 2008) and uniform manifold approximation and projection (UMAP) (McInnes et al., 2018) are grounded in the attraction-repulsion dynamics that bring similar data points closer while pushing dissimilar ones apart. As unsupervised algorithms, these do not rely on labeled data; instead, they identify and preserve the intrinsic structure of high-dimensional data by leveraging local (attractive) and global (repulsive) relationships (forces). This makes these algorithms particularly well-suited for tasks such as clustering (Becht et al., 2019), exploratory data analysis (Fleischer & Islam, 2020), anomaly detection in semiconductor manufacturing (Fan et al., 2021), visual search (González-Márquez et al., 2024), time series analysis (Altin & Cakir, 2024), studying representation convergence (Huh et al., 2024), and outlier image detection (Islam & Fleischer, 2024), where visualizing hidden patterns in unlabeled data is critical and meaningful. By learning the embeddings in a data-driven, label-free manner, DR exemplifies the power of unsupervised methods to distill complex data into easily interpretable forms.

Building upon the attraction-repulsion principle, newer methods have emerged (Amid & Warmuth, 2019; Agrawal et al., 2021; Wang et al., 2021; Narayan et al., 2021; Yang et al., 2022; Wang et al., 2025; Kury et al., 2026), each designed to emphasize specific aspects of the data. Despite their relevance in diverse applications, these methods often rely on heuristic practices that may fail to give meaningful interpretations. Moreover, DR introduces distortions that are unavoidable (Chari & Pachter, 2023). Thus, it is imperative to have a deeper understanding of the algorithms so that practitioners can provide better interpretations of

the embeddings, avoid spurious structures, and optimize performance. In practice, these algorithms achieve compact clusters using a variety of techniques, including specific initialization, learning rate schedule, and kernel function tuning. However, the underlying dynamics of the attractive and repulsive forces, responsible for cluster formation, have not been thoroughly investigated. Furthermore, the essential tunable parameters are concealed within abstract functional forms, making it harder to explain the algorithms.

In this paper, we decompose the forces into their constituent parts and extract the functional shapes of attraction and repulsion. We find that the necessity of learning rate annealing, the challenge of providing consistent output under random initialization, and the origin of cluster formation rely on attraction. Repulsive forces primarily govern inter-cluster distances. Our specific contributions are:

1. We formulate attraction and repulsion shapes from the attractive and repulsive forces, establish the conditions for contraction and expansion of distance, and provide a fresh perspective for these algorithms (Section 4).
2. We show that the attraction shape of UMAP causes the counterintuitive concept of both contraction and expansion of distance. Comparing attraction shapes of different algorithms, we discuss how attraction influences the learning rate annealing scheme (Section 5).
3. We compare attraction and repulsion shapes of UMAP, Parametric UMAP, and NEG-t-SNE, unveil the similarities and distinctions among them, and characterize the stability of the algorithms (Section 5.1).
4. We modify the attraction shape to improve the consistency of embedding under random initialization. This indicates the encoding of a unique structure (Section 5.2).
5. Analyzing repulsion shapes, we provide a deeper understanding of cluster formation and regulating inter-cluster distance (Section 5.3).

We center the main text on UMAP, now a de facto standard in many fields, and bring in other algorithms as needed for comparison and context. We organize the paper as follows. Section 2 reviews related work. Section 3 introduces the dimensionality reduction algorithm and fixes notation. In Section 4, we develop the tools needed to characterize attraction and repulsion shapes. Section 5 then analyzes UMAP through the lens of these shapes. Finally, Section 6 concludes with a discussion and directions for future work.

The datasets used in the paper are described in Appendix A. The metrics used to quantify embedding quality are described in Appendix D. For a detailed discussion of algorithms other than UMAP, see Appendix J. The codes used in the research are available at https://github.com/tariqul-islam/explaining_neighbor_embedding.

2 Related Works

The origin of modern iterative graph-based neighbor embedding algorithms can be traced to stochastic neighbor embedding (SNE) (Hinton & Roweis, 2002) and its extension using the t-distribution (t-SNE) (Maaten & Hinton, 2008). Both methods use a dense graph in which each point in a dataset has a pairwise relation with all the others, regardless of whether they are similar to each other or not. Moreover, the weights of the graphs are normalized to give a notion of probability distribution. Other concurrent methods, including locally linear embedding (Roweis & Saul, 2000) and Laplacian Eigenmaps (LE) (Belkin & Niyogi, 2002), used a k-nearest neighbor (k-NN) graph of pairwise interaction. Known as spectral methods, these algorithms rely on Eigenvalue decomposition. Subsequent work by Tang et al. (Tang et al., 2016) incorporated the k-NN graph in the iterative approach and removed normalization in the lower dimension. This approach was further extended by McInnes et al. (2018) in UMAP, where the normalization step was removed altogether (both in high and low dimension) and the embedding was obtained using pairwise interactions alone. The optimization steps use an explicit attractive force to preserve the local neighborhood and a repulsive force to keep dissimilar points apart. Building on these foundations, methods such as PaCMAP (Wang et al., 2021) and NEG-t-SNE (Damrich et al., 2023) have been proposed. For a recent survey of methods, see de Bodt et al. (2025).

There has been considerable progress in understanding and explaining the relationship among these algorithms. An early analysis of SNE found that if the data is well-clustered in the original space, then they are well-clustered in the embedding space (Shaham & Steinerberger, 2017). A similar analysis for t-SNE by Linderman & Steinerberger (2019) showed that the number of clusters in the embedding space is a lower bound on

the number of clusters in the original space. This was followed up in further characterization (Arora et al., 2018; Cai & Ma, 2022; Linderman & Steinerberger, 2022). Since t-SNE and UMAP originate from the same underlying framework (but with drastically different visualizations), a major undertaking in the literature has been to find the connection between them (Böhm et al., 2022; Damrich et al., 2023; Draganov et al., 2023). Böhm et al. (2022) theorized that methods like Laplacian Eigenmap, ForceAtlas2 (Jacomy et al., 2014), UMAP, and t-SNE are all samples from the same underlying spectrum. Indeed, LE and t-SNE's are connected by the early exaggeration phase (Cai & Ma, 2022). The connection between UMAP and t-SNE can be related through contrastive estimation (Damrich et al., 2023). Around the same time, Hu et al. (2023) independently discovered the relation of contrastive learning and SNE. Recent approaches offer a probabilistic perspective (Ravuri et al., 2023; Ravuri & Lawrence, 2024), employ kernel techniques (Draganov & Dohn, 2023), and utilize information geometry (Kolpakov & Rocke, 2024) to explain dimensionality reduction.

3 Uniform Manifold Approximation and Projection

UMAP constructs a high-dimensional graph of the dataset $X = \{x_i \in R^n | i = 1, \dots, N\}$ by using a pairwise relation: $p_{i,j} = f_h(d(x_i, x_j))$, typically, $\in [0, 1]$, where, f_h is the high dimensional affinity function and $d(\cdot, \cdot)$ is a distance metric (for details, see Appendix K).

The graph of the low-dimensional representation $Y = \{y_i \in \mathbb{R}^d | i = 1, \dots, N\}$ is given by a differentiable function

$$q_{ij} = \frac{1}{1 + a(||y_i - y_j||_2^2)^b}, \tag{1}$$

where the parameters a and b determine the density of the mapping and are chosen by fitting q_{ij} to

$$\Psi(||y_i - y_j||_2) = \begin{cases} 1 & \text{if } ||y_i - y_j||_2 < m_d \\ \exp(-(||y_i - y_j||_2 - m_d)) & \text{otherwise} \end{cases}, \tag{2}$$

where m_d regulates the distance between the two nearest low-dimensional points.
UMAP aims to minimize the following cross-entropy loss function:

$$\mathcal{L} = \sum_{i,j} (-p_{ij} \log(q_{ij}) - (1 - p_{ij}) \log(1 - q_{ij})). \tag{3}$$

The first term provides an attractive force and the second term provides a repulsive force. Instead of optimizing every point in each iteration, UMAP takes the negative sampling approach (Mikolov et al., 2013; Tang et al., 2016). For each edge with $p_{ij} > 0$, named a positive edge, several edges are sampled randomly, named negative edges. The attractive force is applied on the positive edge:

$$y_i^{t+1} = y_i^t + \lambda \nabla_{y_i^t} \log(q_{ij}), \tag{4}$$

$$y_j^{t+1} = y_j^t + \lambda \nabla_{y_j^t} \log(q_{ij}), \tag{5}$$

and the repulsive force is applied on the negative edges:

$$y_i^{t+1} = y_i^t + \lambda \nabla_{y_i^t} \log(1 - q_{ij}), \tag{6}$$

where $\lambda (> 0)$ is the learning rate and t is the step number. Note that y_j is not updated for negative edges. For a detailed analysis of UMAP's loss, see (Damrich & Hamprecht, 2021). Other algorithms follow a similar optimization scheme with different loss functions.

4 Attraction and Repulsion Shapes

The action of the updates (4-6) can be simplified by decomposing the gradients $\nabla_{y_i^t} \log(q_{ij})$ ($\nabla_{y_i^t} \log(1 - q_{ij})$) into a scalar coefficient dependent on the distance $||y_i - y_j||_2$ acting on the vector $(y_i - y_j)$. We call this

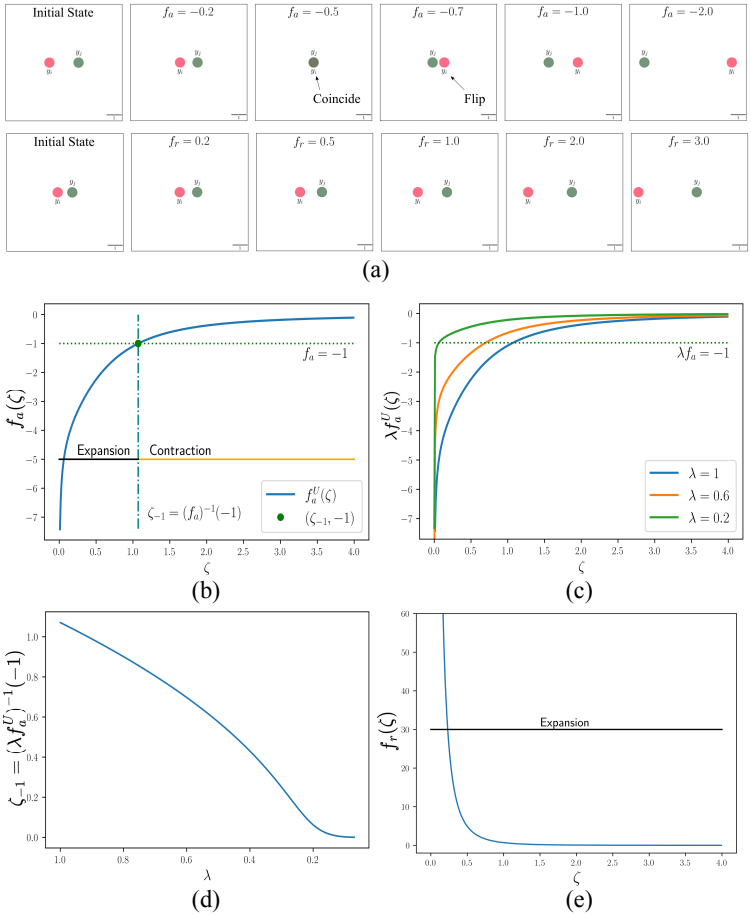


Figure 1: Attraction and repulsion shapes in UMAP. (a) Effect of different values of f_a (top) and f_r (bottom) on a pair. (b) Attraction shape of UMAP. (c) Attraction term scaled by the learning rate, λf_a, for various learning rates λ. (d) Minimum distance for contraction (ζ_{-1}) as λ decreases. (e) Repulsion shape of UMAP. Default UMAP parameters: $a = 1.58$ and $b = 0.89$.

scalar coefficient the attraction (repulsion) shape. While we use UMAP as a specific example, this formalism applies to any method that relies on attraction and repulsion.

By writing $\nabla_{y_i^t} \log(q_{ij}) = f_a(\zeta^t)(y_i^t - y_j^t)$, where $\zeta = ||y_i - y_j||_2$, we can update the equations of a positive edge as

$$y_i^{t+1} = y_i^t + \lambda f_a(\zeta^t)(y_i^t - y_j^t), \tag{7}$$
$$y_j^{t+1} = y_j^t - \lambda f_a(\zeta^t)(y_i^t - y_j^t). \tag{8}$$

Here, $f_a : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\leq 0}$ is the attraction shape, and we use the fact that for Euclidean metric, $\nabla_{y_i^t} \log(q_{ij}) = -\nabla_{y_j^t} \log(q_{ij})$. Similarly, we reformulate the update equation of a negative edge as

$$y_i^{t+1} = y_i^t + \lambda f_r(\zeta^t)(y_i^t - y_j^t), \tag{9}$$
$$y_j^{t+1} = y_j^t, \tag{10}$$

where $f_r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is the repulsion shape.

Such decomposition has appeared previously, e.g., in (McInnes et al., 2018; Agrawal et al., 2021; Draganov & Dohn, 2023), but their formulations and utilization vary. The original UMAP paper (McInnes et al., 2018) used it as a computational trick for fast processing, while Draganov & Dohn (2023) used it for comparing different

algorithms from the kernel perspective. In both cases, the primary focus was computing the derivative without any further analysis of the decomposition. A similar approach was used earlier by Agrawal et al. (2021), but they expressed the decomposition in terms of a scalar coefficient and a unit vector ($(y_i - y_j)/||y_i - y_j||_2$) to emphasize the magnitude of the forces. Here, we treat the decompositions as independent functions that can take various forms. The discussion below shows that our shape decomposition is more illuminating than the magnitude alone.

4.1 Conditions for Attraction and Repulsion

In this section, we establish the conditions of attraction and repulsion from the update equations (7)-(10). The following proposition characterizes the contraction of distance between the pair y_i^t and y_j^t of a positive edge:

Proposition 4.1. *The update Eqs. (7) and (8) provide a contraction of distance ($||y_i^{t+1} - y_j^{t+1}|| < ||y_i^t - y_j^t||$) if $-1 < \lambda f_a < 0$.*

Here, λf_a works as the effective attraction shape, since the contraction condition depends on the scaled quantity λf_a. In particular, scaling by λ can move the update across the critical threshold -1, changing attraction from contractive to expansive. In practice, once f_a is specified, λ can be chosen to place λf_a in a well-behaved regime.

For a negative edge, the following proposition characterizes the expansion of the distance between the pair y_i^t and y_j^t:

Proposition 4.2. *The update Eqs. (9) and (10) provide an expansion of distance ($||y_i^{t+1} - y_j^{t+1}|| > ||y_i^t - y_j^t||$) if $f_r > 0$.*

Note that the inclusion of a symmetric term $-\lambda f_r(\zeta)(y_i^t - y_j^t)$ in Eq. (10) does not alter this conclusion. Unlike attraction, repulsion does not require introducing an effective shape: for any positive learning rate $\lambda > 0$, multiplying f_r by λ changes only the magnitude of the repulsive update, not its sign. Thus, the qualitative condition for per-iteration expansion is determined by $f_r > 0$, and we treat f_r itself as the repulsion shape. Proofs of these propositions are provided in Appendix B.

Overall, these conditions give a per-iteration certificate for guaranteed contraction/expansion, and one should not confuse them with the learning rate tuning/decay mechanism, as we can design the shapes in such a way that the contraction and expansion do not rely on learning rate decay.

Figure 1 (a) shows the effect of different values of $f_a < 0$ and $f_r > 0$ on two points. The latter shape is straightforward, as any positive value increases the distance. The former is much more subtle, as f_a encodes both attractive and repulsive dynamics. For $f_a \in (-1.0, 0)$, the distance decreases, with a sign flip at the value $f_a = -0.5$ of maximum attraction (coincident points). Any value lower than -1.0 causes the distance to increase. Although these forces act locally, they collectively shape the global structure.

5 Analysis of UMAP in terms of Attraction and Repulsion Shape

Using the gradient decomposition and the distance form (1), the attraction and repulsion shapes are given by

$$f_a^U(\zeta) = -\frac{2ab\zeta^{2(b-1)}}{1 + a\zeta^{2b}} \tag{11}$$

and

$$f_r^U(\zeta) = \frac{2b}{\zeta^{2b}(1 + a\zeta^{2b})}, \tag{12}$$

respectively. This section focuses on the default shapes of UMAP, controlled primarily by the parameters a and b, and discusses the insights learned by perturbing them. The discussion below is generally valid for $b \leq 1$, where the attraction shape is strictly increasing and thus invertible (for derivation, see Appendix B; for a discussion of $b > 1$, see Appendix F).

Table 1: Effect of ζ_{-1} in terms of mean±std Euclidean distances among the k-NN pairs in the UMAP embedding space. With annealing, $\zeta_{-1} \to 0$; without annealing (constant learning rate), $\zeta_{-1} = 1.07$, leading to larger k-NN distances. Results for embedding dimensions scaled to unit variance are shown in parentheses.

Dataset	With Annealing	Without Annealing
MNIST (LeCun et al., 2010)	0.56 ± 1.70 (0.10 ± 0.31)	1.71 ± 1.80 (0.29 ± 0.30)
FMNIST (Xiao et al., 2017)	0.39 ± 0.94 (0.06 ± 0.16)	1.52 ± 1.24 (0.23 ± 0.18)
Transcriptomes (Macosko et al., 2015)	0.48 ± 1.16 (0.09 ± 0.22)	1.67 ± 1.54 (0.26 ± 0.23)

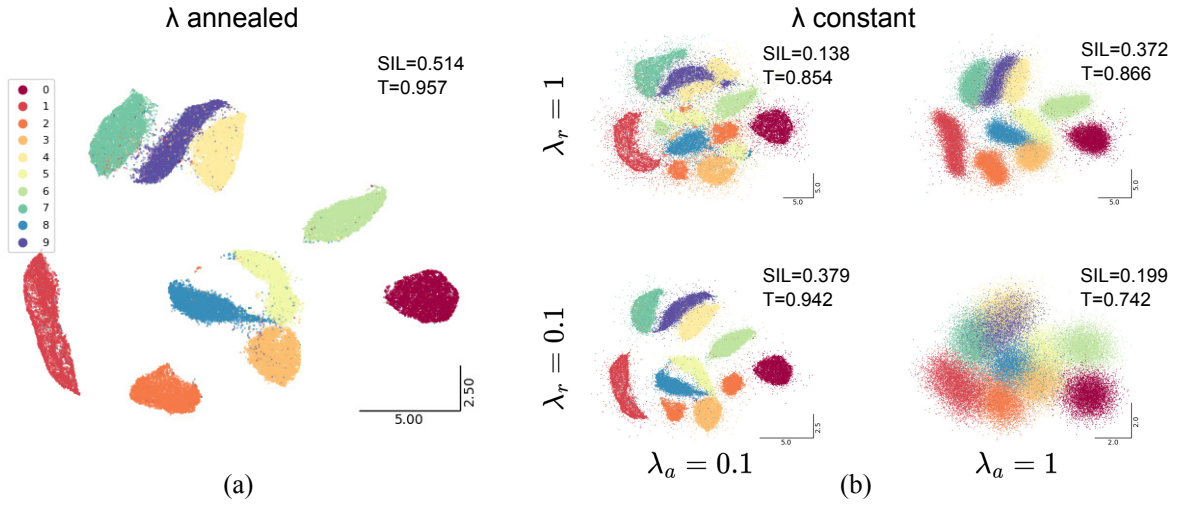


Figure 2: Embedding of the MNIST dataset using UMAP's default attraction and repulsion. (a) Embedding when the learning rate is annealed. (b) Embeddings when the learning rate is constant. We varied the constant learning rate for attraction (λ_a) and repulsion (λ_r). A smaller learning rate gives a sharper cluster boundary. SIL: silhouette scores (Rousseeuw, 1987), a measure of cluster separation, and T: trustworthiness (Venna & Kaski, 2001), a measure of structure preservation.

Figure 1 (b) shows the default attraction shape of UMAP ($a = 1.58, b = 0.89$). It becomes unbounded (approaches $-\infty$) as $\zeta \to 0$. As predicted by Proposition 4.1, the transition from contraction to expansion occurs when λf_a^U crosses -1 as ζ approaches 0. Since the attraction shape is invertible, we can identify the distance at this transition, $\zeta_{-1} = (\lambda f_a^U)^{-1}(-1)$, as the minimum distance for contraction due to attractive updates. Effectively, $\zeta > \zeta_{-1}$ causes contraction, and $0 < \zeta < \zeta_{-1}$ causes expansion, contradicting the intuition that attractive updates consistently bring points closer together.

If ζ_{-1} is high, neighboring points oscillate between contraction and expansion, and the clusters appear fuzzy. For the sharpest boundaries, then, the goal of optimization can be recast as one of achieving the limit $\zeta_{-1} \to 0$. The default shape oscillates around $\zeta = 1.07$, which is large compared to the expectation (i.e., the limit $\zeta_{-1} \to 0$). As a result, UMAP's learning rate schedule requires annealing to zero (Figs. 1 (c,d)). The effect is evident when we quantify the distances among the k-NN pairs in the learned embeddings (Table 1). Without annealing, the average distance among the k-NN pairs increases by ≈ 1.16, consistent with the predicted scale set by ζ_{-1}. This increase persists even when the embedding dimensions are scaled to unit variance (by ≈ 0.18).

On the other hand, the repulsion shape (Eq. 12) is always positive and satisfies Proposition 4.2. f_r^U approaches 0 as $\zeta \to \infty$ and approaches ∞ as $\zeta \to 0$ (Fig. 1 (e)).

To illustrate their effect, Fig. 2 shows UMAP embeddings of MNIST under different optimization settings. With learning rate annealing, UMAP produces the familiar embedding with well-formed clusters of individual digits (Fig. 2 (a)). In contrast, keeping the learning rate constant can lead to drastically different embeddings (Fig. 2 (b)). For clarity, we separately fix the attraction and repulsion learning rates, denoted by λ_a and

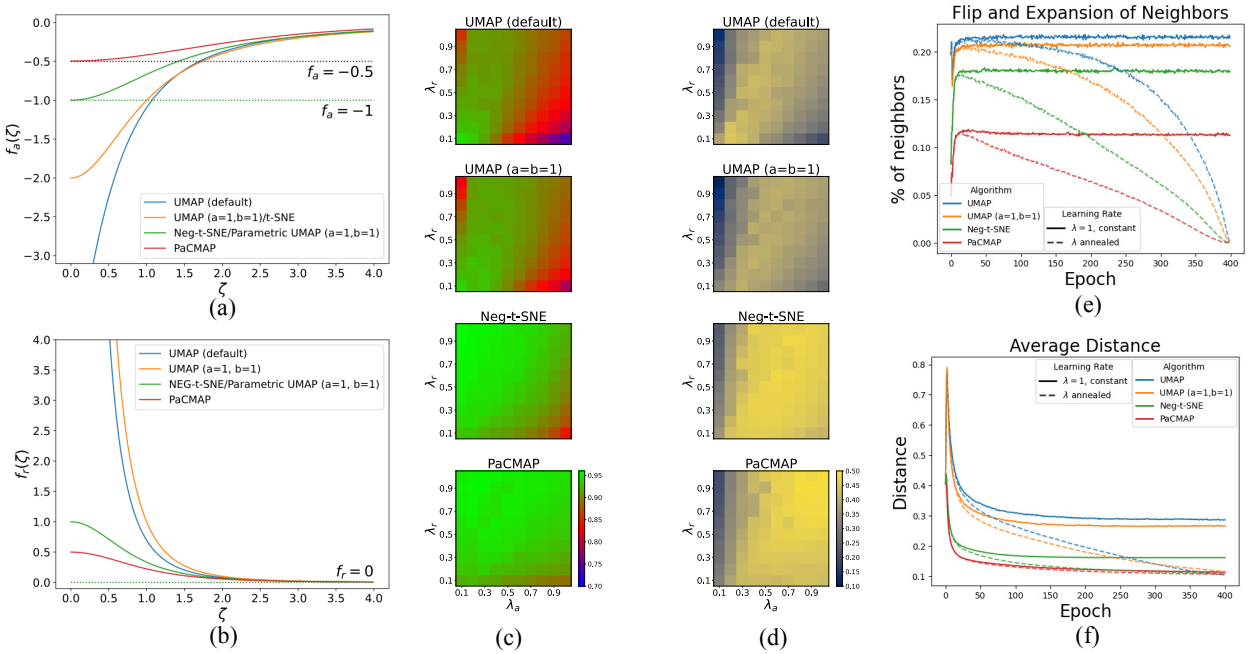


Figure 3: Comparison of different algorithms. (a) Attraction and (b) repulsion shapes of various embedding methods. (c) Trustworthiness and (d) Silhouette Score of various methods as the constant learning rate is varied for attraction (λ_a) and repulsion (λ_r) independently. (e) Portions of neighbors that flip and expand, and (f) average distance between neighbors for various methods due to attraction and repulsion dynamics in each epoch.

λ_r. Clearer cluster boundaries emerge when λ_a is small (e.g., 0.1), which also keeps ζ_{-1} small. When λ_a is large (e.g., 1.0), the clusters become more diffuse. The λ_r mainly affects local jitter; a larger λ_r (1.0) introduces noticeable fluctuations around the clusters, whereas a smaller λ_r (0.1) yields a cleaner embedding (for $\lambda_a = 0.1$). We also observe changes in the overall scale of the embedding, reflecting the balance between attraction and repulsion. As this balance shifts, the equilibrium spacing between clusters changes, causing the embedding to contract or expand (we analyze the scaling further in Section 5.3).

To connect Fig. 2 to the underlying mechanics, note that (λ_a, λ_r) do not merely "speed up" optimization, rather they change the effective forces through $\lambda_a f_a$ and $\lambda_r f_r$. In particular, whenever $\lambda_a f_a$ crosses -1 (equivalently $\zeta_{-1} > 0$), nearest-neighbor updates can oscillate between contraction and expansion around ζ_{-1}, producing fuzzy clusters unless λ_a is reduced (thus, reducing the value of ζ_{-1}). This motivates comparing the attraction/repulsion shapes across methods. Attraction shapes for different algorithms show that only UMAP and t-SNE deal with the issue of having $f_a < -1$ (and consequently $\zeta_{-1} > 0$ at $\lambda = 1$, Fig. 3(a)). t-SNE solves it by weighting the updates with corresponding p_{ij} values, while UMAP relies on learning rate annealing. Methods like Neg-t-SNE and PaCMAP have f_a naturally within $[-1, 0]$ (and thus, satisfies Proposition 4.1 for $\lambda \in [0, 1]$ with $\zeta_{-1} = 0$). Furthermore, PaCMAP's weighted f_a (< 0.5 always) even prevents any flips during attraction. On the other hand, the repulsion shapes for different algorithms show that only UMAP deals with large values for small distances (Fig. 3(b); the shape is unbounded, but during optimization, the values are often clipped).

We show the effect of these attraction/repulsion choices for MNIST embedding by applying separate constant learning for attraction (λ_a) and repulsion (λ_r), and varying their values within $(0, 1]$ (Figs. 3 (c,d)). For UMAP, the lower value of λ_a (< 0.5 and consequently $\zeta_{-1} \simeq 0$) is preferred for better embedding (trustworthiness (T)in Fig. 3 (c)) and clustering (silhouette scores (SIL) in Fig. 3 (d)), whereas the whole range of λ_r ($\in [0, 1]$) could be used (e.g., for a fixed $\lambda_a = 0.3$). For NEG-t-SNE and PaCMAP, for which $\zeta_{-1} = 0$, the whole range of parameters is effective. This confirms that UMAP relies on making ζ_{-1} close to 0 for satisfactory

embedding and clustering, which it achieves in practice by learning rate annealing, whereas $\zeta_{-1} = 0$ is the default in newer algorithms.

To further probe the inner workings of attraction, we can quantify the number of flips and expansions of nearest neighbors. Note that these values account for all interactions during each epoch (while Propositions 4.1 and 4.2 account for local interactions in isolation). We can intuitively predict from the attraction shapes that UMAP will experience the highest number of flips and expansions, while the other methods, UMAP with $a = 1$ and $b = 1$, Neg-t-SNE, and PaCMAP, will gradually go downward. Fig. 3 (e) shows the percentage of neighbors that experience this during optimization and perfectly matches with our intuition (for details of how we estimate these values, see Appendix L). When the learning rate is constant, the number of flips is stable. In UMAP, roughly 21.55% (average of last 200 epochs) of the neighbors flip and expand. Whereas, it becomes lower as we drive the attraction shape toward -1 and then to -0.5. For PaCMAP, where there is no flip from the attraction shape, 11.34% of the neighbors (half of that of UMAP) flip and expand. When the learning rate is annealed, the number of points that flip and expand reduces toward 0.

We can make a similar observation regarding the distance between the neighbors (Fig. 3 (f); for details, see Appendix L). Without learning rate annealing, the average distance in UMAP is highest and then drops significantly for Neg-t-SNE and PaCMAP. When the learning rate is annealed, all the methods reach practically the same value. Notably, PaCMAP, which allows no flip during attractive update, essentially traces the same curve with or without learning rate annealing.

Future sections and appendices provide additional details. In Section 5.1, we focus on the case of $a = 1$ and $b = 1$ and compare UMAP to NEG-t-SNE. Appendix C provides additional discussion regarding constant learning rate in UMAP. Appendix J provides details of different dimensionality reduction algorithms (TriMAp, PaCMAP, LocalMAP, t-SNE, SNE and multidimensional scaling) from the perspective of attraction and repulsion shape. Appendix M discusses the case when flips are deliberately injected during the optimization. Appendix N studies a synthetic scenario in which an attractive pair is initialized with many repelling points between them. Appendix O gives details of the embeddings used in this section as well as extended results for two other datasets.

5.1 Comparison to NEG-t-SNE

In the previous section, we focused on the default UMAP parameters, which causes both the attraction and repulsion shapes to be unbounded. However, setting a and b to 1 is common for various dimensionality reduction algorithms. This particular setting makes the gradients in many algorithms stable (or bounded) and, in turn, makes the optimization easier. Recently, Damrich et al. (2023) explored this for UMAP and proposed Neg-t-SNE as a solution, which, in addition to having a stable gradient, provides a more compact clustering even for a constant learning rate of 1. On the other hand, Parametric UMAP (Sainburg et al., 2021) initially used the UMAP loss formulation, but later it[1] adopted a numerically stable modified cross-entropy loss function (Shi et al., 2023) with a logsigmoid kernel. Analysis of both reveals that the approaches arrive at the same formulation from different starting points, which leads to the following observation:

Proposition 5.1 (Damrich et al. (2023))**.** *Neg-t-SNE is Parametric UMAP with $a = 1$ and $b = 1$.*

With $a = 1$ and $b = 1$, the attraction and repulsion shapes of UMAP are given by $f_a^U = -2/(1 + \zeta^2)$ and $f_r^U = 2/(\zeta^2(1 + \zeta^2))$, respectively. The attraction shape becomes bounded within $[-2, 0]$, with $f_a^U(0) = -2$ (Fig. 3 (a)), while the repulsion shape remains essentially unchanged (i.e., unbounded as $\zeta \to 0$, Fig. 3 (b)). Since $f_a^U < -1$ as $\zeta \to 0$, according to Proposition 4.1 and the discussion provided in Section 5, this unity case still requires learning rate annealing.

Damrich et al. (2023) compared UMAP's negative sampling loss function from the perspective of contrastive embedding (CE) and concluded that the effective kernel of UMAP is $1/\zeta^2$. Under the CE framework, the authors introduced NEG-t-SNE by changing the kernel to $1/(1 + \zeta^2)$. Reverting to UMAP formalism, this results in the low-dimensional affinity function $q_{ij}^N = 1/(2 + \zeta^2)$. Consequently, the attraction and repulsion shapes are $f_a^N = -2/(2 + \zeta^2)$, and $f_r^N = 2/((1 + \zeta^2)(2 + \zeta^2))$, respectively (Figs. 3 (g,h)). The attraction shape is bounded within $[-1, 0]$ and satisfies Proposition 4.1. Any $\lambda \in [0, 1]$ would cause contraction and

[1] https://github.com/lmcinnes/umap/pull/856, merged on April 26, 2022

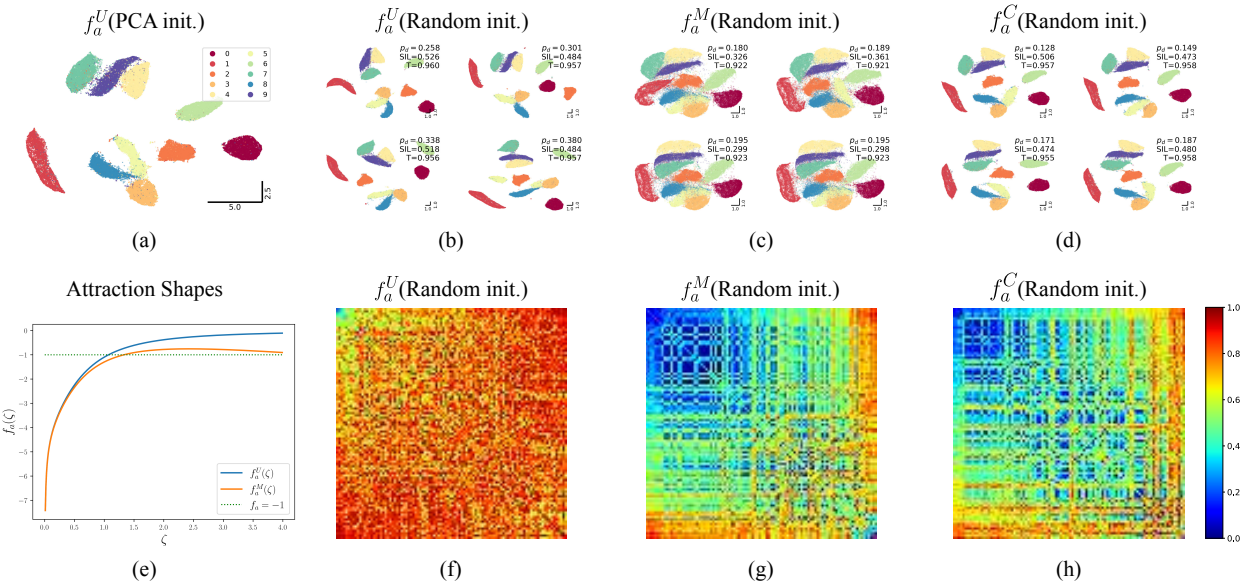


Figure 4: Effect of random initialization on different attraction shapes for the MNIST dataset. (a) Mapping using PCA. (b-d) Four mappings with the lowest Procrustes distance (p_d) from the embedding in (a) for (b) UMAP, (c) modified, and (d) composite attraction shapes. (e) Default UMAP and modified attraction shapes. (f-h) Procrustes matrices from 100 runs of (f) UMAP (0.78 ± 0.13), (g) modified (0.49 ± 0.21), and (h) composite attraction (0.50 ± 0.20) shapes. The diagonal (i, i) entries of the Procrustes matrix are sorted by Procrustes distance (p_d) from (a), and the off-diagonal, (i, j), entries correspond to p_d between i^{th} and j^{th} mapping. The matrices and (mean p_d±std) values show that UMAP's embeddings are not self-similar, while the modified and composite attraction shapes encourage initialization-invariant structure.

avoid oscillation of expansion and contraction. Thus, NEG-t-SNE is less sensitive to learning rate annealing, and the clusters appear less fuzzy even for constant $\lambda = 1.0$. Moreover, the repulsion shape of NEG-t-SNE is also bounded within $[0, 1]$ and does not approach infinity as $\zeta \to 0$, which causes fewer points to leave the clusters when sampled randomly. While Damrich et al. (2023) used only the bounded repulsive forces of NEG-t-SNE and unbounded ones of UMAP to explain this disparity, our analysis, in Fig. 3, shows that the attraction shape is equally responsible for stability and compactness of the clusters. When a and b vary in NEG-t-SNE, it faces the same numerical challenges of UMAP.

For additional discussion on NEG-t-SNE with illustration, see Appendix I.

5.2 UMAP's Consistency under Random Initialization and Probing Its Optimization Landscape

The consistency of UMAP embeddings depends on proper initialization (Kobak & Linderman, 2021; Wang et al., 2021). Typically, principal component analysis (PCA) of the data or spectral decomposition of the high-dimensional graph initializes the embedding, producing consistent mappings despite various sources of stochasticity. If randomly initialized, clusters often fail to form or form in a random orientation each time the algorithm executes. If the initial distance between two points (nearest neighbors in high dimension) is large, the attractive forces become too low to bring them closer. Known as near-sightedness (Wang et al., 2021), this phenomenon is evident in the attraction shape, where $|f_a^U|$ diminishes towards zero as the distance increases (since, $|f_a^U| = o(1/\zeta)$, and thus, $\lim_{\zeta \to \infty} |f_a^U(\zeta)|\zeta = 0$). If we can alleviate this near-sightedness, we can probe whether an informed initialization (such as PCA) leads to a preferred low-dimensional arrangement, or whether multiple such arrangements are equally accessible under the objective.

One can induce "far-sightedness" in the mapping by increasing attraction for large distances, facilitating faraway neighbors to come closer. To test this hypothesis, we modify the attraction shape of UMAP to

Table 2: Effect of random initialization on different datasets using different attraction shapes and comparison to PCA initialization. The metrics (mean±std) are reported based on 100 runs of each. Datasets: MNIST (LeCun et al., 2010), FMNIST (Xiao et al., 2017), Transcriptomes- (Macosko et al., 2015), (Shekhar et al., 2016), and 20NewsGroup (20NG) (Mitchell, 1997).

		Embedding Quality			Run to Run Consistency	
Dataset	Shape (initialization)	Trustworthiness	Silhouette Score	Spearman	Spearman	Procrustes Distance
MNIST	*Default (PCA, baseline)*	*0.957 ± 0.001*	*0.51 ± 0.01*	*0.31 ± 0.01*	*0.95 ± 0.06*	*0.13 ± 0.05*
	Default (Rand)	**0.958 ± 0.001**	0.47 ± 0.04	0.24 ± 0.06	0.44 ± 0.12	0.78 ± 0.13
	Modified (Rand)	0.923 ± 0.003	0.33 ± 0.03	**0.33 ± 0.03**	**0.71 ± 0.12**	**0.49 ± 0.21**
	Composite (Rand)	0.956 ± 0.001	**0.48 ± 0.02**	0.29 ± 0.04	0.70 ± 0.11	0.50 ± 0.21
FMNIST	*Default (PCA, baseline)*	*0.975 ± 0.001*	*0.18 ± 0.00*	*0.60 ± 0.00*	*0.99 ± 0.00*	*0.02 ± 0.00*
	Default (Rand)	**0.976 ± 0.001**	0.11 ± 0.05	0.42 ± 0.07	0.54 ± 0.13	0.72 ± 0.15
	Modified (Rand)	0.959 ± 0.004	0.16 ± 0.03	**0.57 ± 0.04**	**0.82 ± 0.11**	**0.38 ± 0.19**
	Composite (Rand)	0.975 ± 0.001	**0.18 ± 0.03**	0.53 ± 0.05	0.78 ± 0.10	0.43 ± 0.18
Macosko	*Default (PCA, baseline)*	*0.950 ± 0.001*	*0.42 ± 0.04*	*0.77 ± 0.01*	*0.95 ± 0.02*	*0.16 ± 0.07*
	Default (Rand)	**0.950 ± 0.001**	0.23 ± 0.06	**0.75 ± 0.02**	0.76 ± 0.03	0.91 ± 0.06
	Modified (Rand)	0.935 ± 0.001	0.15 ± 0.05	0.71 ± 0.02	0.82 ± 0.05	**0.61 ± 0.13**
	Composite (Rand)	0.949 ± 0.001	**0.28 ± 0.05**	0.73 ± 0.02	**0.83 ± 0.04**	0.64 ± 0.15
Shekhar	*Default (PCA, baseline)*	*0.974 ± 0.000*	*0.52 ± 0.01*	*0.51 ± 0.01*	*0.94 ± 0.03*	*0.07 ± 0.03*
	Default (Rand)	**0.974 ± 0.000**	0.41 ± 0.08	**0.48 ± 0.03**	0.52 ± 0.09	0.70 ± 0.16
	Modified (Rand)	0.965 ± 0.001	**0.58 ± 0.03**	0.40 ± 0.02	**0.81 ± 0.06**	**0.48 ± 0.19**
	Composite (Rand)	0.973 ± 0.000	0.51 ± 0.03	**0.48 ± 0.01**	0.73 ± 0.08	0.51 ± 0.19
20NG	*Default (PCA, baseline)*	*0.819 ± 0.001*	−0.17 ± 0.00	*0.58 ± 0.00*	*0.96 ± 0.01*	*0.05 ± 0.01*
	Default (Rand)	**0.818 ± 0.002**	−0.18 ± 0.01	0.57 ± 0.01	0.92 ± 0.03	0.24 ± 0.13
	Modified (Rand)	0.777 ± 0.001	**−0.16 ± 0.00**	**0.59 ± 0.00**	**0.95 ± 0.04**	0.20 ± 0.13
	Composite (Rand)	0.817 ± 0.002	−0.18 ± 0.01	0.58 ± 0.01	0.94 ± 0.04	**0.19 ± 0.15**

increase the attractive force:

$$f_a^M = f_a^U - \beta\zeta, \tag{13}$$

where β is a parameter that regulates the strength of the added term (we used $\beta = 0.2$, Fig. 4 (e)). This addition in the attraction shape translates to adding a regularizer in the attractive term of the loss function (i.e., $\mathcal{L}^M = \mathcal{L} + \sum_{i,j} \beta/3 \; ||y_i - y_j||_2^3$). In addition to attracting pairs at faraway distances, this technique enables intermixing of points that help convergence under random initialization (akin to early exaggeration in t-SNE). In (13), we chose the simplest linear correction; other functions, such as $\log \zeta$ or ζ^p ($p \in \mathbb{R}_{\geq 0}$), may also be suitable.

We also consider a composite attraction shape:

$$f_a^C = \begin{cases} f_a^M, & \text{epoch} \leq 100 \\ f_a^U, & \text{otherwise} \end{cases}. \tag{14}$$

The composite shape attempts to remove any distortions introduced by f_a^M by reverting to the original UMAP. Below, we discuss the effects these modified and composite attraction shapes have on DR from random initialization.

We first created a PCA-initialized embedding of the MNIST dataset Figure 4 (a). Then, we produced embeddings using random initialization (Gaussian) for each shape and repeated the experiment 100 times (to probe the optimization landscape starting from various initial position). To quantify the results, we use Procrustes analysis (Gower, 1975) that aligns two point clouds under scaling, translation, rotation, and reflection (for details, see Appendix D.1). Here, we align the randomly initialized embeddings to that of the PCA-initialized one and characterize their separation using the Procrustes distance (p_d). Figure 4 (b) shows four embeddings with the lowest p_d. While the cluster shapes are consistent, their placements are not. Outputs from the modified and composite attraction shapes (Figs. 4 (c) and (d), respectively) show improved consistency of cluster placements.

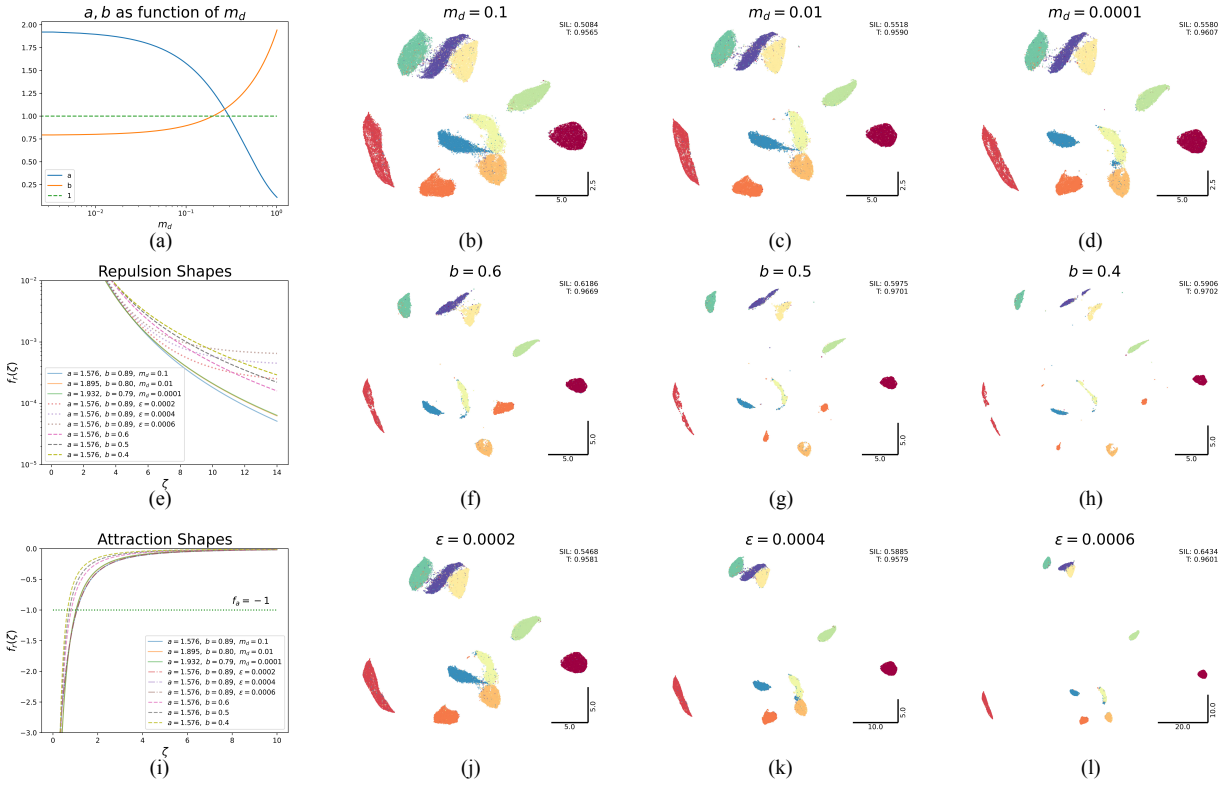


Figure 5: Control of inter-cluster distances on the MNIST dataset. (a) Computing a, b by varying the low-dimensional distance m_d restricts exploration. (b-d) UMAP output by setting m_d to 0.1, 0.01, and 0.0001, respectively, shows little improvement in compactness of clusters. (e) Repulsion shapes for different parameters. (f-h) Increasing repulsion by explicitly varying b results in more compact clusters and forms new ones that were absent otherwise. (i) Attraction shapes by varying parameters. (j-l) Increasing repulsion by adding a small positive value (ε) to the repulsion shape increases inter-cluster distance.

To quantify the placements further, we consider the Procrustes matrix: the diagonal of the matrix is sorted by p_d from the PCA-initialized mapping, and the off-diagonal values are p_d between two randomly initialized mappings (for details, see Appendix D.1). This quantification is analogous to the similarity matrix (Foote, 1999). The embeddings due to the default UMAP attraction shape are not similar to each other (Fig. 4 (f)), but the modified (Fig. 4 (g)) and composite (Fig. 4 (h)) shapes show strong similarity to each other. Additional results on other datasets, showing consistent behavior, are provided in Appendix E.

Table 2 shows embedding quality and run-to-run variance of five different datasets when randomly initialized and compare it to the corresponding PCA initialized version (standard and the most consistent one). Overall, we can observe that the embedding quality is comparable among different shapes. Overall, modified and composite shapes improve consistency in terms of Spearman's rank correlation and Procrustes distance.

This indicates that UMAP, regardless of the initialization, aims to encode a unique structure (in our experiments, PCA initialized embedding is an attractor). However, attaining that structure in the low dimension may fall short due to small attraction at longer distances.

5.3 Cluster Formation and Compactness

The primary controllable parameter influencing cluster formation in UMAP is the minimum distance parameter m_d (through Eq. 2). However, varying m_d restricts the exploration of different values of a and b (Fig. 5 (a)). Thus, reducing m_d often results in embeddings that do not provide additional benefit (Figs. 5 (b-d)). The key factor is the limited influence of varying m_d on the repulsion shape (Fig. 5 (e)). Alternatively, we can

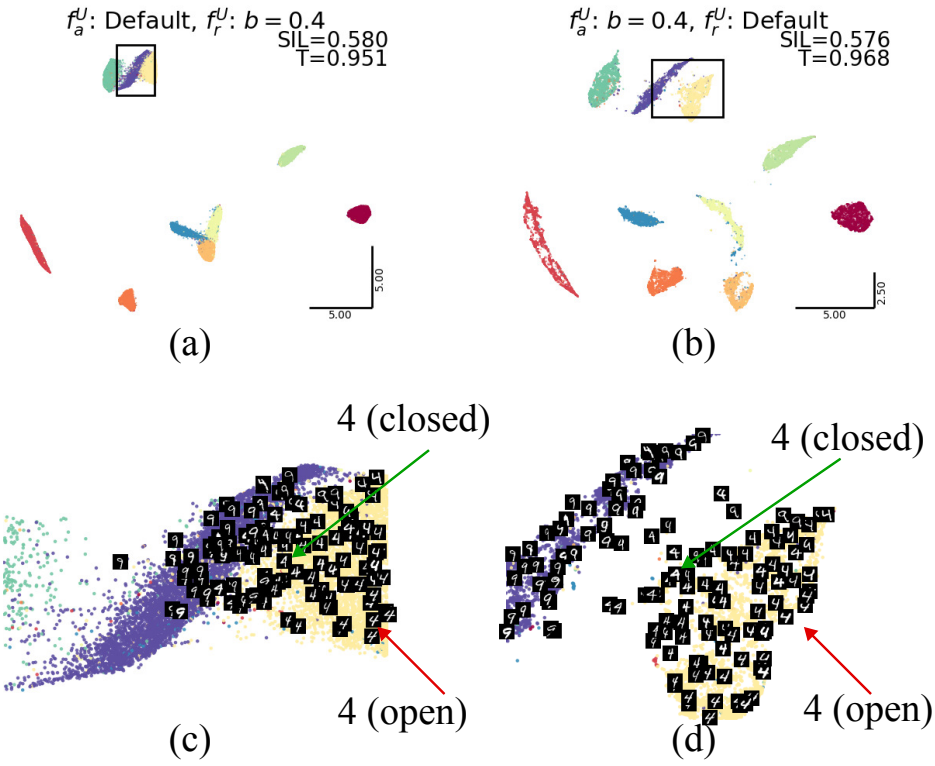


Figure 6: Embedding of the MNIST dataset with (a) default attraction shape but repulsion shape with $b = 0.4$ and (b) default repulsion shape but attraction shape with $b = 0.4$. The former shows the same clusters of default UMAP with increased compactness. The latter develops new structures within clusters and forms new clusters. (c) Clusters of 4 and 9 from (a). (d) The same clusters from (b). Both images show the same samples of 4s and 9s.

explicitly vary the values of a and b. Decreasing a increases repulsion, but it decreases attraction at a faster rate (causing a worse case of near-sightedness; we provide a discussion in Appendix G). On the other hand, decreasing b gives a better control (Figs. 5 (e,i)). Figures 5 (f-h) show increasing inter-cluster distance and breaking up of previous clusters by varying b to 0.6, 0.5, and 0.4, respectively. This breaking up occurs due to the increasingly heavy-tailed nature of the kernel as b decreases (heavy-tailed kernels result in smaller and distinct clusters (Van der Maaten & Hinton, 2008; Yang et al., 2009; Kobak et al., 2019). See Kobak et al. (2019) for how varying b in UMAP modulates the tail and Lu & Calder (2025) for varying b-like parameter in t-SNE).

Although this approach separates all ten MNIST labels into distinct clusters, the relative contributions of attraction and repulsion are hard to disentangle; changing b amplifies repulsion more than changing m_d, but it also reshapes the attraction profile.

To quantify the effect of repulsion independently, we can keep attraction fixed and vary the other. To achieve this, we modify the repulsion shape by adding a small positive value (ε):

$$f_r^M = f_r^U + \varepsilon, \tag{15}$$

while keeping the values of a and b constant, which effectively adds a regularizer in the repulsive term of the loss function (i.e, $\mathcal{L}^M = \mathcal{L} - \sum_{i,j} \varepsilon/2 \; ||y_i - y_j||_2^2$). Using this, we can obtain stronger repulsion than previously (Fig. 5(e)). Figs. 5 (j-l) show that as ε increases, the inter-cluster distances also increase. However, the clustering properties show similarity to those obtained by varying m_d, and we get a loose separation of all the labels as ϵ increases. Overall, the parameter ε keeps the attraction shape unaffected, and varying m_d

effectively traces similar attraction shapes (Fig. 5 (i)), suggesting cluster formation is governed predominantly by attraction.

To explore further, we change either the attraction shape or the repulsion shape individually while leaving the other at the default by simply setting b to 0.4 (Fig. 6)[2]. The default attraction is unable to show new structures or clusters in the embedding, but the increased repulsion ($b = 0.4$) gives smaller clusters than the original UMAP (Fig. 6 (a)). On the other hand, when the attraction increases by setting $b = 0.4$ with the default repulsion, the embedding shows additional structures within each cluster (Fig. 6 (b)). Some of the older clusters even separate into smaller ones.

For illustration, we zoom into the region where labels 4 and 9 meet (Figs. 6(c,d)). With default attraction and stronger repulsion ($b = 0.4$), the embedding mainly tightens existing groups—the 4–9 boundary remains relatively diffuse, and repulsion alone does not reveal finer substructure within the 4s (Fig. 6(c)). In contrast, when we strengthen attraction ($b = 0.4$) while keeping repulsion at its default value, the two labels separate more cleanly due to newer structure formation, and a clear internal organization emerges (Fig. 6(d)). In particular, closed-top 4s lie between open-top 4s and 9s, which is consistent with their visual similarity to both classes. We examine a few more labels in Appendix H. While this has been enlightening, we do not claim that every additional cluster induced by heavier-tailed attraction is necessarily desirable; in an unsupervised embedding, such structure may reflect either meaningful heterogeneity or over-fragmentation.

This shows that attraction causes cluster formation, while repulsion makes the clusters more compact (we interpret compactness as follows: if the embeddings are rescaled to same scaling, repulsion shape does this by making smaller clusters, and otherwise it increases inter-cluster distance). By controlling attraction and repulsion behavior explicitly, we can create embeddings with various levels of granularity.

As an aside, one of the claims of LocalMAP is to separate the ten labels of MNIST into individual clusters. Here, we have achieved the same result in UMAP by simply manipulating the attraction and the repulsion shapes. In Appendix J.3, we explore how LocalMAP embeddings exploit this interplay of attraction and repulsion to separate the clusters and relate it to the experiments of this section.

6 Discussion and Conclusion

In this work, we showed that attraction and repulsion play distinct roles in pairwise dimensionality reduction: attraction primarily governs whether clusters form, stabilize, and converge consistently, whereas repulsion mainly regulates compactness and inter-cluster spacing. By decomposing the update rules into attraction and repulsion shapes, we obtained a simple mechanistic language for comparing UMAP and related methods.

Our main finding is that UMAP's default attraction is locally too strong at short distances: when $\lambda f_a < -1$, attractive updates become expansive rather than contractive. This explains why, under fixed learning rate, UMAP can produce fuzzy clusters, and why learning rate annealing is not merely a heuristic but a mechanism for driving the system toward the contractive regime. More generally, attraction shapes that remain within a contractive range lead to more stable optimization and easily reach cleaner cluster boundaries.

We further showed that near-sighted attraction contributes to UMAP's sensitivity to random initialization. By increasing attraction at larger distances, we obtained embeddings that are substantially more consistent across random starts, suggesting that the objective favors a common low-dimensional structure but may fail to reach it when long-range attraction is too weak. In contrast, modifying repulsion primarily changes compactness and inter-cluster distance; it does not by itself induce the new internal structures that arise from changing attraction.

These insights into attraction–repulsion dynamics offer new tools for optimizing dimensionality reduction algorithms. Overall, the practical message is simple: if an embedding method produces fuzzy clusters, unstable outputs, or poor random-initialization behavior, the first place to look is the attraction shape. Repulsion matters, but mainly for spacing. Beyond this, the close connection between dimensionality reduction and

[2]This manipulation is analogous to separating the roles of *amplitude* and *phase* in the Fourier transform of an image: changing one while holding the other fixed can produce qualitatively different percepts, even though both components contribute to the final reconstruction. See Fig. 3 of Oppenheim & Lim (1981)

contrastive learning (Damrich et al., 2023; Hu et al., 2023) suggests that our approach can also enhance representation learning. Taken together, our work aims to make embeddings and their interpretations more principled, consistent, and reliable, and guide future research.

Acknowledgment

This work was supported by AFOSR grant FA9550-21-1-0317 and the Schmidt DataX Fund at Princeton University, made possible through a major gift from the Schmidt Futures Foundation. Mohammad Tariqul Islam is supported by MIT-Novo Nordisk Artificial Intelligence Fellowship.

Software and Data

All the data used in this paper are publicly available.
The MNIST dataset is available at https://yann.lecun.com/exdb/mnist/.
Fashion-MNIST is available at https://github.com/zalandoresearch/fashion-mnist.
Single-cell transcriptomes data is available at https://github.com/biolab/tsne-embedding.
Shekhar's Transcriptomes are obtained from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE81904.
20NG dataset is obtained from https://scikit-learn.org/20NG.
Additional details are provided in the Implementation Details section in Appendix P.

References

Akshay Agrawal, Alnur Ali, Stephen Boyd, et al. Minimum-distortion embedding. *Foundations and Trends® in Machine Learning*, 14(3):211–378, 2021.

Mahsun Altin and Altan Cakir. Exploring the influence of dimensionality reduction on anomaly detection performance in multivariate time series. *IEEE Access*, 2024.

Ehsan Amid and Manfred K Warmuth. TriMap: Large-scale dimensionality reduction using triplets. *arXiv preprint arXiv:1910.00204*, 2019.

Sanjeev Arora, Wei Hu, and Pravesh K Kothari. An analysis of the t-sne algorithm for data visualization. In *Conference On Learning Theory*, pp. 1455–1462. PMLR, 2018.

Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37(1):38–44, 2019.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pp. 585–591, 2002.

Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Attraction-repulsion spectrum in neighbor embeddings. *The Journal of Machine Learning Research*, 23(1):4118–4149, 2022.

Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2007.

T Tony Cai and Rong Ma. Theoretical foundations of t-sne for visualizing high-dimensional clustered data. *Journal of Machine Learning Research*, 23(301):1–54, 2022.

Tara Chari and Lior Pachter. The specious art of single-cell genomics. *PLOS Computational Biology*, 19(8): e1011288, 2023.

Sebastian Damrich and Fred A Hamprecht. On UMAP's true loss function. *Advances in Neural Information Processing Systems*, 34:5798–5809, 2021.

Sebastian Damrich, Niklas Böhm, Fred A Hamprecht, and Dmitry Kobak. From t-SNE to UMAP with contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Cyril de Bodt, Alex Diaz-Papkovich, Michael Bleher, Kerstin Bunte, Corinna Coupette, Sebastian Damrich, Enrique Fita Sanmartin, Fred A Hamprecht, Emőke-Ágnes Horvát, Dhruv Kohli, et al. Low-dimensional embeddings of high-dimensional data. *arXiv preprint arXiv:2508.15929*, 2025.

Andrew Draganov and Simon Dohn. Relating tsne and umap to classical dimensionality reduction. *arXiv e-prints*, pp. arXiv–2306, 2023.

Andrew Draganov, Jakob Jørgensen, Katrine Scheel, Davide Mottin, Ira Assent, Tyrus Berry, and Cigdem Aslay. ActUp: Analyzing and consolidating tSNE and UMAP. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pp. 3651–3658, 8 2023.

Shu-Kai S Fan, Du-Ming Tsai, Chih-Hung Jen, Chia-Yu Hsu, Fei He, and Li-Ting Juan. Data visualization of anomaly detection in semiconductor processing tools. *IEEE Transactions on Semiconductor Manufacturing*, 35(2):186–197, 2021.

Jason Fleischer and Mohammad Tariqul Islam. Late breaking abstract-identifying and phenotyping COVID-19 patients using machine learning on chest x-rays. *European Respiratory Journal*, 2020.

Jonathan Foote. Visualizing music and audio using self-similarity. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pp. 77–80, 1999.

Rita González-Márquez, Luca Schmidt, Benjamin M Schmidt, Philipp Berens, and Dmitry Kobak. The landscape of biomedical research. *Patterns*, 2024.

John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975.

Geoffrey Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, volume 15, pp. 833–840, 2002.

Tianyang Hu, Zhili Liu, Fengwei Zhou, Wenjia Wang, and Weiran Huang. Your contrastive learning is secretly doing stochastic neighbor embedding. In *The Eleventh International Conference on Learning Representations*, 2023.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 20617–20642, 2024.

Mohammad Tariqul Islam and Jason W Fleischer. Manifold-aligned neighbor embedding. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022.

Mohammad Tariqul Islam and Jason W Fleischer. Outlier detection in large radiological datasets using umap. In *International Workshop on Topology-and Graph-Informed Imaging Informatics*, pp. 111–121. Springer, 2024.

Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4): 295–307, 1988.

Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PloS one*, 9(6): e98679, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Dmitry Kobak and George C Linderman. Initialization is critical for preserving global data structure in both t-sne and umap. *Nature biotechnology*, 39(2):156–157, 2021.

Dmitry Kobak, George Linderman, Stefan Steinerberger, Yuval Kluger, and Philipp Berens. Heavy-tailed kernels reveal a finer cluster structure in t-sne visualisations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 124–139. Springer, 2019.

Alexander Kolpakov and Aidan Rocke. The information geometry of umap. *Le Matematiche*, 79(1):151–164, 2024.

Nikita Kotlov, Kirill Shaposhnikov, Cagdas Tazearslan, Madison Chasse, Artur Baisangurov, Svetlana Podsvirova, Dawn Fernandez, Mary Abdou, Leznath Kaneunyenye, Kelley Morgan, et al. Procrustes is a machine-learning approach that removes cross-platform batch effects from clinical rna sequencing data. *Communications Biology*, 7(1):392, 2024.

Joseph B Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.

Noël Kury, Dmitry Kobak, and Sebastian Damrich. DREAMS: Preserving both local and global structure in dimensionality reduction. *Transactions on Machine Learning Research*, 2026.

Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6, 2015.

Pierre Lambert, Cyril De Bodt, Michel Verleysen, and John A Lee. Squadmds: a lean stochastic quartet mds improving global structure preservation in neighbor embedding like t-sne and umap. *Neurocomputing*, 503: 17–27, 2022.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

George C Linderman and Stefan Steinerberger. Clustering with t-SNE, provably. *SIAM Journal on Mathematics of Data Science*, 1(2):313–332, 2019.

George C Linderman and Stefan Steinerberger. Dimensionality reduction via dynamical systems: The case of t-sne. *SIAM Review*, 64(1):153–178, 2022.

Jingcheng Lu and Jeff Calder. Attraction–repulsion swarming: a generalized framework of t-sne via force normalization and tunable interactions. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 383(2298), 2025.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Tom Mitchell. Twenty Newsgroups. UCI Machine Learning Repository, 1997. DOI: https://doi.org/10.24432/C5C323.

Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology*, 39(6):765–774, 2021.

Alan V Oppenheim and Jae S Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5): 529–541, 1981.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Pavlin G. Poličar, Martin Stražar, and Blaž Zupan. opentsne: A modular python library for t-sne dimensionality reduction and embedding. *Journal of Statistical Software*, 109(3):1–30, 2024.

Aditya Ravuri and Neil D Lawrence. Towards one model for classical dimensionality reduction: A probabilistic perspective on umap and t-sne. *arXiv preprint arXiv:2405.17412*, 2024.

Aditya Ravuri, Francisco Vargas, Vidhi Lalchand, and Neil D Lawrence. Dimensionality reduction as probabilistic inference. In *ICML 2023 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2023.

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

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

Tim Sainburg, Leland McInnes, and Timothy Q Gentner. Parametric umap embeddings for representation and semisupervised learning. *Neural Computation*, 33(11):2881–2907, 2021.

Uri Shaham and Stefan Steinerberger. Stochastic neighbor embedding separates well-separated clusters. *arXiv preprint arXiv:1702.02670*, 2017.

Karthik Shekhar, Sylvain W Lapan, Irene E Whitney, Nicholas M Tran, Evan Z Macosko, Monika Kowalczyk, Xian Adiconis, Joshua Z Levin, James Nemesh, Melissa Goldman, et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, 166(5):1308–1323, 2016.

Xintong Shi, Wenzhi Cao, and Sebastian Raschka. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26(3):941–955, 2023.

Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th international conference on world wide web*, pp. 287–297, 2016.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Jarkko Venna and Samuel Kaski. Neighborhood preservation in nonlinear projection methods: An experimental study. In *International Conference on Artificial Neural Networks*, pp. 485–491. Springer, 2001.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: an empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization. *The Journal of Machine Learning Research*, 22(1):9129–9201, 2021.

Yingfan Wang, Yiyang Sun, Haiyang Huang, and Cynthia Rudin. Dimension reduction with locally adjusted graphs. In *Thirty-Ninth AAAI Conference on Artificial Intelligence*, 2025.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. 2017.

Yang Yang, Hongjian Sun, Jialei Gong, Yali Du, and Di Yu. Interpretable dimensionality reduction by feature preserving manifold approximation and projection. *arXiv preprint arXiv:2211.09321*, 2022.

Zhirong Yang, Irwin King, Zenglin Xu, and Erkki Oja. Heavy-tailed symmetric stochastic neighbor embedding. *Advances in neural information processing systems*, 22, 2009.

Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. *Advances in neural information processing systems*, 17, 2004.

Jonathan X Zheng, Samraat Pawar, and Dan FM Goodman. Graph drawing by stochastic gradient descent. *IEEE transactions on visualization and computer graphics*, 25(9):2738–2748, 2018.

Appendix

Appendix A gives a brief description of the datasets used in this paper. In Appendix B, we provide necessary derivations regarding attraction and repulsion shapes of UMAP. In Appendix C we explore using a constant learning rate following our discussion in the main text. We formally define Procrustes distance and matrix and discuss the metrics in Appendix D. We explore additional datasets for the random initialization experiment in Appendix E. We discuss the UAMP embeddings for $b > 1$ in Appendix F and for varying a in Appendix G. We extend the discussion on Fig. 6 in Appendix H. We provide extended discussion of NEG-t-SNE and UMAP in Appendix I. After that, in Appendix J, we explore alternate dimensionality reduction methods. Then, for completion, we discuss the construction of the high-dimensional graph in these methods in Appendix K. In Appendix L we discuss the methods used to quantify flips and expansions, and mean distance of the neighbors. In Appendix M, we inject flips within UMAP optimization and explore its effects on the embeddings. Appendix N discusses a synthetic Separated-Neighbor dataset, a suspected pathological case where attraction and repulsion may break down. In Appendix O, we show the detailed result of varying λ_a and λ_r (from Fig. 1) for UMAP, NEG-t-SNE, and PaCMAP on different datasets. Finally, we provide implementation details in Appendix P.

A Datasets

Before analyzing the algorithms and visualizations, we briefly describe the datasets used throughout the paper. Our primary benchmark is MNIST (LeCun et al., 2010), which contains 60,000 training (and 10,000 test) 28×28 grayscale images of handwritten digits (0–9). When embedded into two dimensions, MNIST typically exhibits ten largely separated digit clusters with minor overlap, making it a canonical visualization dataset for evaluating dimensionality reduction methods (Damrich et al., 2023; Wang et al., 2021). We extend most experiments to two additional datasets: Fashion-MNIST (FMNIST) (Xiao et al., 2017), a drop-in replacement for MNIST with the same size and image format but 10 classes of clothing items—several of which are substantially more entangled—and a single-cell transcriptomic dataset from Macosko et al. (2015) ("Transcriptomes"), consisting of 44,808 mouse retinal cells annotated into 12 broad classes. For analyses involving random initialization, we additionally use a second retinal single-cell dataset from Shekhar et al. (2016) (27,499 cells; 19 classes) and the 20 Newsgroups corpus (20NG) (Mitchell, 1997), which contains 18,846 Usenet posts across 20 discussion groups. MNIST and FMNIST are used in their standard form, while the transcriptomic datasets and 20NG are first reduced using PCA to 50 and 100 dimensions, respectively, before applying the embedding methods.

B Proofs and Derivations

B.1 Proof of Proposition 4.1

Proof. We subtract Eq. (7) from Eq. (8) and take a norm:

$$||y_i^{t+1} - y_j^{t+1}|| = |1 + 2\lambda f_a|||y_i^t - y_j^t||. \tag{16}$$

This distance contracts as long as $|1 + 2\lambda f_a| < 1$, i.e., provided

$$-1 < \lambda f_a < 0, \tag{17}$$

□

B.2 Proof of Proposition 4.2

Proof. We subtract Eq. (9) from Eq. (10) and take a norm:

$$||y_i^{t+1} - y_j^{t+1}|| = |1 + \lambda f_r| ||y_i^t - y_j^t||. \tag{18}$$

This distance increases when $|1 + \lambda f_r| > 1$, i.e.,

$$\lambda f_r < -2 \quad \text{or} \quad f_r > 0. \tag{19}$$

From the definition of f_r, the latter suffices. □

B.3 Attraction Shape:

We use a general form of the low-dimensional affinity function, i.e., $q_{ij} = (\gamma + a||y_i - y_j||_1^{2b})^{-1}$, to derive the attraction shape. It reduces to UMAP for $\gamma = 1$ and to NEG-t-SNE for $\gamma = 2$. The attractive force is given by

$$\begin{aligned}
\nabla_{y_i} \log q_{ij} &= -\nabla_{y_i} \log \left(\gamma + a(||y_i - y_j||_2^2)^b\right) \\
&= -\frac{1}{\gamma + a(||y_i - y_j||_2^2)^b} \nabla_{y_i} (\gamma + a(||y_i - y_j||_2^2)^b) \\
&= -\frac{1}{\gamma + a(||y_i - y_j||_2^2)^b} ab(||y_i - y_j||_2^2)^{b-1} \nabla_{y_i} ||y_i - y_j||_2^2 \\
&= -\frac{2ab(||y_i - y_j||_2^2)^{b-1}}{\gamma + a(||y_i - y_j||_2^2)^b} (y_i - y_j).
\end{aligned} \tag{20}$$

Defining $\zeta = ||y_i - y_j||_2$, the first term gives the attraction shape as:

$$f_a(\zeta) = -\frac{2ab\zeta^{2(b-1)}}{\gamma + a\zeta^{2b}} \tag{21}$$

B.4 Condition for strictly increasing f_a^U:

Its behavior with distance can be characterized by computing the derivative of f_a^U:

$$\frac{df_a^U(\zeta)}{d\zeta} = -\frac{2ab\zeta^{2b-3}}{\gamma + a\zeta^{2b}} \left((b-1) - \frac{ab\zeta^{2b}}{\gamma + a\zeta^{2b}} \right). \tag{22}$$

This leads to a strictly increasing condition ($\frac{df_a^U}{d\zeta} > 0$),

$$g(\zeta, b, a) < 0, \tag{23}$$

where $g(\zeta, b, a) = b - 1 - \frac{ab\zeta^{2b}}{\gamma + a\zeta^{2b}}$. This inequality is valid as long as $0 < b \leq 1$ (using the derivative and asymptotes of g). Figure 7 shows values of g for different b and a. As b increases above 1, the inequality (23) no longer holds.

B.5 Repulsion Shape:

The repulsive force, using the general form of the low-dimensional affinity, is given by

$$\begin{aligned}
\nabla_{y_i} \log (1 - q_{ij}) &= \nabla_{y_i} \log \left[1 - \frac{1}{\gamma + a(||y_i - y_j||_2^2)^b} \right] \\
&= \frac{\gamma + a(||y_i - y_j||_2^2)^b}{(\gamma - 1) + a(||y_i - y_j||_2^2)^b} \nabla_{y_i} \left[1 - \frac{1}{\gamma + a(||y_i - y_j||_2^2)^b} \right] \\
&= \frac{1}{\gamma - 1 + a(||y_i - y_j||_2^2)^b} \frac{ab(||y_i - y_j||_2^2)^{b-1}}{\gamma + a(||y_i - y_j||_2^2)^b} \nabla_{y_i} ||y_i - y_j||_2^2 \\
&= \frac{2ab(||y_i - y_j||_2^2)^{b-1}}{(\gamma - 1 + a(||y_i - y_j||_2^2)^b)(\gamma + a(||y_i - y_j||_2^2)^b)} (y_i - y_j).
\end{aligned} \tag{24}$$

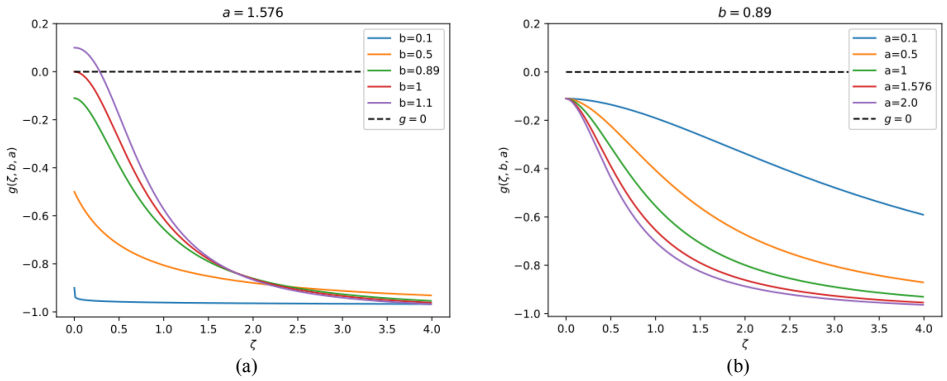


Figure 7: Values of $g(\zeta, b, a)$ for (a) a fixed at 1.576 and (b) b fixed at 0.89.

The first term gives the repulsion shape as:

$$f_r(\zeta = ||y_i - y_j||_2) = \frac{2ab\zeta^{2(b-1)}}{(\gamma - 1 + a\zeta^{2b})(\gamma + a\zeta^{2b})}. \tag{25}$$

Generally, $f_r > 0$.

B.6 Loss functions due to modified attraction and repulsion shape:

The cost function of the attractive term with the modification in Eq. (13) is given by

$$-\int (f_a^U(||y_i - y_j||_2) - \beta||y_i - y_j||_2)(y_i - y_j)dy_i = -\log(q_{ij}) + \frac{\beta}{3}||y_i - y_j||_2^3, \tag{26}$$

whereas the repulsive term due to Eq. (15) is given by

$$-\int (f_r^U(||y_i - y_j||_2) + \epsilon)(y_i - y_j)dy_i = -\log(1 - q_{ij}) - \frac{\epsilon}{2}||y_i - y_j||_2^2. \tag{27}$$

In both cases, the additional term acts as a regularizer, simply using norms. However, when we directly add this term to attraction and repulsion shapes, we can easily explain what each term is doing. For the attractive term, it is increasing far-sightedness, whereas for the repulsive term, it adds a constant repulsive coefficient.

C More on UMAP's Learning Rate

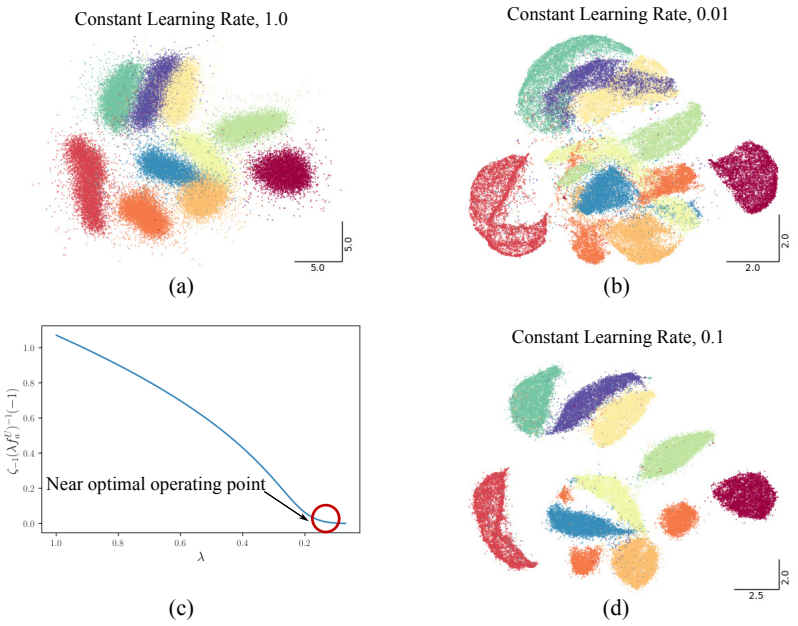


Figure 8: Effect of constant learning rate in embeddings. (a) When the learning rate is too high ($\lambda = 1.0$), the embeddings are diffuse (because of the high value of ζ_{-1}). (b) When the learning rate is too low ($\lambda = 0.01$), clusters don't form (the strengths of attraction and repulsion are too low). (c) ζ_{-1} decreases nonlinearly as the learning rate decreases. The goal of the algorithm is to reduce ζ_{-1} while keeping effective levels of attraction and repulsion. (d) Distinct and compact clusters form at a constant, near-optimal learning rate $\lambda = 0.1$.

As we discussed in Sections 5 and 5.1 of the main text, it is believed that UMAP requires learning rate annealing (Fig 8). To explain this, in Section 5, we defined the concept of minimum distance for contraction (ζ_{-1}) and established that reducing this value through learning rate annealing results in compact clusters. Later, in Section 5.1, we compared attraction shapes of UMAP (for $a = 1.0$ and $b = 1.0$) and Neg-t-SNE and explained that Neg-t-SNE can withstand a constant learning rate of 1.0 better than UMAP because it's attraction shape resides within $[-1, 0]$ while UMAP's is within $[-2, 0]$. Following the same logic, we showed that UMAP can withstand a constant learning of 0.5 by making its attraction shape stay within $[-1, 0]$ (Figs. 18(f,h)).

However, the embeddings are still better if the learning rate anneals (for a wide range of parameters of a and b). This is because the goal of the algorithm is to eventually reduce ζ_{-1} to zero and it occurs when the learning rate reduces close to zero. Otherwise, the embedding becomes diffused (Fig 8(a)). On the other hand, if the learning rate is too low, to begin with, the strength of attraction and repulsion is too low, and thus no clear clusters can form (Fig 8(b)). By analyzing ζ_{-1} vs λ curve (Fig 8(c)), we see that a near optimal point is $\lambda = 0.1$ where the value of ζ_{-1} is low with considerable attraction and repulsion strength. Setting the constant learning rate to 0.1, we obtain compact clusters with clear boundaries (Fig 8(d)).

D Metrics

D.1 Procrustes Distance and Procrustes Matrix

The Procrustes distance (Gower, 1975) measures similarity between two point clouds $\{x\}$ and $\{y\}$ under linear transformations, viz. translation, scaling, and rotation. Operationally, we hold the former fixed and vary the latter until the two sets are in maximum alignment. Let $\{y'\}$ be the transformation of $\{y\}$ that

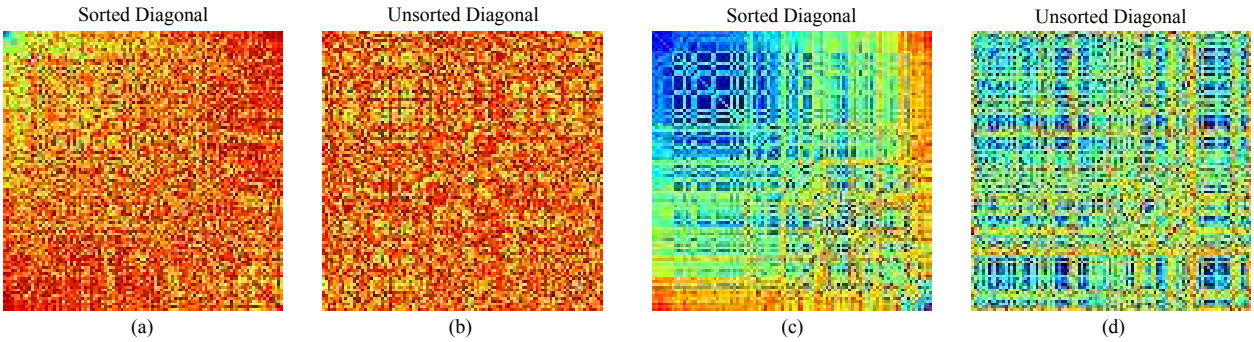


Figure 9: Effect of sorting the diagonal of Procrustes matrix on its visualization. (a) Procrustes matrix reproduced from 4(f) for the default UMAP attraction shape. The diagonal of the matrix is sorted by p_d of a sample embedding with PCA initialization. (b) The same data as in (a), but the diagonal is unsorted (or randomly sorted). (c) Procrustes matrix reproduced from 4(g) for modified attraction shape, where the diagonal of the matrix is sorted by p_d of a sample embedding with PCA initialization. (d) The same data as in (c), but the diagonal is unsorted.

achieves this objective. Then the Procrustes distance is given by

$$p_d(\{x\},\{y\}) = \sqrt{\sum_k (x_k - y'_k)}. \tag{28}$$

The Procrustes distance is a linear measure that has proven useful in a variety of settings (McInnes et al., 2018; Islam & Fleischer, 2022; Kotlov et al., 2024).

Here, we use the Procrustes distance to measure the consistency of embedding under random initialization. Let $\{x\}_p$ be a reference embedding (using PCA initialization in our experiments), and $X_r = \{\{x\}_i | i = 1, 2, 3, \ldots, N\}$ be a set of N embeddings obtained from random initialization. The similarity of the embeddings can be quantified by taking a mean and standard deviation of the strictly lower triangular values of the matrix P (reported as mean±std in Figs. 4, 10, and 11), with

$$\text{mean} = \frac{2}{N(N-1)} \sum_{i,j(i>j)} P_{i,j}, \quad \text{and} \quad \text{std} = \sqrt{\sum_{i,j(i>j)} \frac{2(P_{i,j} - \text{mean})^2}{N(N-1)}} \tag{29}$$

The indexes of X_r can be sorted such that $p_d(\{x\}_i, \{x\}_p) \leq p_d(\{x\}_{i+1}, \{x\}_p)$, so that the diagonal values of the Procrustes matrix are given by

$$P_{i,i} = p_d(\{x\}_i, \{x\}_p) \tag{30}$$

and the off-diagonal values are given by

$$P_{i,j} = p_d(\{x\}_i, \{x\}_j). \tag{31}$$

Numerically, the sorting of the diagonal adds little value. Visually, however, the ordering reveals the underlying self-similarity of the embeddings. For example, in Fig. 9(a), the sorted diagonal shows that similar embeddings clump in the upper left region of the matrix. When we use a modified attraction shape, the number of points that are similar to each other increases (as shown by the larger blue region in Fig. 9(c)), indicating the presence of a metastable point in the embedding algorithm. On the other hand, when the diagonal is unsorted, this region disappears, and any sense of similar embeddings is lost (Fig. 9(b,d)).

D.2 Trustworthiness

The trustworthiness metric (Venna & Kaski, 2001) quantifies how well local neighborhoods are preserved after dimensionality reduction:

$$T = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{y_j \in \mathrm{KNN}(y_i,k)} \max(0, r(i,j) - k) \tag{32}$$

where $KNN(y_i, k)$ is the k-NN graph in the embedding space and $r(i,j)$ is the rank of x_j in the high-dimensional k-NN graph. In practice, k is often set to 5, assessing preservation of each point's five nearest neighbors. For computational efficiency, when we report trustworthiness, we randomly sampled 10,000 indices. When comparing different embeddings, we used the same indices.

D.3 Silhouette Score

While the silhouette score (Rousseeuw, 1987) aims to assess clustering algorithms, we use it to evaluate label separation in the embeddings, i.e, how well the ground truth labels have been clustered. The idea is that the embedding algorithms naturally produce clusters and should separate the labels as much as possible. For a point y_i in a point cloud $\{y\}$, let a_i be the mean distance from y_i to other points in its own label, and let b_i be the minimal mean distance from y_i to points in any other label. The pointwise silhouette is thus given by

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \tag{33}$$

This value lies within $[-1, 1]$. A value close to 1 means that y_i fits within its own label cluster, near 0 suggests a boundary point, and close to -1 indicates failed label separation. The overall silhouette score is

$$\mathrm{SIL} = \frac{1}{N} \sum s_i. \tag{34}$$

We computed the silhouette score for the whole embedding (no random sampling) using Euclidean distances.

E Effect of Random Initialization in Additional Datasets

In the main text (Section 5.2), we showed results only on the MNIST dataset. Here we perform the same experiment on the Fashion-MNIST (FMNIST), both single-cell transcriptomes set and 20NewsGroup data (Fig. 10- 13, respectively). The main conclusion remains unchanged: modified and composite attraction shapes, such as those that increase attraction at large distances, significantly improve the consistency of reconstruction.

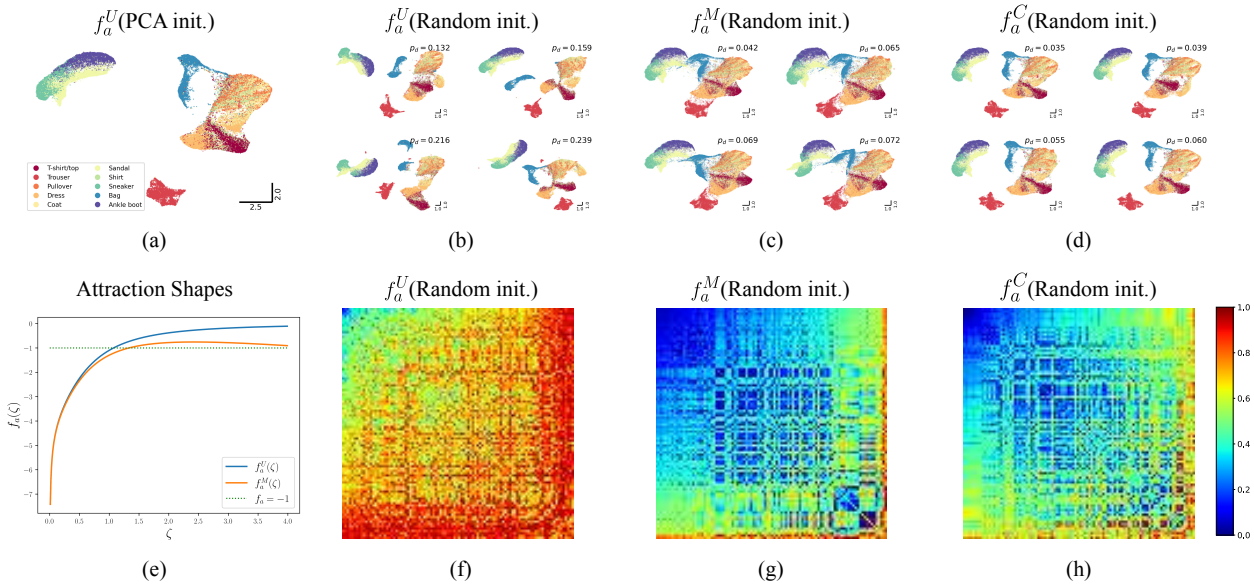


Figure 10: Effect of random UMAP initialization on different attraction shapes on FMNIST data. (a) Mapping using PCA as a standard. (b-d) Four mappings with the lowest Procrustes distance (p_d) from the embedding in (a) for (b) default, (c) modified, and (d) composite attraction shapes. (e) Default UMAP and modified attraction shapes. (f-h) Procrustes matrix obtained from 100 runs of (f) default (0.72 ± 0.15), (g) modified (0.38 ± 0.19), and (h) composite (0.43 ± 0.18) attraction shapes.

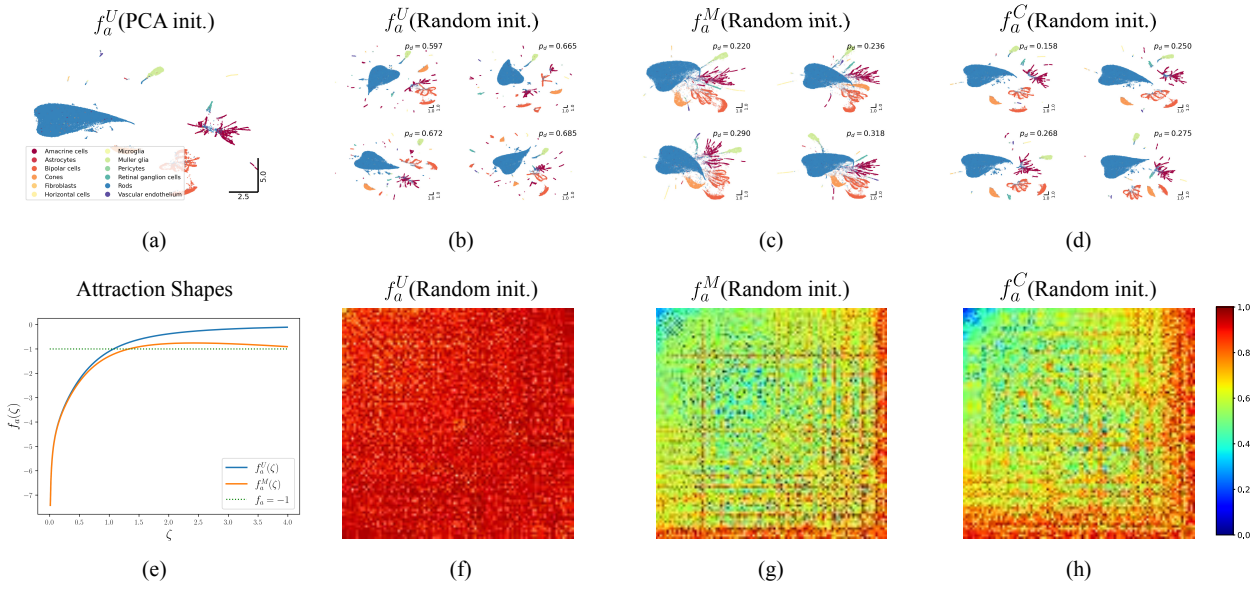


Figure 11: Effect of random UMAP initialization on different attraction shapes on single-cell transcriptomes data. (a) Mapping using PCA as a standard. (b-d) Four mappings with the lowest Procrustes distance (p_d) from the embedding in (a) for (b) default, (c) modified, and (d) composite attraction shapes. (e) Default UMAP and modified attraction shapes. (f-h) Procrustes matrix obtained from 100 runs of (f) default (0.91 ± 0.06), (g) modified (0.61 ± 0.13), and (h) composite (0.64 ± 0.15) attraction shapes.

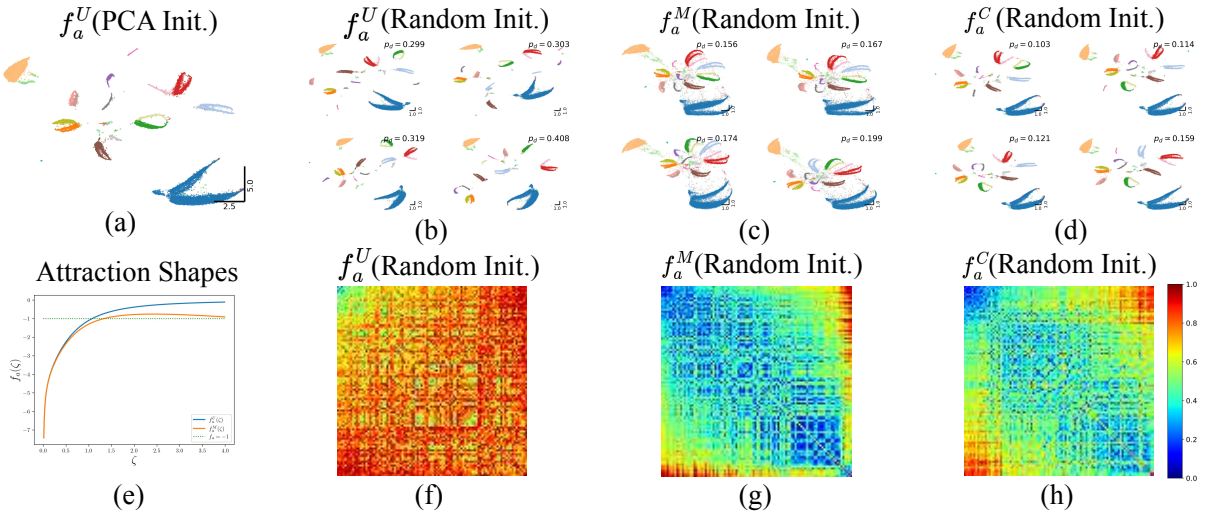


Figure 12: Effect of random UMAP initialization on different attraction shapes on Transcriptomes data from (Shekhar et al., 2016) (a) Mapping using PCA as a standard. (b-d) Four mappings with the lowest Procrustes distance (p_d) from the embedding in (a) for (b) default, (c) modified, and (d) composite attraction shapes. (e) Default UMAP and modified attraction shapes. (f-h) Procrustes matrix obtained from 100 runs of (f) default (0.77 ± 0.13), (g) modified (0.42 ± 0.17), and (h) composite (0.48 ± 0.17) attraction shapes.

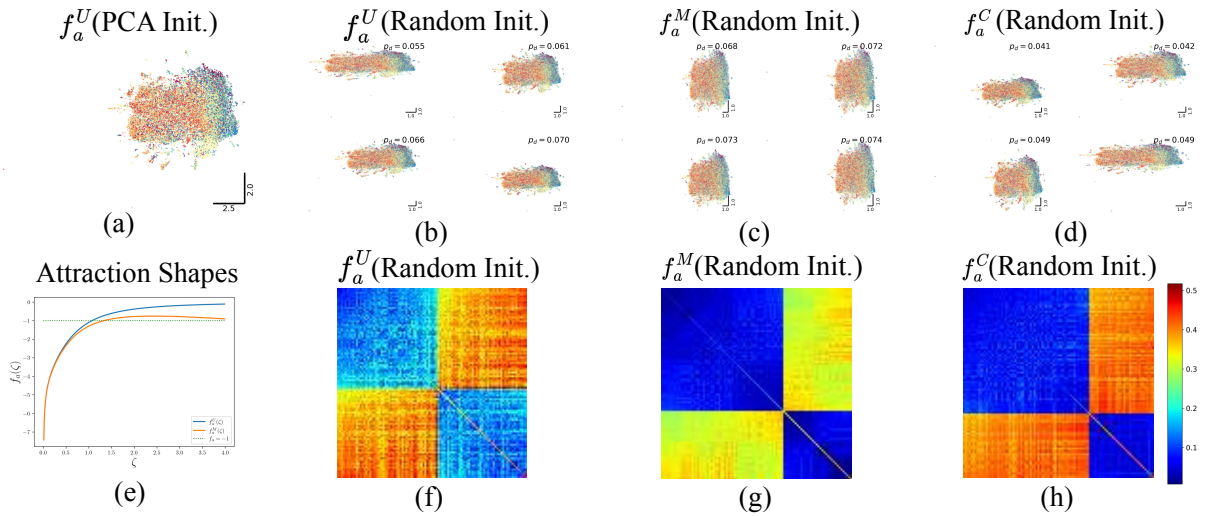


Figure 13: Effect of random UMAP initialization on different attraction shapes on 20NewsGroup (20NG) data. (a) Mapping using PCA as a standard. (b-d) Four mappings with the lowest Procrustes distance (p_d) from the embedding in (a) for (b) default, (c) modified, and (d) composite attraction shapes. (e) Default UMAP and modified attraction shapes. (f-h) Procrustes matrix obtained from 100 runs of (f) default (0.27 ± 0.13), (g) modified (0.18 ± 0.14), and (h) composite (0.22 ± 0.17) attraction shapes. In 20NG, the embedding consistently settles into one of two dominant modes. The modified and composite shapes show a clear bias toward the mode that lies nearer to the PCA-initialized configuration.

F Discussion regarding $b > 1$ in UMAP

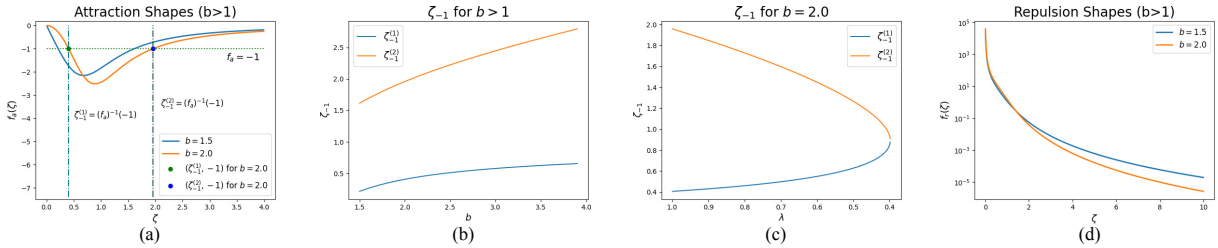


Figure 14: Attraction and repulsion shapes for $b > 1$. (a) Attraction shapes. (b) ζ_{-1} as b varies. (c) ζ_{-1} as λ varies for $b = 2.0$. (d) Repulsion shapes.

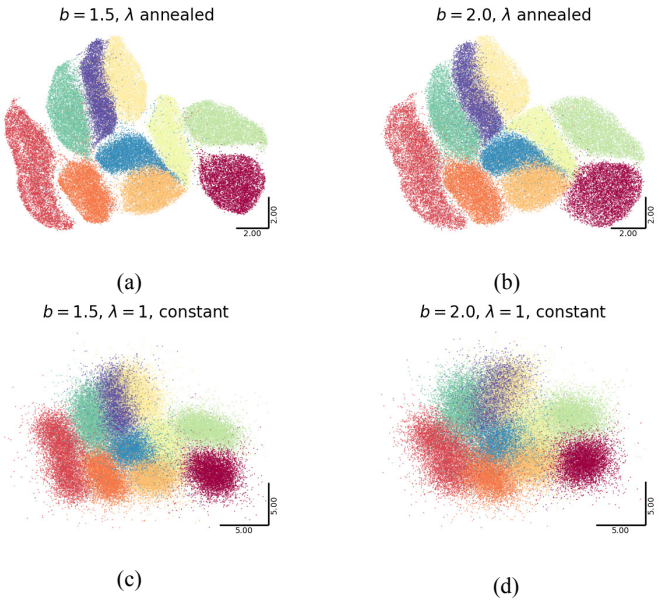


Figure 15: UMAP embeddings when $b > 1$. Left column: $b = 1.5$, right column: $b = 2.0$. Top row: λ annealed, bottom row: $\lambda = 1$, constant. For $b = 2.0$, $\zeta_{-1}^{(2)}$ is larger than that of $b = 1.5$ and thus, the clusters are more diffused (i.e., they overlap more).

In the main text, we focused solely on $b \leq 1$. Here we provide a brief note on $b > 1$. The attraction shape changes significantly for $b > 1$ (Fig. 14 (a)). The shape crosses $f_a = -1$ line at two points, denoted as $\zeta_{-1}^{(1)}$ and $\zeta_{-1}^{(2)}$ (with $\zeta_{-1}^{(2)} > \zeta_{-1}^{(1)}$). The embedding contracts as long as $f_a < \zeta_{-1}^{(1)}$ and $f_a > \zeta_{-1}^{(2)}$. The in between region, $\zeta_{-1}^{(1)} < f_a < \zeta_{-1}^{(2)}$, causes expansion. Thus, $\zeta_{-1}^{(2)}$ is the primary point around which a point contracting from a larger distance will oscillate.

As b increases, the gap between $\zeta_{-1}^{(1)}$ and $\zeta_{-1}^{(2)}$ increases (Fig. 14 (b)). On the other hand, decreasing learning rate, λ, decreases this gap, eventually the attraction shape confines to $[-1, 0]$ (Fig. 14 (c)). As b increases, repulsion strength at larger distances decreases (Fig. 14 (d)).

Overall, based on our analysis, increasing b, diffuses the embedding more and reduces inter-cluster distance (Fig. 15 (a→b) and (c→d)). While diffused clusters are easily visible in the embeddings, the reduction of inter-cluster distance isn't that clear. We quantify it using average inter-label distance (for MNIST, inter-cluster and inter-label distances are highly correlated). In $b = 1.5$ (Fig. 15 (a)), the inter-label distance is 6.00 and for $b = 2.0$ (Fig. 15 (b)), it is 5.45 (if the embeddings are rescaled to unit variance, the inter-label distances are 1.79 and 1.77, respectively). This follows our arguments regarding attraction and repulsion.

G Varying a in UMAP

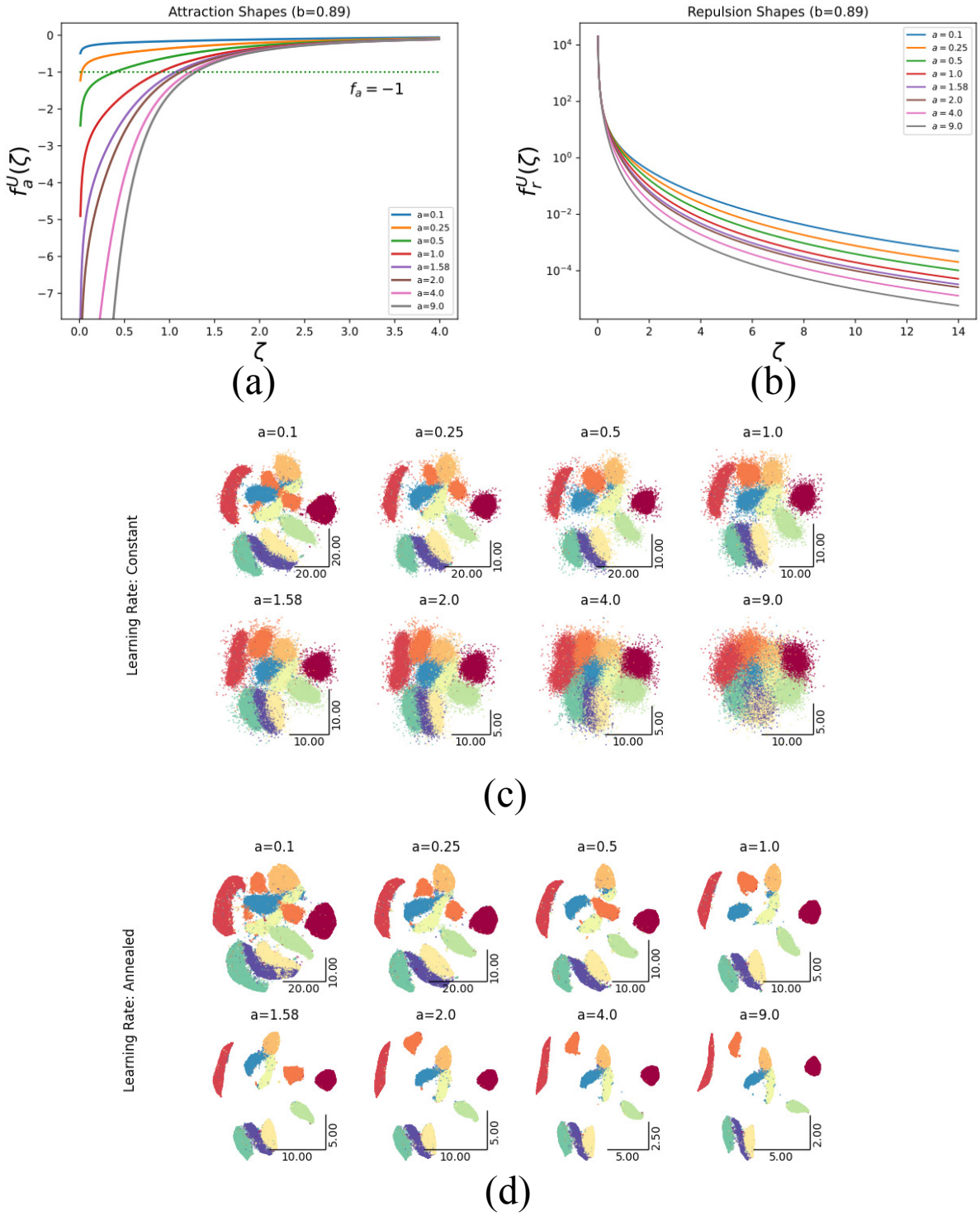


Figure 16: Effect of varying a in UMAP. (a) Attraction and (b) repulsion shapes as a varies ($b = 0.89$). MNIST embeddings for varying a when the learning rate is (c) constant and (d) annealed.

In section 5.3 of the main text, we varied b to explore the effect of attraction and repulsion shapes, and discussed compactness and structure creation within the embeddings. The second controllable parameter is a. A simple analysis of the kernel shows that a effectively rescales the distances ($1/\log(1 + ad_{ij}^{2b}) = 1/\log(1 + (a^{1/2b}d_{ij})^{2b}) = 1/\log(1 + \tilde{d}_{ij}^{2b})$, and thus, $\tilde{d}_{ij} = a^{1/2b}d_{ij}$)). In principle, an optimum for one value of a can therefore be obtained by rescaling an optimum for another value of a. However, the optimization procedure is not scale-equivariant in practice: using the same initialization scale, learning-rate schedule, and finite optimization budget across different values of a can lead to trajectories that are not simple rescalings of one another. As a result, the final embeddings can differ in practice, even though the underlying objective changes primarily in scale.

Varying a, varies the attraction and repulsion shapes (Fig. 16(a,b)). For small values of a, the attraction strength decreases quickly as ζ increases, whereas for higher values of a, this decrease is gradual. As a increases, the attraction shape starts falling below -1 with increasing ζ_{-1} and thus, when the learning rate remains constant, the embeddings become diffused and overlapping with clusters (Fig. 16(c)). This effect is remedied when the learning rate is annealed (Fig. 16 (d)).

On the contrary, the repulsion strength remains higher for small values of s (as ζ increases). As a result, following our exploration in Section 5.3, the inter-cluster distances are higher for smaller values of a and lower for large values of a.

Overall, two broad dynamics are observed here:

- For smaller a, long-range attraction is lower, but repulsion is higher. Repulsion causes larger inter-cluster distances (qualitatively, centroid to centroid), but attraction fails to build structures/clusters (as attraction shape approaches 0 faster).
- For large a, long-range attraction is higher (ζ_{-1} is also higher), and repulsion is lower. Repulsion causes small inter-cluster distances. Since ζ_{-1} is high, we can obtain a good structure/cluster if ζ_{-1} is reduced toward zero by learning rate annealing.

Due to the first dynamic, even though we can increase repulsion by decreasing a, it causes a worse case of near-sightedness, and thus we fail to obtain compact clusters.

H Additional Discussion on Fig. 6

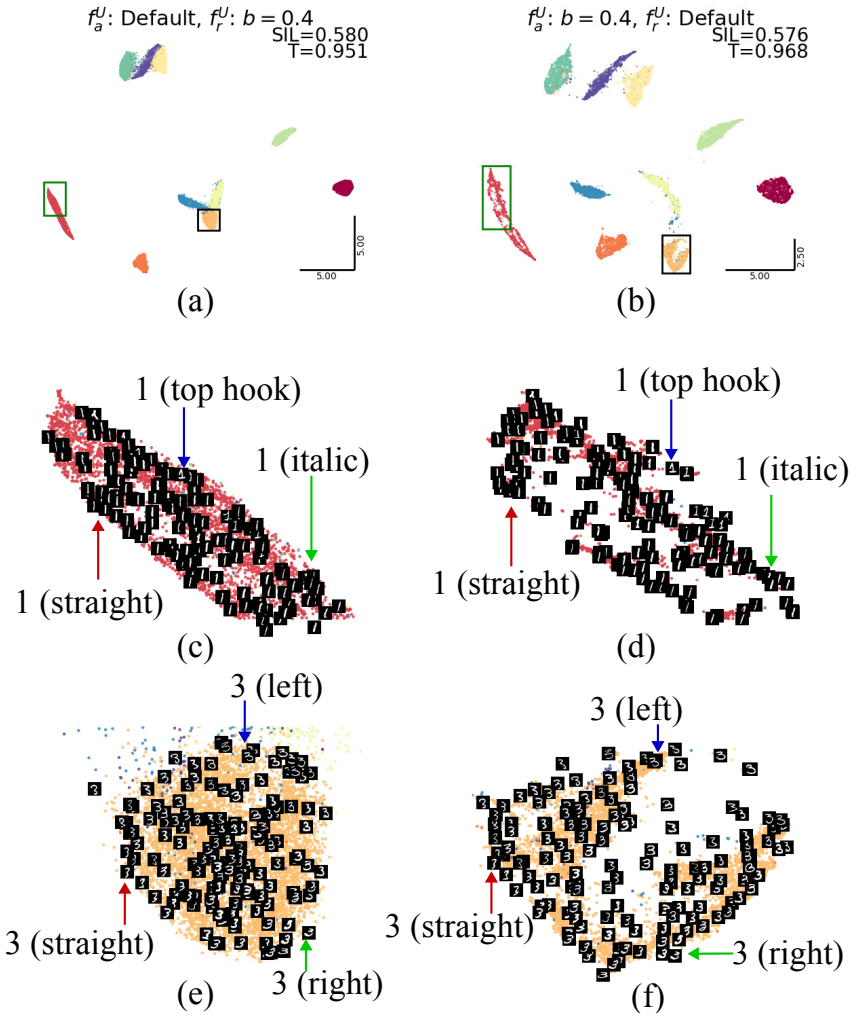


Figure 17: Embedding of the MNIST dataset with (a) default attraction shape but repulsion shape with $b = 0.4$ and (b) default repulsion shape but attraction shape with $b = 0.4$. (c) Cluster of label 1 from (a, green rectangle). (d) Cluster of label 1 from (b, green rectangle). (e) Cluster of label 3 from (a, black rectangle). (f) cluster of label 3 from (b, black rectangle).

In this section, we examine additional structures from Figs. 6 (a,b), reproduced in Figs. 17 (a,b). First, we examine the cluster of labels 1. Default attraction and stronger repulsion ($b = 0.4$) (Fig. 17 (c)) exhibit a gradient of writing variation, but the individual writing styles are not well separated. When we use $b = 0.4$ for attraction and default repulsion (Fig. 17 (d)), additional structures and branches emerge, that separate different writing styles (1 with top hat, written in italic vs. upright strokes).

We observe the same pattern for label 3. Under default attraction and stronger repulsion (Fig. 17(e)), the cluster forms a smooth continuum of styles, from right-slanted italic 3s, through more upright forms, to left-slanted italic 3s, without a clear separating boundary. With stronger attraction and default repulsion (Fig. 17(f)), the cluster organizes along a trajectory that separates the right- and left-slanted variants, producing a visible "tear" (hypothetically, a loop-like structure cycling through styles: straight → right-slanted → straight → left-slanted).

I More on NEG-t-SNE

I.1 Comparison to UMAP

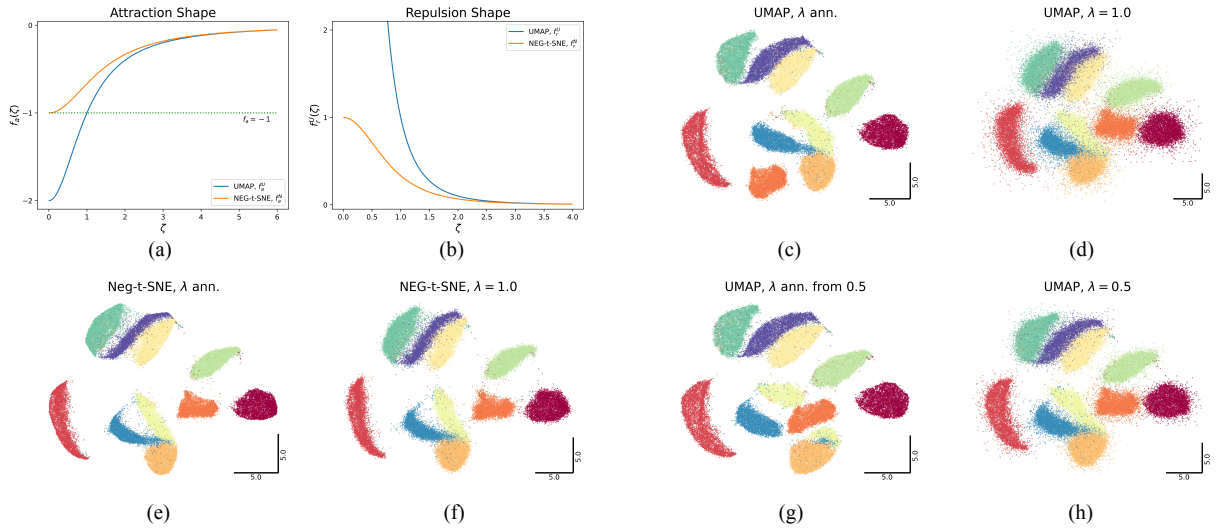


Figure 18: Sensitivity of UMAP and NEG-t-SNE to learning rate on the MNIST dataset. (a) Attraction and (b) repulsion shapes for UMAP ($a = 1$, $b = 1$) and NEG-t-SNE. (c,d) UMAP is very sensitive to the learning rate λ, as $f_a^U < -1$ as the separation distance ζ decreases. Thus, without annealing, the clusters become fuzzy. (e,f) NEG-t-SNE is less sensitive to λ as $f_a^N \in [-1, 0]$ always, and the clusters are thus less fuzzy even when not annealed. (g,h) Confining f_a^U to $[-1, 0]$ by setting $\lambda = 0.5$ shows less sensitivity to λ.

Figure 18 shows the shapes of UMAP and NEG-t-SNE, along with various MNIST embeddings. When the learning rate is annealed, both UMAP and NEG-t-SNE show similar output (Figs. 18(c,e)). However, when the learning rate is a constant value of 1, the UMAP shows a fuzzy structure, while NEG-t-SNE shows a structure with much cleaner boundaries (Fig. 18(d,f)). The discussion in Section 5.1 suggests that constraining f_a^U within $[-1, 0]$ can potentially result in less fuzzy clusters for fixed λ. We have seen this previously in Fig. 3(c,d) as well, where UMAP provided better embedding and clustering when the learning rate for attraction (λ_a) was $\lesssim 0.5$. A straightforward way to achieve this is to initialize λ to 0.5, which satisfies Proposition 4.1 for all ζ. The resulting embeddings, shown in Figs. 18 (g) and (h), confirm that clusters are similar to those of NEG-t-SNE's, and for a constant $\lambda = 0.5$, the clusters are less fuzzy than before as predicted with sharper boundaries (Fig. 18(d,h)). There are still a few points outside the clusters due to the characteristics of UMAP's repulsion shape, which NEG-t-SNE solves.

Next, we can introduce the parameters a and b into NEG-t-SNE (essentially the formulation of Parametric UMAP; see Proposition 5.1). The affinity function becomes $q_{ij}^N = 1/(2 + a\zeta^{2b})$, and the attraction and repulsion shapes become

$$f_a^N = -\frac{2ab\zeta^{2(b-1)}}{2 + a\zeta^{2b}}, \tag{35}$$

and

$$f_r^N = \frac{2ab\zeta^{2(b-1)}}{(1 + a\zeta^{2b})(2 + a\zeta^{2b})}, \tag{36}$$

respectively. For $0 < b < 1$, both shapes become unbounded as $\zeta \to 0$. Thus, NEG-t-SNE will face similar numerical challenges to UMAP if a and b vary, and corresponding limitations carry over. One notable distinction is that, compared to UMAP, the attraction shape attains a lower minimum distance (ζ_{-1}) for the attraction. While this may enhance cluster formation, it approaches zero faster (increased near-sightedness as distance increases), potentially diminishing its effectiveness for attraction over longer distances.

I.2 Comparison to Parametric UMAP

Parametric UMAP was initially trained with the original UMAP objective (Sainburg et al., 2021), but later work adopted a numerically stable, log-sigmoid–based modified cross-entropy loss (Shi et al., 2023). This modification makes Parametric UMAP and Neg-t-SNE (Damrich et al., 2023) equivalent (Proposition 5.1). We show the equivalence below.

Proof of Proposition 5.1

Proof. The kernel function under this modification becomes

$$q_{ij}^P = -\log\left(1 + a||y_i - y_j||_2^{2b}\right). \tag{37}$$

The attractive term is

$$\begin{aligned} -\text{logsigmoid}(q_{ij}^P) &= -\log\left(\frac{1}{1+\exp(-q_{ij}^P)}\right) && (38)\\ &= -\log\left(\frac{1}{2 + a||y_i - y_j||_2^{2b}}\right) && (39)\\ &= -\log(q_{ij}^N). && (40) \end{aligned}$$

And the repulsive term is

$$\begin{aligned} \text{logsidmoid}(q_{ij}^P) - q_{ij}^P &= \log\left(\frac{1}{2 + a||y_i - y_j||_2^{2b}}\right) + \log\left(1 + a||y_i - y_j||_2^{2b}\right) && (41)\\ &= \log\left(\frac{1 + a||y_i - y_j||_2^{2b}}{2 + a||y_i - y_j||_2^{2b}}\right) && (42)\\ &= \log\left(1 - \frac{1}{2 + a||y_i - y_j||_2^{2b}}\right) && (43)\\ &= \log(1 - q_{ij}^N). && (44) \end{aligned}$$

Both these are NEG-t-SNE with explicit parameters a and b (while in Neg-t-SNE these are set to 1). □

J Alternate Dimensionality Reduction Algorithms

The alternative algorithms we consider use the same kernel function as UMAP (with $a = 1$ and $b = 1$ in their low-dimensional weight):

$$q_{ij} = \frac{1}{1 + ||y_i - y_j||_2^2}. \tag{45}$$

In this section, we first discuss the TriMap (Amid & Warmuth, 2019) algorithm. Even though, this algorithm relies on triplets (and not pairwise interactions), this works as a primer for analyzing attraction and repulsion that are a bit more involved than UMAP. This discussion is followed by Pairwise Controlled Manifold Approximation (PaCMAP) (Wang et al., 2021) and it's extension Pairwise Controlled Manifold Approximation with Local Adjusted Graph (LocalMAP) (Wang et al., 2025) that modify TriMAP's loss function that works for pairwise interactions. We then tackle t-SNE (Van der Maaten & Hinton, 2008). Then, we provide a short note on SNE (Hinton & Roweis, 2002) that uses an alternate kernel function and finally end the section by briefly discussing multidimensional scaling (Borg & Groenen, 2007).

J.1 TriMap

TriMap (Amid & Warmuth, 2019) optimizes low-dimensional using a triplet loss

$$\mathcal{L}^T = \sum_{(i,j,k)} w_{ijk} \frac{1}{1 + \frac{q_{ij}}{q_{ik}}}, \tag{46}$$

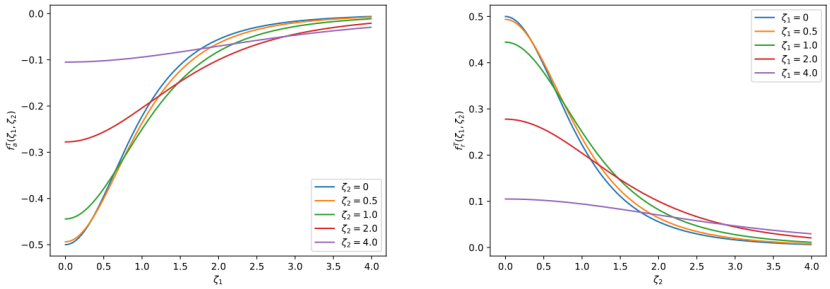


Figure 19: (a) Attraction shapes of TriMap for different ζ_2 and (b) repulsion shapes of TriMap for different ζ_1.

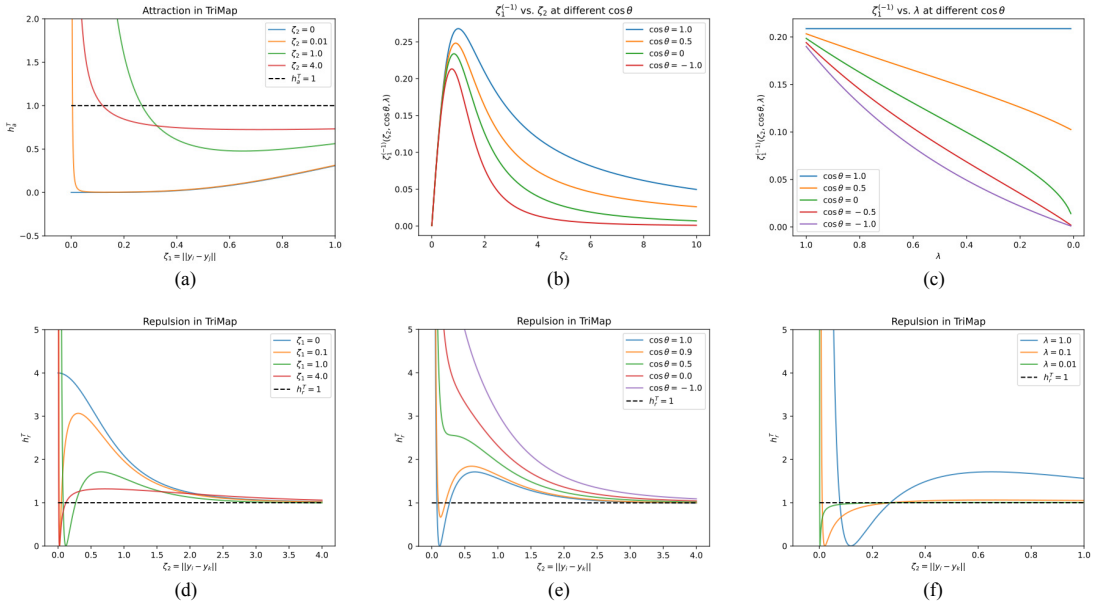


Figure 20: Attraction and repulsion behavior in TriMap. (a) h_a^T vs ζ_1 for different ζ_2. Any values below the dotted line indicate attraction. Like UMAP, TriMAP shows attraction and repulsion for nearest neighbors (y_i, y_j). (b,c) Unlike UMAP, the minimum distance for contraction ($\zeta_1^{(-1)}$) varies due to dependence on (b) ζ_2 and (c) λ; the function $\cos\theta$ regulates the range of these values. (d-f) Repulsion in TriMap by varying (d) ζ_1, (e) $\cos\theta$, and (e) λ. Values above (below) the dotted line indicate repulsion (attraction). While the repulsion force of UMAP shows only repulsion, that of TriMap can provide both attraction and repulsion. Unless otherwise labeled, $\zeta_1 = 1.0$, $\zeta_2 = 0.5$, $\cos\theta = 1.0$, and $\lambda = 1.0$.

where w_{ijk} is the weight of the triplet (y_i, y_j, y_k), y_j is in the k-nearest neighbor set of y_i in the high dimension, and y_k is a far-away point. When minimized, we expect that y_i and y_j attract each other, while y_i and y_k repel each other. The update equations are

$$y_i^{t+1} = y_i^t + \lambda f_a^T(\zeta_1^t, \zeta_2^t)(y_i^t - y_j^t) + \lambda f_r^T(\zeta_1, \zeta_2)(y_i^t - y_k^t), \tag{47}$$

$$y_j^{t+1} = y_j^t - \lambda f_a^T(\zeta_1^t, \zeta_2^t)(y_i^t - y_j^t), \tag{48}$$

$$y_k^{t+1} = y_k^t - \lambda f_r^t(\zeta_1^t, \zeta_2^t)(y_i^t - y_k^t), \tag{49}$$

where $\zeta_1 = ||y_i - y_j||_2$ is the distance between nearest neighbors, $\zeta_2 = ||y_i - y_k||_2$ is the distance from the faraway point, and f_a^T and f_r^T are attraction and repulsion shapes of TriMap, respectively. Unlike UMAP, the attractive and repulsive components are non-separable and the shapes depend on two distance measures (making them 2D). The functional form of the attraction shape is

$$f_a^T(\zeta_1, \zeta_2) = -\frac{2(1+\zeta_2^2)}{(2+\zeta_1^2+\zeta_2^2)^2}, \tag{50}$$

and the repulsion shape is

$$f_r^T(\zeta_1, \zeta_2) = \frac{2(1+\zeta_1^2)}{(2+\zeta_1^2+\zeta_2^2)^2}. \tag{51}$$

The attraction and repulsion shapes (Fig. 19) of TriMap shows similar trends of that of UMAP. However, the minium value of repulsion shape is -0.5; thus, unlike UMAP there is no position flipping in TriMAP due to attraction alone. However, since Eqs. (47-49) are not decoupled between attractive and repulsive terms, Propositions 4.1 and 4.2 do not apply. Focusing on attraction first, we show

Proposition J.1. *Update equations (47)-(49) provide a contraction if*

$$h_a^T(\zeta_1, \zeta_2, \theta, \lambda) < 1, \tag{52}$$

where $h_a^T(\zeta_1, \zeta_2, \theta, \lambda) = (1 + 2\lambda f_a^T)^2 + 2(1 + 2\lambda f_a^T)\lambda f_r^t \frac{\zeta_2}{\zeta_1}\cos\theta + (\lambda f_r^T)^2 \frac{\zeta_2^2}{\zeta_1^2}$ *and* θ *is the angle between the vectors* $(y_i - y_j)$ *and* $(y_i - y_k)$*, i.e.,* $cos\theta = \frac{(y_i^t - y_j^t)^T(y_i^t - y_k^t)}{||y_i^t - y_j^t||_2 ||y_i^t - y_k^t||_2}$.

Proof. We require

$$||y_i^{t+1} - y_j^{t+1}||_2^2 < ||y_i^t - y_j^t||_2^2. \tag{53}$$

From Eq. (47) and (48): $y_i^{t+1} - y_j^{t+1} = (1 + 2\lambda f_a^T)(y_i^t - y_j^t) + \lambda f_r^T(y_i^t - y_k^t)$. Putting this value in Eq. (53), we obtain the desired inequality. □

Inequality (52) depends on ζ_1, ζ_2, $\cos\theta$ and λ. In particular, the value of $\zeta_1 = ||y_i - y_j||_2$, where we want a contraction, is coupled with additional variables. Figure 20(a) shows the attraction behavior for various values of ζ_2, while $\cos\theta = 1$ and $\lambda = 1$. The values below the dotted line indicate attraction (and thus contraction of distance ζ_1), whereas the values above indicate repulsion (and therefore expansion of distance ζ_1). The value where the dotted line and h_a^T meet gives the minimum distance for contraction ($\zeta_1^{(-1)}$) [analogous to ζ_{-1} of UMAP], which we define as

$$\zeta_1^{(-1)}(\zeta_2, \theta, \lambda) = \arg\min_{\zeta_1} |h_a^T(\zeta_1, \zeta_2, \theta, \lambda) - 1|, \tag{54}$$

$$\text{s.t.} \quad \zeta_1 \geq 0. \tag{55}$$

$\zeta_1^{(-1)}$ has a finite value and is > 0 for most cases (Fig. 20(b,c)). As a result, TriMap can show behavior similar to UMAP and thus require learning rate annealing or a similar approach (in the TriMap implementation, the authors use the delta-bar-delta (Jacobs, 1988) method under appropriate initialization).

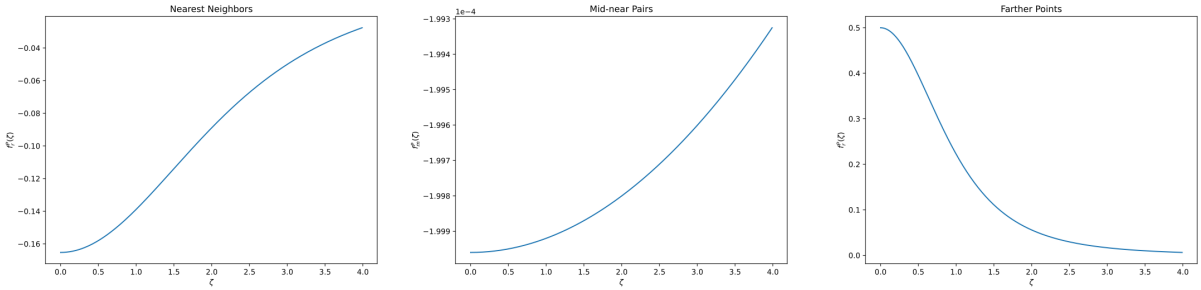


Figure 21: Attraction and repulsion shapes of PaCMAP. (a,b) Attraction shapes for (a) nearest-neighbor and (b) mid-near points (note that the values are on the order of 10^{-4}). (c) Repulsion shape for farther pairs. $\lambda = 1$ for all figures.

Proposition J.2. *Update equations (47)-(49) provide expansion if*

$$h_r^T(\zeta_1, \zeta_2, \theta, \lambda) > 1, \tag{56}$$

where $h_r^T(\zeta_1, \zeta_2, \theta, \lambda) = (1 + 2\lambda f_r^T)^2 + 2\lambda f_a^T(1 + \lambda f_r^T)\frac{\zeta_1}{\zeta_2}\cos\theta + (\lambda f_a^T)^2\frac{\zeta_1^2}{\zeta_2^2}$.

Proof. We require

$$||y_i^{t+1} - y_k^{t+1}||_2^2 > ||y_j t - y_k^t||_2^2. \tag{57}$$

From Eq. (47) and (49): $y_j^{t+1} - y_k^{t+1} = (1 + 2\lambda f_r^T)(y_i^t - y_k^t) + \lambda f_a^T(y_i^t - y_j^t)$. Putting this value in Eq. (57) we obtain inequality (56). □

Inequality (56) also depends on the set ζ_1, ζ_2, $\cos\theta$ and λ. Here, we are interested in the expansion of $\zeta_2 = ||y_i - y_k||_2$. Figures 20(d-f) show the repulsion behavior by varying the other quantities. Any values above the dotted line indicate repulsion (and thus expansion of distance), while the values below indicate attraction. The striking difference compared to UMAP is that repulsion in TriMap can cause contraction instead of expansion. Since this anomaly occurs for small distances, it can be avoided by an appropriate initialization and choice of triplets.

J.2 Pairwise Controlled Manifold Approximation (PaCMAP)

PaCMAP (Wang et al., 2021) optimizes low-dimensional embedding at different scales. The loss function is

$$\mathcal{L}^P = w_{NB}\sum_{(i,j)\in NN}\frac{1}{1+10q_{ij}} + w_{MN}\sum_{(i,k)\in MN}\frac{1}{1+10000q_{ik}} + w_{FP}\sum_{(i,l)\in FP}\frac{1}{1+\frac{1}{q_{il}}}, \tag{58}$$

where w_{NB}, w_{MN}, and w_{FP} are weights of the nearest neighbor (NN) pairs, mid-near (MN) pairs, and further pairs (FP), respectively (details in Appendix K). The first two terms provide attraction, whereas the last term provides repulsion. A closer look at the loss function reveals that the function is a modified form of TriMap's triplet loss. For the attractive terms, it replaces TriMap's affinity for distant points with constant terms (1/10 for nearest neighbors, 1/10000 for mid-near pairs, and 1 for farther points). This loss function is thus separable into three terms and decouples the update equations. The update equations of the nearest neighbor term are

$$y_i^{t+1} = y_i^t + \lambda f_a^P(\zeta_1^t)(y_i^t - y_j^t), \tag{59}$$

$$y_j^{y+1} = y_j^t - \lambda f_a^P(\zeta_1^t)(y_i^t - y_j^t), \tag{60}$$

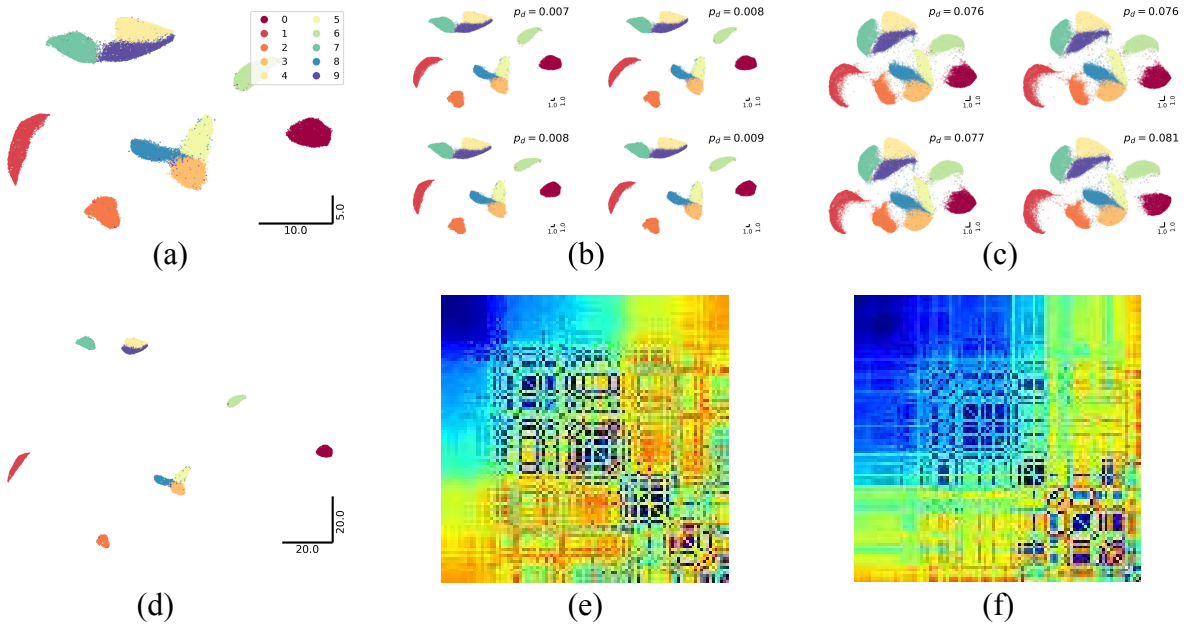


Figure 22: PaCMAP behavior for different conditions. (a) PaCMAP of MNIST data with PCA initialization. (b,c) Four samples that best match with (a) by (b) original PaCMAP and (c) PaCMAP with modified attraction shape, $f_a^M(\zeta) = f_a^P(\zeta) - 0.001\zeta$, when randomly initialized. (d) PaCMAP of MNIST with a modified repulsion shape $f_r^M = f_r^P + 0.00005$. (e,f) Procrustes matrix for (e) original PaCMAP (0.51 ± 0.22) and (f) PaCMAP with modified attraction shape (0.43 ± 0.22). Similar to UMAP, increased attraction at farther distances show improved consistency, while increased repulsion shows smaller clusters and larger inter-cluster distances.

of the mid-near pairs are

$$y_i^{t+1} = y_i^t + \lambda f_m^P(\zeta_2^t)(y_i^t - y_k^t), \tag{61}$$
$$y_k^{y+1} = y_k^t - \lambda f_m^P(\zeta_2^t)(y_i^t - y_k^t), \tag{62}$$

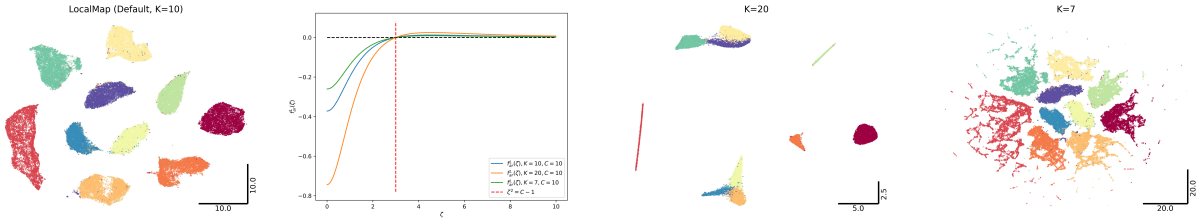
and of the farthest pairs are

$$y_i^{t+1} = y_i^t + \lambda f_r^P(\zeta_3^t)(y_i^t - y_l^t), \tag{63}$$
$$y_l^{t+1} = y_l^t - \lambda f_r^P(\zeta_3^t)(y_i^t - y_l^t), \tag{64}$$

where $\zeta_1 = ||y_i - y_j||_2$, $\zeta_2 = ||y_i - y_k||_2$, $\zeta_3 = ||y_i - y_l||_2$ are distances, f_a^P and f_m^P are attraction shapes for nearest neighbors and mid-near pairs, respectively, and f_r^P is the repulsion shape for the farthest pairs. Correspondingly, the functional forms of the shapes are

$$f_a^P(\zeta) = -\frac{20}{(11 + \zeta^2)^2}, \tag{65}$$
$$f_m^P(\zeta) = -\frac{20000}{(10001 + \zeta^2)^2}, \tag{66}$$
$$f_r^P(\zeta) = \frac{2}{(2 + \zeta^2)^2}. \tag{67}$$

f_a^P and f_m^P follow Proposition 4.1, and f_r follows Proposition 4.2 (Fig. 21). The attraction is quite low compared to UMAP, but it is good enough for a wide range of learning rates (modulated by the Adam algorithm (Kingma & Ba, 2015)); with $w_{NB} = 3$ the maximum attraction is always below 0.5 preventing any flips during attractin update. Typically, PaCMAP initializes the embedding within a small sphere in the low



(a) (b) (c) (d)

Figure 23: Behavior of LocalMAP on MNIST data. (a) Default embedding. (b) Attraction-repulsion shape of nearest neighbors. A value below (above) the dotted line indicates attraction (repulsion). Transition from attraction to repulsion occurs as ζ increases and crosses $\sqrt{C-1}$. Following implementation of LocalMAP, we used $C = 10$. (c) When K is large ($= 20$), repulsion dominates and clusters become compact. (d) When K is small ($= 7$), attraction dominates and clusters break up.

dimension (e.g., in (Wang et al., 2021), the initialization is often on the order of 10^{-3}) and relies on repulsion to separate the individual clusters. Overall, it mostly recovers UMAP's clustering properties (especially for MNIST) with improved ordering. The consistency under random initialization is better than UMAP (Fig. 22(b,e)) which can be improved further using a modified attraction (Fig. 22(c,f)). Increasing repulsion by adding a small value to the repulsion shape increases compactness of the embedding (by increasing inter-cluster distance).

J.3 Pairwise Controlled Manifold Approximation with Local Adjusted Graph (LocalMAP)

LocalMAP (Wang et al., 2025) is an iteration of the PaCMAP algorithm. One of the defining features of LocalMAP is the separation of all 10 clusters of the MNIST data (Fig. 23(a)), with behavior similar to the ones in Figs. 5(g,h,j,k) and Figs. 6(a,b). Here, we explore the interplay of attractive and repulsive forces on the compactness and connectedness of clusters.

LocalMAP performs PaCMAP and then does additional optimization on the attraction to decouple some clusters. To this end, it minimizes the following loss function

$$\mathcal{L}^L = \sum_{(i,j)\in NN} \frac{K}{\frac{1}{\sqrt{q_{ij}}} + C\sqrt{q_{ij}}} + \sum_{(i,l)\in FP} \frac{1}{1+\frac{1}{q_{il}}}. \tag{68}$$

The first term amalgamates the attractive and repulsive nature of the triplet loss function that works on the same pair. In one regime, this function causes attraction, while in the other, it causes repulsion. The second term is identical to the ones in PaCMAP; the only difference is that the algorithm resamples further pairs every few iterations. Thus, we analyze only the first term involving nearest neighbors. The update equations are

$$y_i^{t+1} = y_i^t + \lambda f_{ar}^L(\zeta_1^t)(y_i^t - y_j^t), \tag{69}$$

$$y_j^{y+1} = y_j^t - \lambda f_{ar}^L(\zeta_1^t)(y_i^t - y_j^t), \tag{70}$$

where f_{ar}^L is the attraction-repulsion shape given by

$$f_{ar}^L(\zeta) = -\frac{K(C-1-\zeta^2)}{2\sqrt{1+\zeta^2}(1+C+\zeta^2)^2}. \tag{71}$$

The update provides contraction as long as $\zeta^2 < C-1$ and $-1 < \lambda f_{ar}^L < 0$ (from Proposition 4.1). When $\zeta^2 > C-1$, $f_{ar} > 0$, and by Proposition 4.2 the update equations provide expansion. The values of K and C determine whether attraction or repulsion dominates the dynamics. In LocalMAP implementation, both C and K are set to 10 (Fig. 23(b), and the strength of attraction is higher than PaCMAP (Fig. 21(a)). As a result, when $\zeta > 3$, the nearest neighbors face repulsion, causing pairs bridging two clusters to separate. The value of K, working as a scaling parameter for the forces, regulates this separation.

Using Proposition 4.1, $\lambda f_{ar}^{L}(0) \geq -\frac{1}{2}$ gives the maximum values of K, and the maximum attraction possible by f_{ar}^{L}, without flipping the placements of the pairs. (We would want to avoid flipping the pairs at the LocalMAP optimization to preserve the ordering from PaCMAP; otherwise, it may inhibit cluster separation.) This simplifies (71) to $K \leq \frac{\lambda(1+C)^2}{C-1}$, which for $C = 10$ and $\lambda = 1$ gives $K \lesssim 13.44$. Moreover, at a higher value of $K \gtrsim 13.44$, the repulsive forces dominate, and the clusters become more compact, but objective of LocalMAP fails (the bridge between clusters persists in Fig. 23(c) for $K = 20$). On the other hand, as K decreases, attractive forces dominate (because repulsive forces are too low), and the embedding shows the breaking up of existing clusters (Fig. 23(d) for $K = 7$, which mimics the one in Fig. 6(b)).

J.4 t-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE (Van der Maaten & Hinton, 2008) optimizes pairwise distances. The loss function is

$$\mathcal{L} = -\sum_{i,j} w_{i,j} \log\left(\frac{q_{i,j}}{\sum_{k\neq l} q_{k,l}}\right), \tag{72}$$

where $w_{i,j}$ is the weight of the pair. The original implementation of t-SNE considers all the pairs (not just nearest neighbors). This loss function decomposes into attraction and repulsion forces by

$$\mathcal{L} = \sum_{i,j}\left[-w_{i,j}\log\left(q_{i,j}\right) + w_{i,j}\log\left(\sum_{k\neq l} q_{k,l}\right)\right]. \tag{73}$$

As previously, the first term provides the attractive forces and the second term provides the repulsive forces. While the attractive term is identical to that of UMAP (with $a = 1$ and $b = 1$) and is easy to compute, the repulsive term is coupled among every pair and thus, is very costly. Using the same principles we applied for UMAP, we can write the update equations of t-SNE. Since the original t-SNE didn't rely on the nearest neighbor graph, the weight $w_{i,j}$, computed for all the pairs, is important in the update equations. The attractive update equations are

$$y_i^{t+1} = y_i^t + \lambda w_{i,j} f_a^{t-SNE}(\zeta_{i,j}^t)(y_i^t - y_j^t), \tag{74}$$

$$y_j^{t+1} = y_i^t - \lambda w_{i,j} f_a^{t-SNE}(\zeta_{i,j}^t)(y_i^t - y_j^t), \tag{75}$$

where f_a^{t-SNE} is the attraction shape of t-SNE and $\zeta_{i,j} = ||y_i - y_j||_2$. The update equation for the repulsive parts are

$$y_i^{t+1} = y_i^t + \lambda\frac{w_{i,j}}{Z}\sum_k f_r^{t-SNE}(\zeta_{i,k}^t)(y_i^t - y_k^t), \tag{76}$$

$$y_j^{t+1} = y_j^t - \lambda\frac{w_{i,j}}{Z}\sum_l f_r^{t-SNE}(\zeta_{l,j}^t)(y_l^t - y_j^t), \forall j, j \neq i, \tag{77}$$

where f_r^{t-SNE} is the repulsion shape of t-SNE and $Z = \sum_{k\neq l} q_{k,l}$. The functional forms of these shapes are

$$f_a^{t-SNE}(\zeta) = -\frac{2}{1+\zeta^2}, \text{and} \tag{78}$$

$$f_r^{t-SNE}(\zeta) = \frac{2}{(1+\zeta^2)^2}. \tag{79}$$

From the attractive update Eqs. (74)-(75), f_a^{t-SNE} follows Proposition 4.1 (with $0 < \lambda w_{i,j} f_a^{t-SNE} < -1$) and gives the minimum distance for contraction, ζ_{-1}). The repulsive update is coupled with all the pairs and thus does not have a simple relation to the repulsion shape. Rather, enabled by our experience from TriMap's analysis, we can derive the following:

Proposition J.3. *The update Eqs. (76)-(77) provide an expansion if*

$$h_r^{t-SNE}(\zeta_{i,j}, v, \theta, \lambda, w_{i,j}) > 1, \tag{80}$$

where
$h_r^{t-SNE}(\zeta_{i,j}, v, \theta, \lambda, w_{i,j}) = (1 + 2\lambda \frac{w_{i,j}}{Z} f_r^{t-SNE}(\zeta_{i,j}))^2 + \frac{||v||_2^2}{\zeta_{i,j}^2} + 2\lambda \frac{w_{i,j}}{Z} f_r^{t-SNE}(\zeta_{i,j}) \frac{||v||_2}{\zeta_{i,j}} \cos\theta$,
$v = \lambda \frac{w_{i,j}}{Z} \Big(\sum_{k,k\neq j} f_r^{t-SNE}(\zeta_{i,k})(y_i - y_k) + \sum_{l,l\neq i} f_r^{t-SNE}(\zeta_{l,j})(y_l - y_j) \Big)$,
and θ is the angle between $(y_i - y_j)$ and v.

Proof. We require,

$$||y_i^{t+1} - y_j^{t+1}||_2^2 > ||y_i^t - y_j^t||_2^2. \tag{81}$$

From Eqs. (76) and (77), $y_i - y_j = (1 + 2\lambda \frac{w_{i,j}}{Z} f_r^{t-SNE}(\zeta_{i,j}))(y_i - y_j) + v$. Putting this value in Eq. (81), we obtain the desired inequality. □

Since t-SNE's condition for repulsion (Eq. 80) resembles that of TriMap, the repulsion behavior will be the same. Thus, t-SNE's repulsive forces can give both attraction and repulsion.

Since, t-SNE's forces are scaled (by $w_{i,j}$ for attraction and $\frac{w_{i,j}}{Z}$ for repulsion), the attractive and repulsive forces are typically lower than that of UMAP. As a result, the algorithm generally uses large values for learning rate (e.g., $\approx 10^3$ in the Open-t-SNE package (Poličar et al., 2024)). Moreover, most t-SNE implementations require an 'early exaggeration' step where the attractive forces are multiplied by a constant value for the first few iterations. This causes points that are supposed to be closer but currently placed far apart to approach each other (inducing far-sightedness). On the other hand, if some points are very close ($\zeta_{i,j} \approx \zeta_{-1}$) but require separation, this early exaggeration trick achieves that as well. Thus, 'early exaggeration' plays a vital role in finding a consistent embedding in t-SNE and is an indispensable feature of the algorithm; especially when initialized randomly. (This trick, when applied throughout the optimization, makes t-SNE embeddings look closer to UMAP embeddings (Böhm et al., 2022).)

J.5 Stochastic Neighbor Embedding (SNE)

SNE (Hinton & Roweis, 2002) is one of the oldest algorithms in this class. This follows the same formula of t-SNE (Eq. 73), but with $q_{i,j} = \exp(-||y_i - y_j||_2^2)$. This results in the attraction shape

$$f_a^{SNE}(\zeta) = -2, \tag{82}$$

that follows Proposition 4.1 (with $0 < \lambda w_{i,j} f_a^{SNE} < -1$) and the repulsion shape

$$f_r^{SNE}(\zeta) = 2\exp(-\zeta^2), \tag{83}$$

that follows proposition J.3 (by replacing the t-SNE symbols with SNE counterparts). The attraction shape is ill-posed and thus mainly relies on the values of learning rate (λ) and the weights ($w_{i,j}$) for contraction resulting in clusters overlapping each other even for small number of samples (called crowding problem (Van der Maaten & Hinton, 2008)), which t-SNE and the subsequent algorithms improve by moving away from the Gaussian kernel to a heavy-tailed one, i.e., Eq. (45).

J.6 Multidimensional Scaling (MDS)

Multidimensional scaling (MDS) (Borg & Groenen, 2007) typically does not use gradient methods, as they often fail to converge to good mappings; instead, it employs stress majorization. Nevertheless, few works discuss gradient methods (Kruskal, 1964; Zheng et al., 2018). We offer a brief treatment for this below. Particularly, Zheng et al. (2018) formulates a successful gradient descent-based MDS algorithm for graph drawing. We start with the cost function (we can ignore the weight $w_{i,j}$ without loss of generality to their approach, and for discussion, we can lump it into the learning rate):

$$\mathcal{L}^{MDS} = \sum_{i,j,i<j} (||y_i - y_j|| - d_{ij})^2. \tag{84}$$

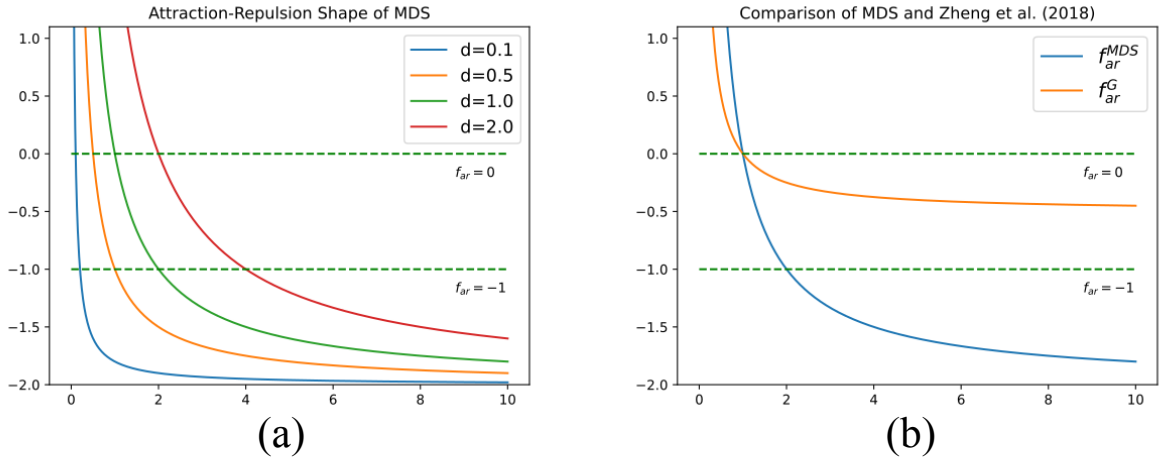


Figure 24: (a) Attraction-repulsion characteristic of MDS using f_{ar}. (b) Comparison of attraction-repulsion characteristic of MDS and the method in (Zheng et al., 2018) for $d = 1.0$.

The loss has a singleton term for each pair and does not have explicit attractive and repulsive terms. The term itself will provide both attraction and repulsion. Fortunately, the loss is separable into individual (i, j) components, allowing us to analyze a single pair. The derivative is given by

$$\frac{\partial}{\partial y_i}(||y_i - y_j|| - d_{ij})^2 = 2\frac{||y_i - y_j|| - d_{ij}}{||y_i - y_j||}(y_i - y_j). \tag{85}$$

Thus, the attraction-repulsion shape of MDS becomes

$$f_{ar}^{MDS}(\zeta) = -2\frac{\zeta - d}{\zeta}, \tag{86}$$

where $\zeta = ||y_i - y_j||_2$. Here, the shape is attractive when $\zeta > d$. In this range, the shape is bounded within $[-2, 0]$ (Fig. 24(a)). However, this is not suitable for convergence as values only within $[-1, 0]$ contract, while the others expand. To make things worse, values lower than -2 cause the points to flip and expand. To work around this, Zheng et al. (2018) uses an ad-hoc formulation inspired by the force-directed graphs, given by

$$f_{ar}^{G}(\zeta) = -\frac{1}{2}\frac{\zeta - d}{\zeta}, \tag{87}$$

which works for an effective learning rate $\lambda \leq 1$. f_{ar}^{G} is bounded within $[-0.5, 0]$ and thus, $\lambda \leq 1$ works. On the other hand, for $\zeta < d$, $f_{ar}^{MDS}(> 0)$ shows repulsive behavior. Overall, this attraction-repulsion interaction works best when the initialization is already close to a desired output. If one keeps optimizing for a pair, it will oscillate around the distance ($\zeta = d$) where $f_{ar}^{G} = 0$ (and consequently, we can have a notion of distance similar to ζ_{-1} of UMAP). No choice of learning rate reduces this distance to zero (in fact, this achieves the objective of MDS). Thus, the clusters are often fuzzy compared to methods like UMAP and PaCMAP (for relevant illustrations, see Lambert et al. (2022), de Bodt et al. (2025), and Kury et al. (2026)). Note that Lambert et al. (2022) converts the MDS loss function to a quartet stress (loss involving four samples) and a relative distance (distance divided by the sum of six distances in the quartet), enabling global regularization and reduced computation than the majorization approach.

K Construction of the High-Dimensional Graph

Stochastic Neighbor Embedding (SNE) Hinton & Roweis (2002) underpins modern dimensionality reduction algorithms. It constructs a high-dimensional graph of the dataset $X = \{x_i \in R^n | i = 1, \ldots, N\}$ by the

following system of equations:

$$w_{ij} = \frac{w_{j|i} + w_{i|j}}{2N}, \tag{88}$$

$$w_{j|i} = \frac{\exp(-||x_i - x_j||_2^2/2\sigma_i^2)}{\sum_{t \neq v} \exp(-||x_t - x_v||_2^2/2\sigma_t^2)}, \tag{89}$$

where σ_i^2 is chosen to match a user-defined value *perplexity*, P, defined as

$$P = 2^{H_i} \tag{90}$$

$$H_i = \sum_{j \neq i} w_{j|i} \log_2 w_{j|i}. \tag{91}$$

On the other hand, UMAP constructs its high-dimensional graph by the following system of equations relying on the k-nearest neighbor (k-NN) algorithm:

$$p_{i|j} = \begin{cases} \exp\left(-\frac{d(x_i,x_j)-\rho_i}{\sigma_i}\right) & \text{if } x_j \in \mathrm{KNN}(x_i, k) \\ 0 & \text{otherwise} \end{cases}, \tag{92}$$

$$\rho_i = \min_{x_j \in \mathrm{KNN}(x_i,k)} d(x_i, x_j), \tag{93}$$

where $\mathrm{KNN}(x_i, k)$ is the set of k-nearest neighbors of the point x_i and σ_i is a scaling parameter such that $\sum_j p_{i|j} = \log_2(k)$. The graph is then symmetrized by a t-conorm:

$$p_{i,j} = p_{i|j} + p_{j|i} - p_{i|j}p_{j|i}. \tag{94}$$

PaCMAP uses just the k-NN graph for the affinities (all equal to 1) with a self-tuning distance measure (Zelnik-Manor & Perona, 2004),

$$d_{i,j}^2 = \frac{||x_i - x_j||^2}{\sigma_i \sigma_j}, \tag{95}$$

where σ_i is the average distance between x_i and its Euclidean nearest fourth to sixth neighbors. The purpose of this is the same as the corresponding σ_i parameters in Eq. (89) and (92), despite all three being defined differently.

Regardless of these choices, the optimization, and by our analysis, the attraction and repulsion shapes are the primary drivers of the low-dimensional embedding.

L Details of Estimating Flip and Expansion, and Average Distance

L.1 Translation-invariance based Flip and Expansion

For each neighboring pair in d dimensions, let $y_i^e \in \mathbb{R}^d$ and $y_j^e \in \mathbb{R}^d$ denote the embeddings at epoch e. We define the relative (pairwise) vector at epoch e as

$$u_{ij}^e = y_j^e - y_i^e. \tag{96}$$

Similarly, at the next epoch we define

$$u_{ij}^{e+1} = y_j^{e+1} - y_i^{e+1}. \tag{97}$$

Because u_{ij}^e and u_{ij}^{e+1} depend only on differences, they are invariant to any global translation applied to all points.

We consider that a *flip* has occurred from epoch e to $e+1$ if the relative ordering of the pair reverses along the original direction, which is detected by a sign change of the inner product:

$$\text{flip}(i, j, e) = \text{sgn}\left(\left\langle u_{ij}^e,\, u_{ij}^{e+1} \right\rangle\right). \tag{98}$$

Intuitively, $\langle u_{ij}^e, u_{ij}^{e+1} \rangle < 0$ indicates that the new relative displacement points opposite to the previous relative displacement, i.e., the two points have swapped their order along the axis defined by u_{ij}^e. We compute whether an expansion has occurred by simply evaluating the statement $d(y_i^{e+1}, y_j^{e+1}) > d(y_i^e, y_j^e)$.

In practice, we ignore degenerate pairs with $\|u_{ij}^e\|_2 \approx 0$ (nearly identical points), since the flip direction is ill-defined in that case.

L.2 Perpendicular Bisector-Based Flip and Expansion

In addition to the above, we use a second method to compute flip and expansion using the perpendicular bisector. For each neighboring pair in 2D, $y_i^e = (y_{i1}^e, y_{i2}^e)$ and $y_j^e = (y_{j1}^e, y_{j2}^e)$, where e is the epoch number, we define a reference normal line (perpendicular bisector) going through the midpoint of connecting line by

$$\text{line}_{ij}^e(y_1, y_2) = Ax + By + C \tag{99}$$

where, A, B, and C are constants computed as:

$$A = y_{i1}^e - y_{j1}^e, \tag{100}$$

$$B = y_{i2}^e - y_{j2}^e, \qquad \text{and} \tag{101}$$

$$C = -\left(A \times \frac{y_{i1}^e + y_{j1}^e}{2} + B \times \frac{(y_{i2}^e + y_{j2}^e)}{2}\right), \tag{102}$$

respectively. Then we compute the initial position using the signum function by

$$S_{ij1}^e = \text{sgn}(\text{line}_{ij}^e(y_{i1}^e, y_{i2}^e)) \quad \text{and} \tag{103}$$

$$S_{ij2}^e = \text{sgn}(\text{line}_{ij}^e(y_{j1}^e, y_{j2}^e)). \tag{104}$$

In the next epoch, the new positions of the points are $y_i^{e+1} = (y_{i1}^{e+1}, y_{i2}^{e+1})$ and $y_j^{e+1} = (y_{j1}^{e+1}, y_{j2}^{e+1})$ respectively. We compute their position from the reference line by

$$S_{ij1}^{e+1} = \text{sgn}(\text{line}_{ij}^e(y_{i1}^{e+1}, y_{i2}^{e+1})) \quad \text{and} \tag{105}$$

$$S_{ij2}^{e+1} = \text{sgn}(\text{line}_{ij}^e(y_{j1}^{e+1}, y_{j2}^{e+1})). \tag{106}$$

We consider, a flip has occurred if $S_{ij1}^e \neq S_{ij1}^{e+1}$ and $S_{ij2}^e \neq S_{ij2}^{e+1}$. We ignore degenerate cases where a point lies exactly on the bisector, i.e., $\text{line}_{ij}^e(y_1, y_2) = 0$. Like before, the expansion is given by evaluating the statement: $d(y_i^{e+1}, y_j^{e+1}) > d(y_i^e, y_j^e)$.

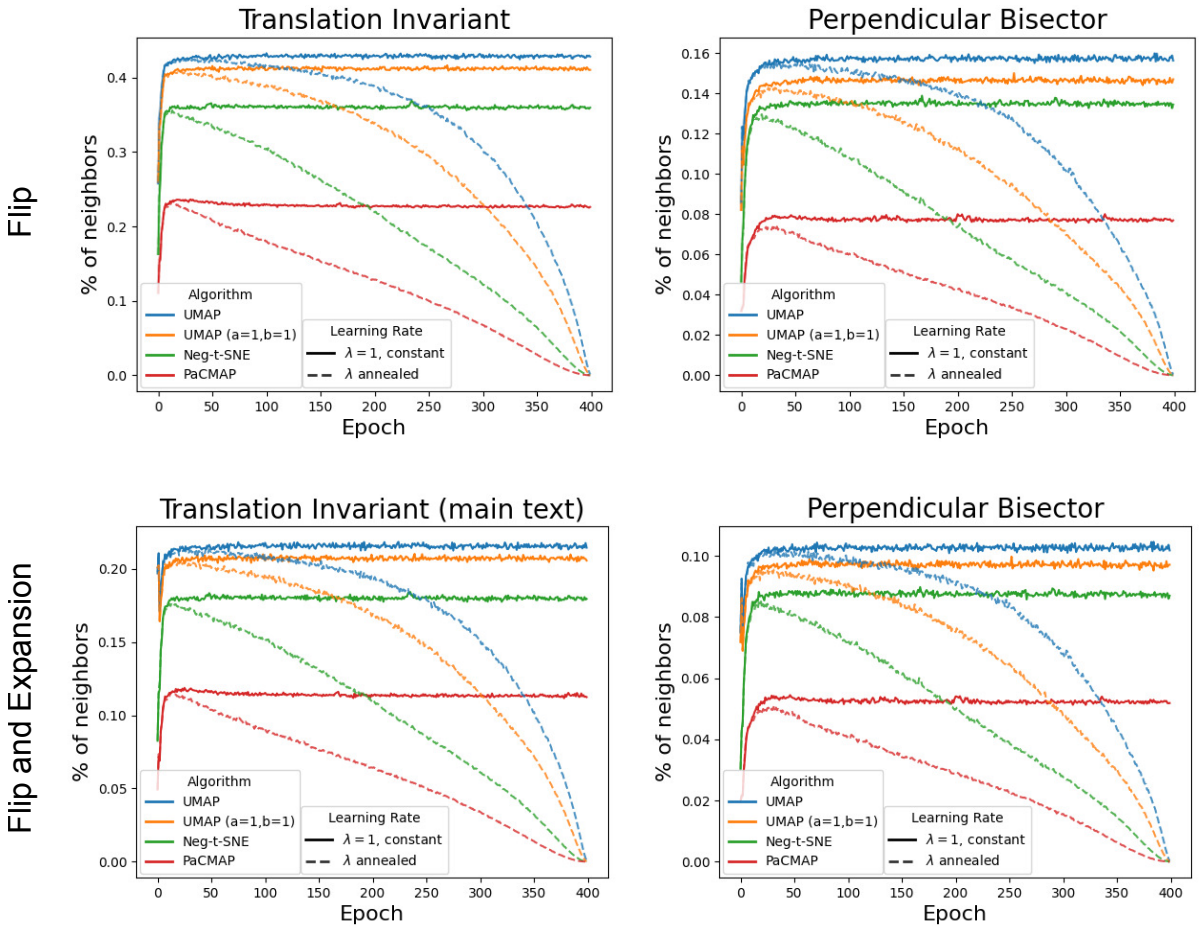


Figure 25: Effect of different methods of computing flip and expansion on the MNIST dataset. Percentage of neighbors that flip computed using (a) translation invariant and (b) perpendicular bisector method. Percentage of neighbors that flip and expand using (a) translation invariant (reproduced form main text) and (d) perpendicular bisector method.

A major detriment of this method is that if both pairs have translated to one side of the reference line, it will miss that point. On the other hand, both methods miss the points that have flipped during attractive update, but due to other interactions, either flipped back, remained unchanged, or shrunk. Notably, PaCMAP prevents flips during attractive update. Thus, we can use PaCMAP as a baseline and compare other methods accordingly.

Figure 25 shows results on counting the number of points that go through flip and expansion. Translation invariant method shows that roughly 42.86% (mean of last 200 epochs) points experience flips in UMAP per epoch when the learning rate is constant, while it is relatively less for other methods (Fig. 25 (a)). For PaCMAP, the value is 22.68%, nearly half of that of UMAP. On the other hand, nearly half of the flipped points expand (25 (b)). For UMAP, only 21.54% points flip and expand, whereas it is 11.34% for PaCMAP (again, nearly half of UMAP).

The perpendicular bisector method (Fig. 25 (c,d)) can identify a lot fewer flips and expansions than the translation invariant method. This is expected as the reference normal line may change from epoch to epoch. However, the trend is similar to that of the translation-invariant method. As we guide attraction shape toward -1 and then to -0.5, the number of flips and expansion decreases.

In all cases, reducing the learning rate reduces the number of flips and expansions.

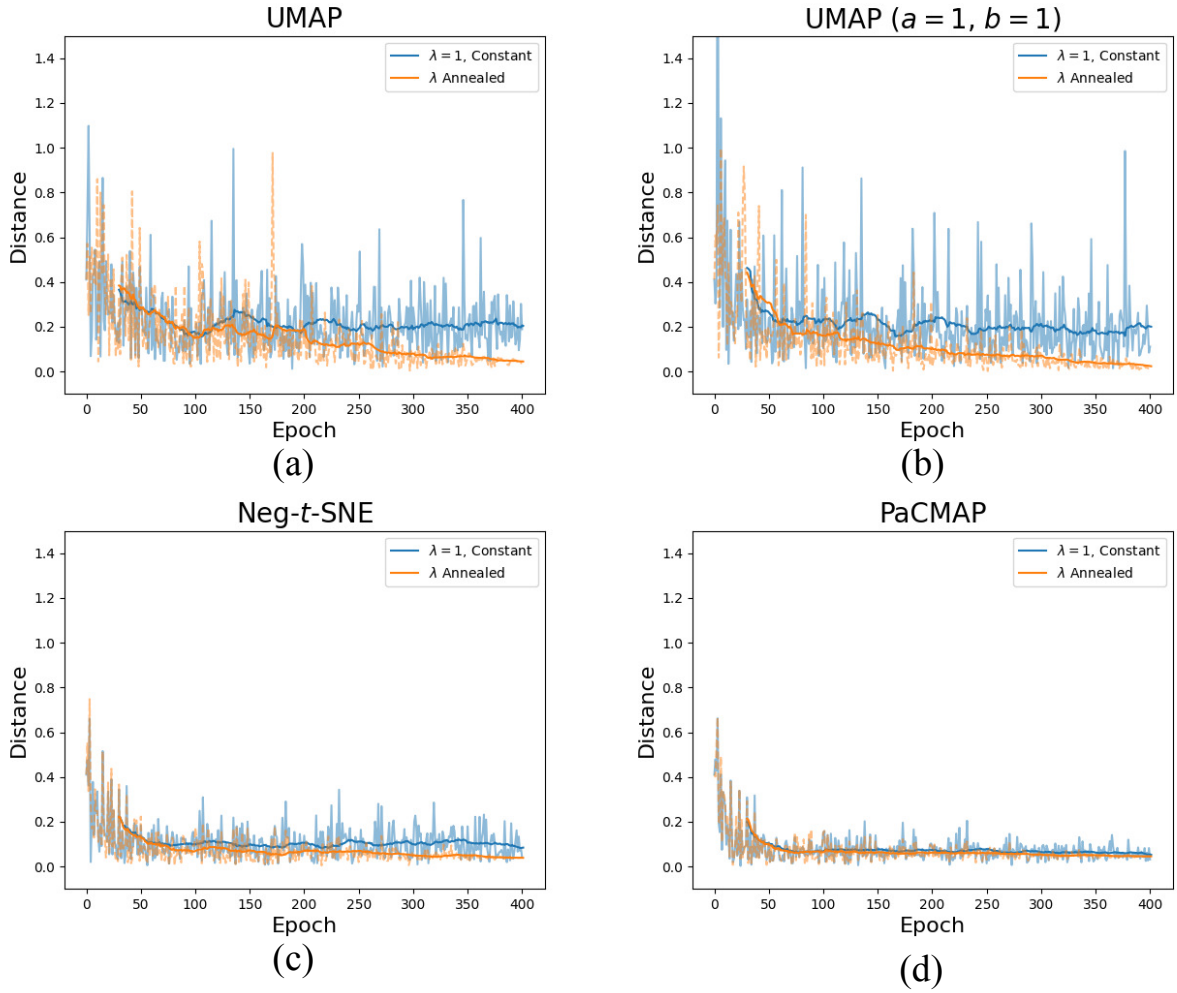


Figure 26: Tracking a neighboring pair across epochs. (a) UMAP, (b) UMAP ($a = 1$, $b = 1$), (c) Neg-t-SNE, and (d) PacMAP.

L.3 Measuring Distance

To compute the average distance, we rescaled each dimension of the embeddings to unit variance. After that, we computed the average distances of the nearest neighbor pairs by

$$d^e_{avg} = \text{mean}\left(\sum_{\substack{i,j;i<j;\\ x_j \in \text{KNN}(x_i,k)}} d(y^e_i, y^e_j)\right). \tag{107}$$

In addition, we tracked one neighboring pair across epochs (Fig. 26). Transparent curves show the distances after each epoch, while the dark curves show the moving average estimate (30 points). UMAP shows the largest gap between the constant and the annealed curve at the end of optimization, whereas Neg-t-SNE shows a much lower gap. PaCMAP shows very little gap throughout, and the curves nearly coincide.

M Effect of Controlled Flips on Embedding Quality

To directly test whether flip events can degrade embedding quality, we introduce a controlled flip intervention during optimization. Specifically, whenever the attraction value f_a lies in $[0, 0.5]$, we replace it by $1 - f_a$ with probability p, while keeping learning-rate annealing enabled. We consider $p \in \{0, 0.25, 0.5, 0.75, 1.0\}$, where $p = 0$ corresponds to default UMAP. In practice, for each eligible attractive update, we draw $u \sim \text{Uniform}(0, 1)$ and apply the flip only when $u < p$. This creates approximately 0%, 25%, 50%, 75%, and 100% additional flips on top of the natural ones that already occur during optimization.

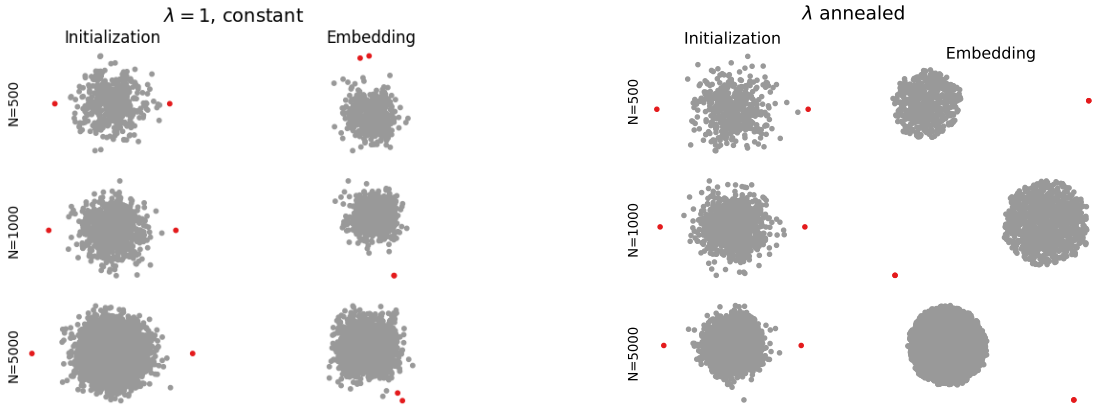
Table 3: Trustworthiness and silhouette score under controlled flip injection on MNIST.

p	0	0.25	0.5	0.75	1.0
Trustworthiness	0.9588	0.9471	0.9382	0.9346	0.9306
Silhouette Score	0.52	0.43	0.41	0.40	0.39

We evaluate this intervention on MNIST using trustworthiness and silhouette score (Table 3). As the flip probability increases, both metrics decrease monotonically: trustworthiness drops from 0.9588 at $p = 0$ to 0.9306 at $p = 1.0$, while the silhouette score drops from 0.52 to 0.39. Since trustworthiness measures preservation of local neighborhoods and silhouette score measures cluster separation, this experiment shows that artificially increasing the number of flips leads to systematically worse embeddings.

This controlled experiment complements our observational analysis of flips during standard optimization. There, flips are measured as a consequence of the attraction shape and learning-rate schedule; here, they are directly injected as an intervention. The resulting monotonic degradation in both neighborhood preservation and clustering quality provides direct evidence that excessive flips are not merely incidental, but can actively harm the learned representation.

N A Possible Pathological Example: Separated-Neighbor Dataset



(a) (b)

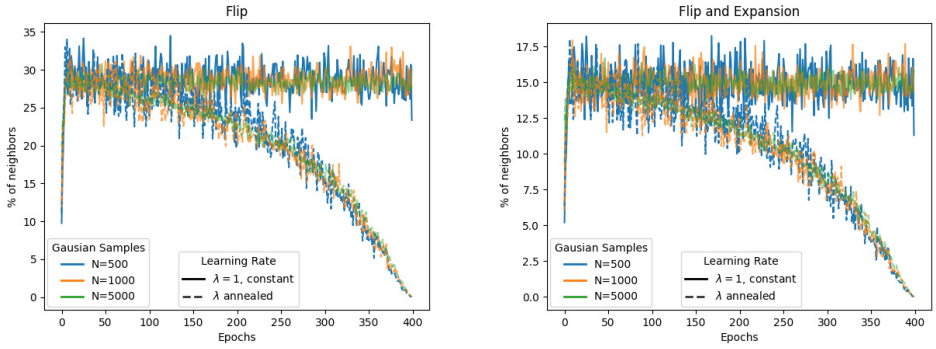
Figure 27: Embeddings of the Separated-Neighbor dataset under different optimization scenarios. (a) Constant learning rate ($\lambda = 1$); (b) annealed learning rate. In each panel, the left column shows the initialization and the right column shows the final embedding. The three rows correspond to $N = 500$, 1000, and 5000, respectively. Red dots denote the free pair, and gray dots denote the Gaussian samples.

One can hypothesize pathological counterexamples in which two neighboring points struggle to contract. One such scenario is when two points are neighbors in the high-dimensional graph but, at initialization, are placed far apart with many repelling points lying between them. One may then argue that, under such a configuration, the neighboring pair might fail to contract. In this section, we investigate precisely this scenario in a 2D embedding setting and examine how UMAP, and more broadly related methods of the same class, behave.

To construct the dataset, we sample N points from a 400-dimensional normal distribution and build the high-dimensional graph according to Eq. (92) with $k = 15$. We then add two *free* vertices that are connected only to each other and to no other vertices in the graph. The embedding is initialized as follows: the original N points are initialized from a 2D normal distribution, while the two free points are placed at $(-1.5, 0)$ and $(1.5, 0)$. This setup simulates a situation in which roughly N repelling points lie between two neighboring points that should contract. We then run the default UMAP optimization for 400 epochs, allowing both the annealed and non-annealed learning-rate settings to reach stable behavior. We refer to this synthetic construction as the *Separated-Neighbor* dataset, since the special neighboring pair must contract across a crowd of repelling points. We perform this experiment for $N = 500$, 1000, and 5000.

Examining the embeddings shows that the free pair indeed contracts, regardless of whether the learning rate is annealed (Fig. 27). Tracking flips and expansions (Fig. 28) reveals a trend similar to that observed on MNIST. With a constant learning rate, the fraction of neighbors that flip is about 28.5%, while the fraction that both flip and expand is about 14.9%. Importantly, these values remain broadly consistent as the dataset size N increases. When the learning rate is annealed, both quantities decrease toward zero. To probe the dynamics more directly, we track the free pair together with a representative pair from the Gaussian cloud throughout optimization (Fig. 29). By around 60 epochs, the free pair contracts and effectively decouples from the Gaussian cloud under both constant and annealed learning rates.

Thus, even in this deliberately adverse configuration, the local contraction behavior identified by our analysis remains visible in the full many-point optimization and does not disappear as N increases over the tested range.



(a) (b)

Figure 28: Flips and expansions in the Separated-Neighbor dataset for $N = 500$, 1000, and 5000. (a) Flips, and (b) combined flip and expansion events. Here, we employed the translation-invariant method.

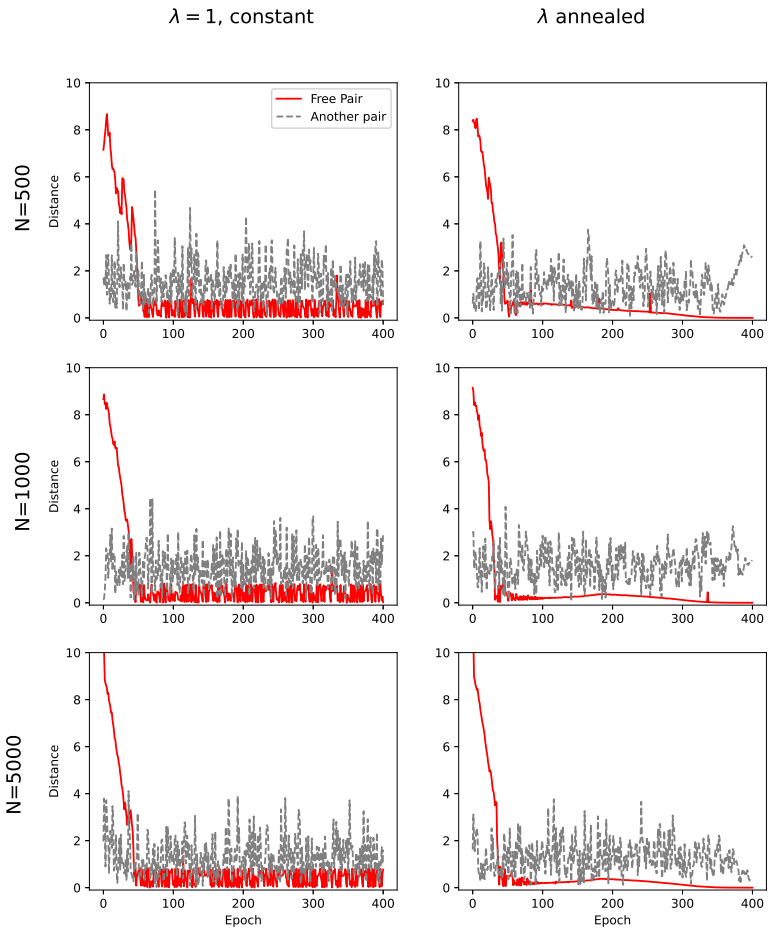


Figure 29: Tracking the embedding-space distances during optimization on the Separated-Neighbor dataset. We show the distance between the free pair and, for comparison, the distance between a representative pair of Gaussian points over the course of optimization.

O Detailed Results for Varying λ_a and λ_b

In this section, we provide detailed results for the experiments in Fig. 1((h,i). We obtained these results, by changing the attraction and repulsion shapes of UMAP to that of the other methods. Figure 30 reproduces the results given in the main text. Figures 31-34 show the individual embeddings for each of the choices of λ_a and λ_r for each methods along with their initialization, a reference embedding when learning rate, λ, is annealed, and when either the λ_a or λ_r set to 0. For PaCMAP, which uses the concept of mid-near pairs, we show additional reference of mid-near pairs included as well. Figures 35- 39 and Figures. 40- 44 provide results on FMNIST and single-cell transcriptomes dataset, respectively.

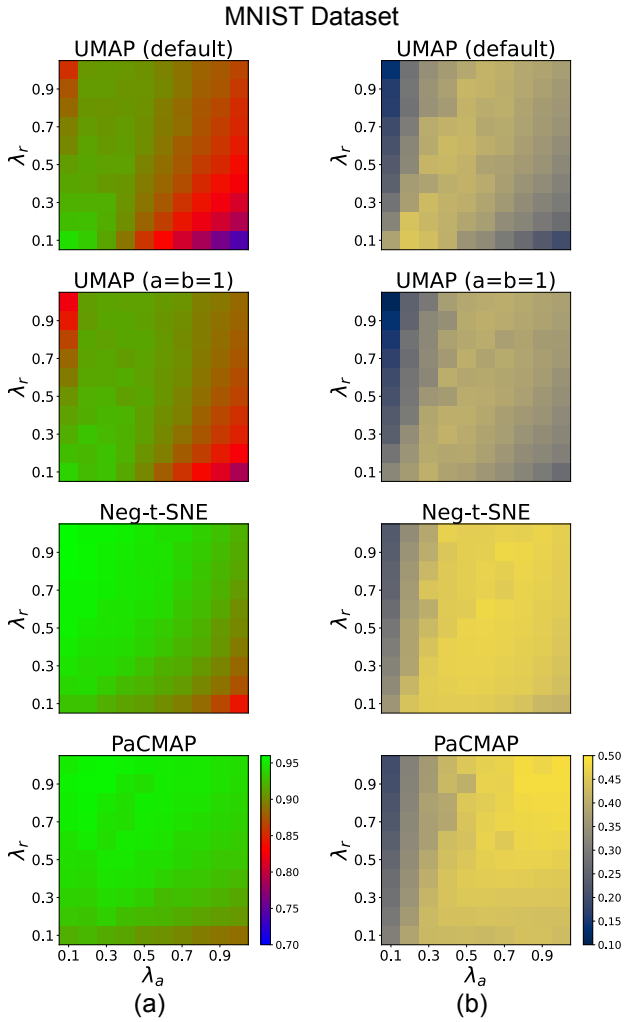


Figure 30: (a) Trustworthiness and (b) Silhouette score of different methods for the MNIST dataset.

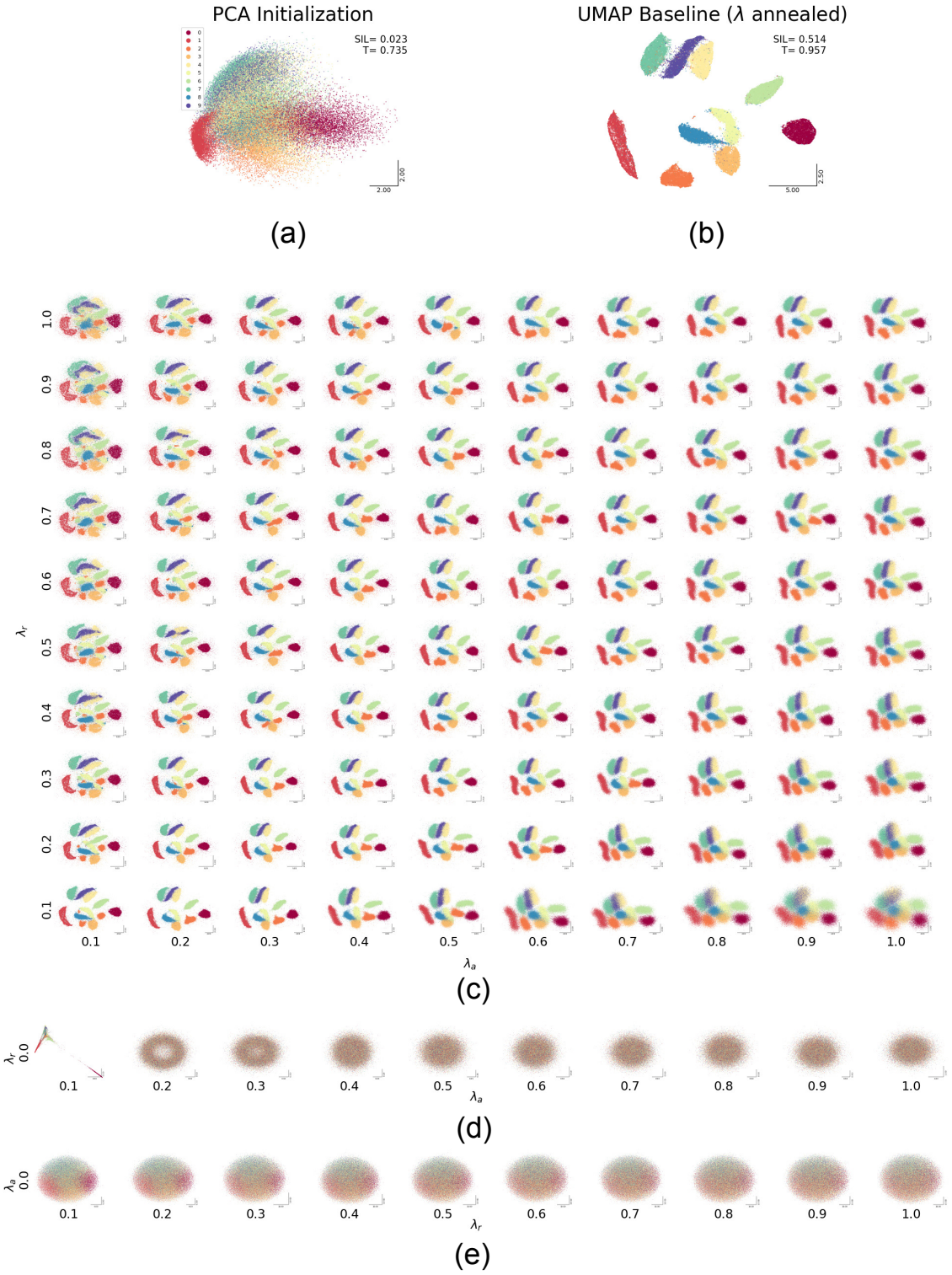


Figure 31: Varying λ_a and λ_r for the MNIST dataset using UMAP's attraction and repulsion shapes. (a) Initialization for the embeddings. (b) Baseline when λ is annealed from 1. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

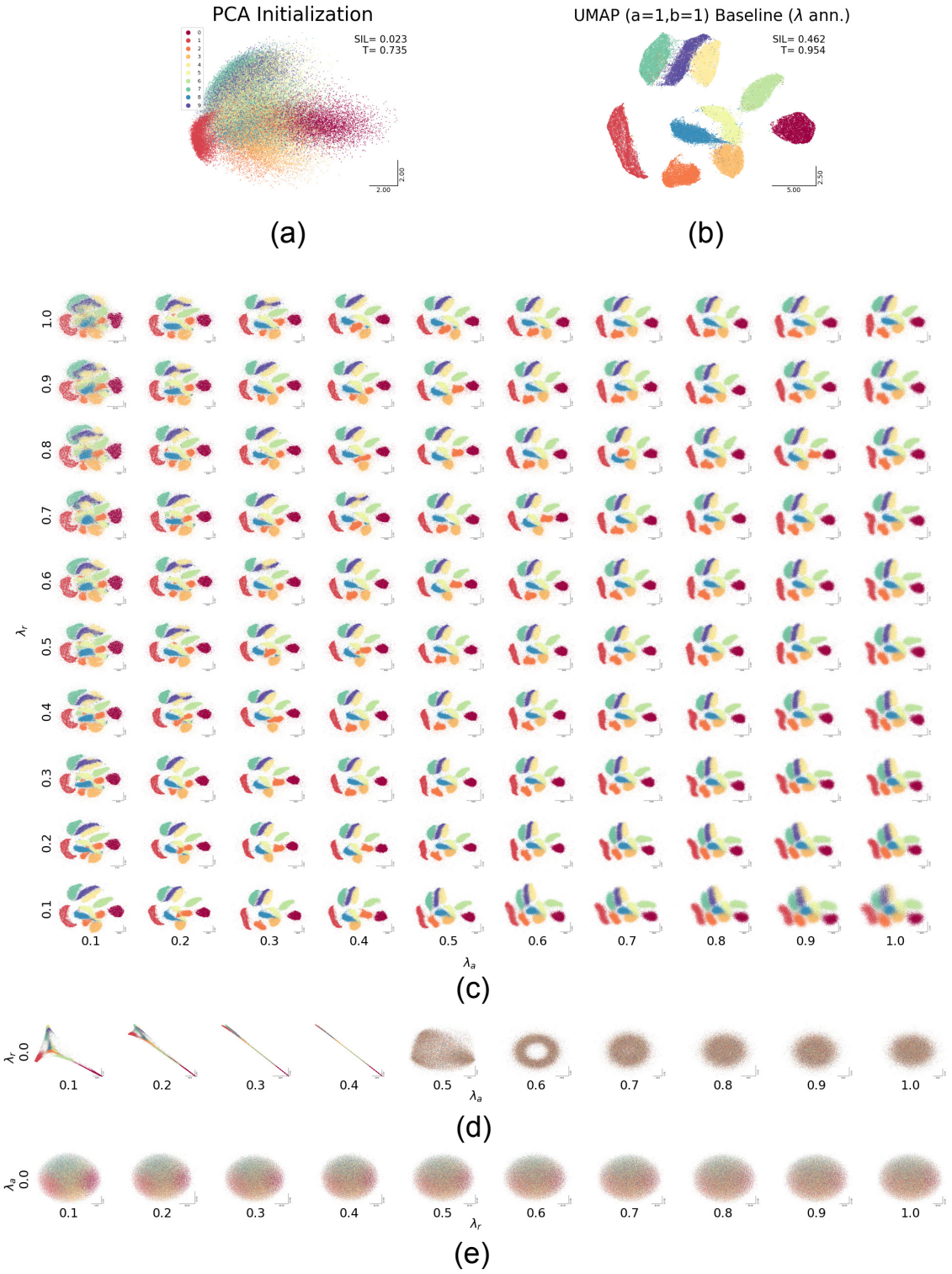


Figure 32: Varying λ_a and λ_r for the MNIST dataset using UMAP's attraction and repulsion shapes (by setting $a = 1$ and $b = 1$). (a) Initialization for the embeddings. (b) Baseline when λ is annealed from 1. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

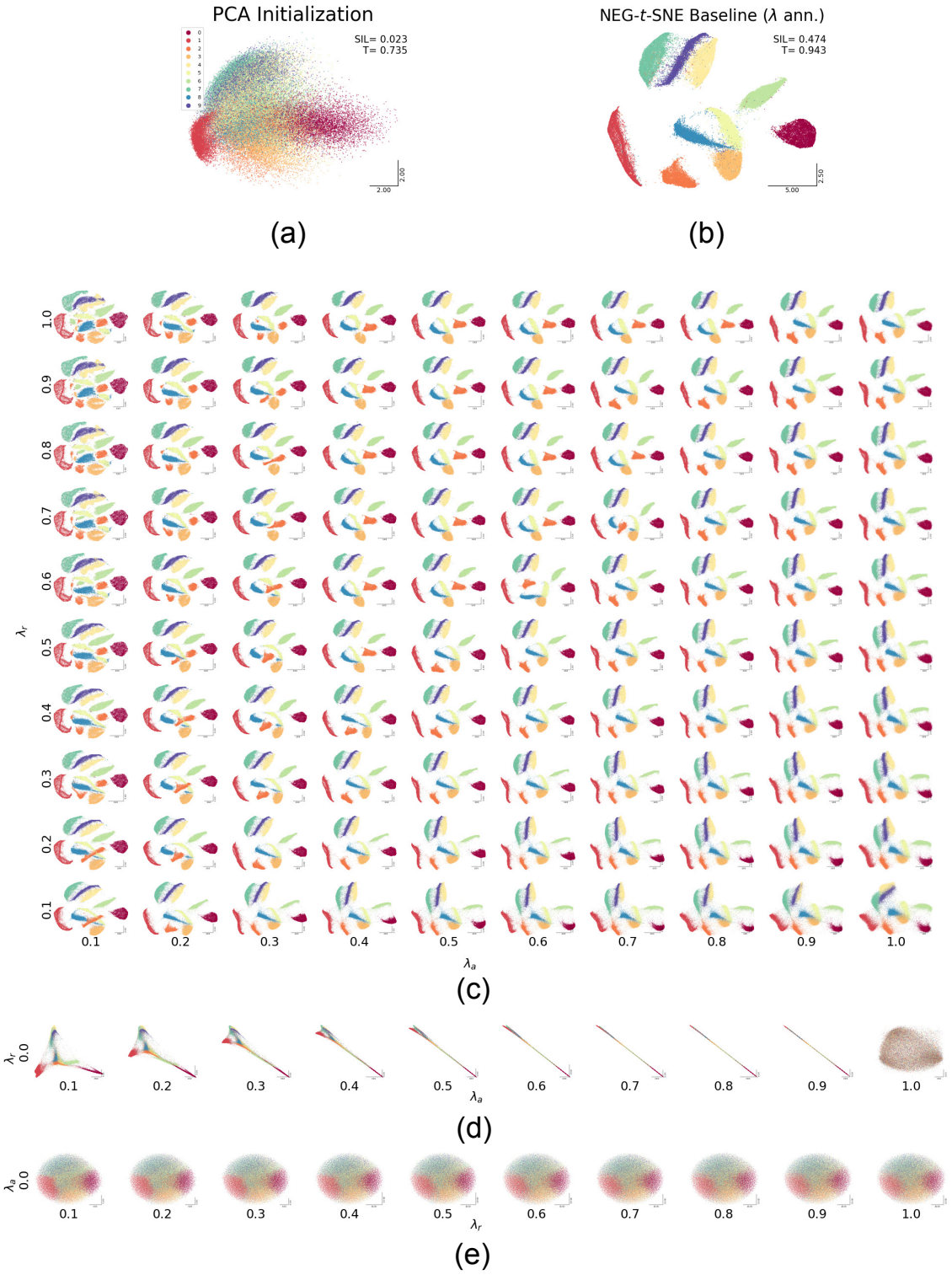


Figure 33: Varying λ_a and λ_r for the MNIST dataset using NEG-t-SNE's attraction and repulsion shapes. (a) Initialization for the embeddings. (b) Baseline when λ is annealed from 1. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

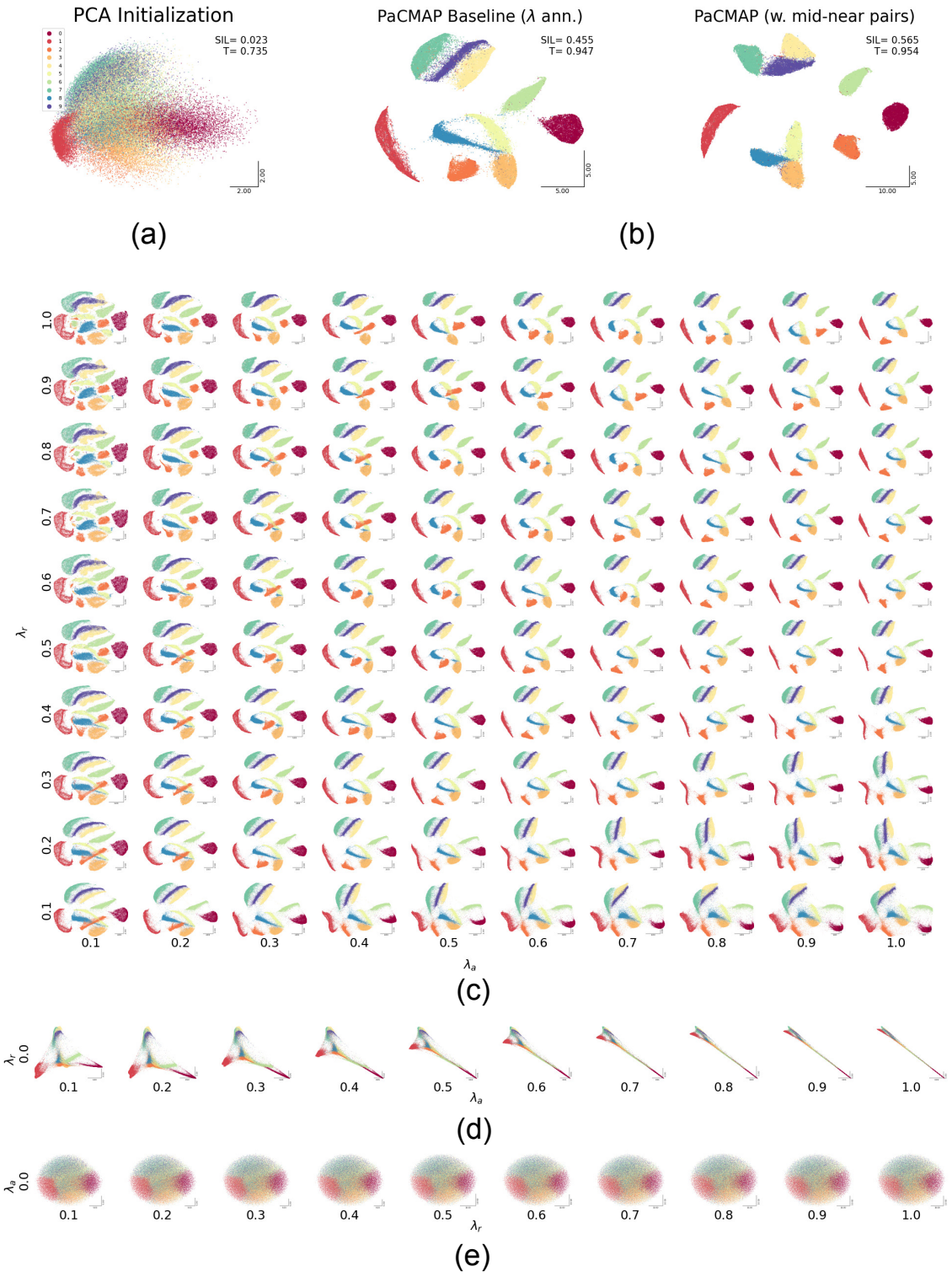


Figure 34: Varying λ_a and λ_r for the MNIST dataset using PaCMAP's attraction and repulsion shapes. (a) Initialization for the embeddings. (b) Left: Baseline when λ is annealed from 1 (mid-near points are excluded to observe the interaction of attraction-repulsion alone), right: when mid-near points are considered. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

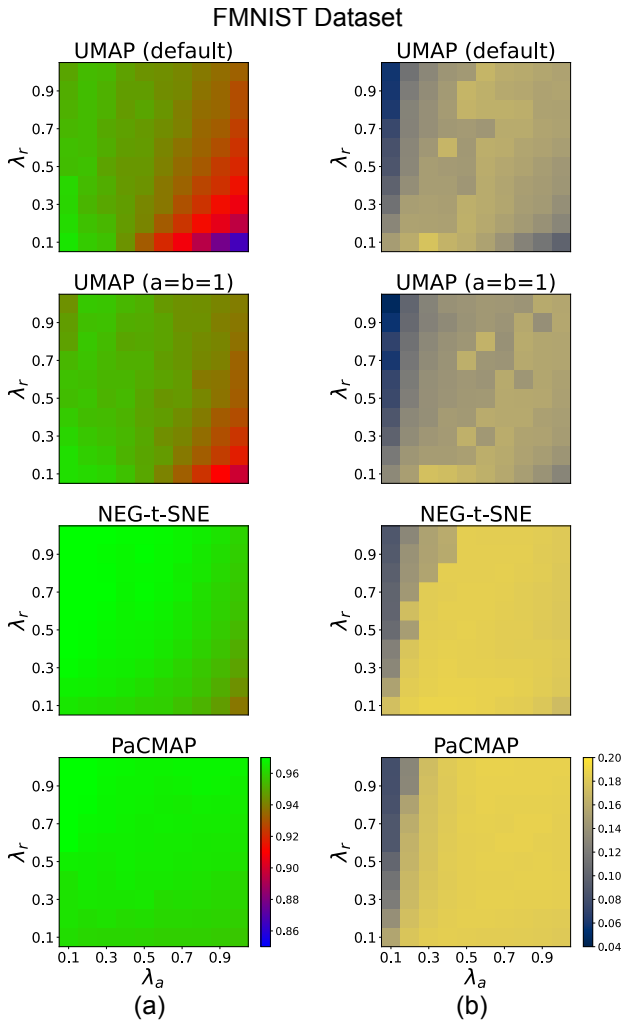


Figure 35: (a) Trustworthiness and (b) Silhouette score of different methods for the FMNIST dataset.

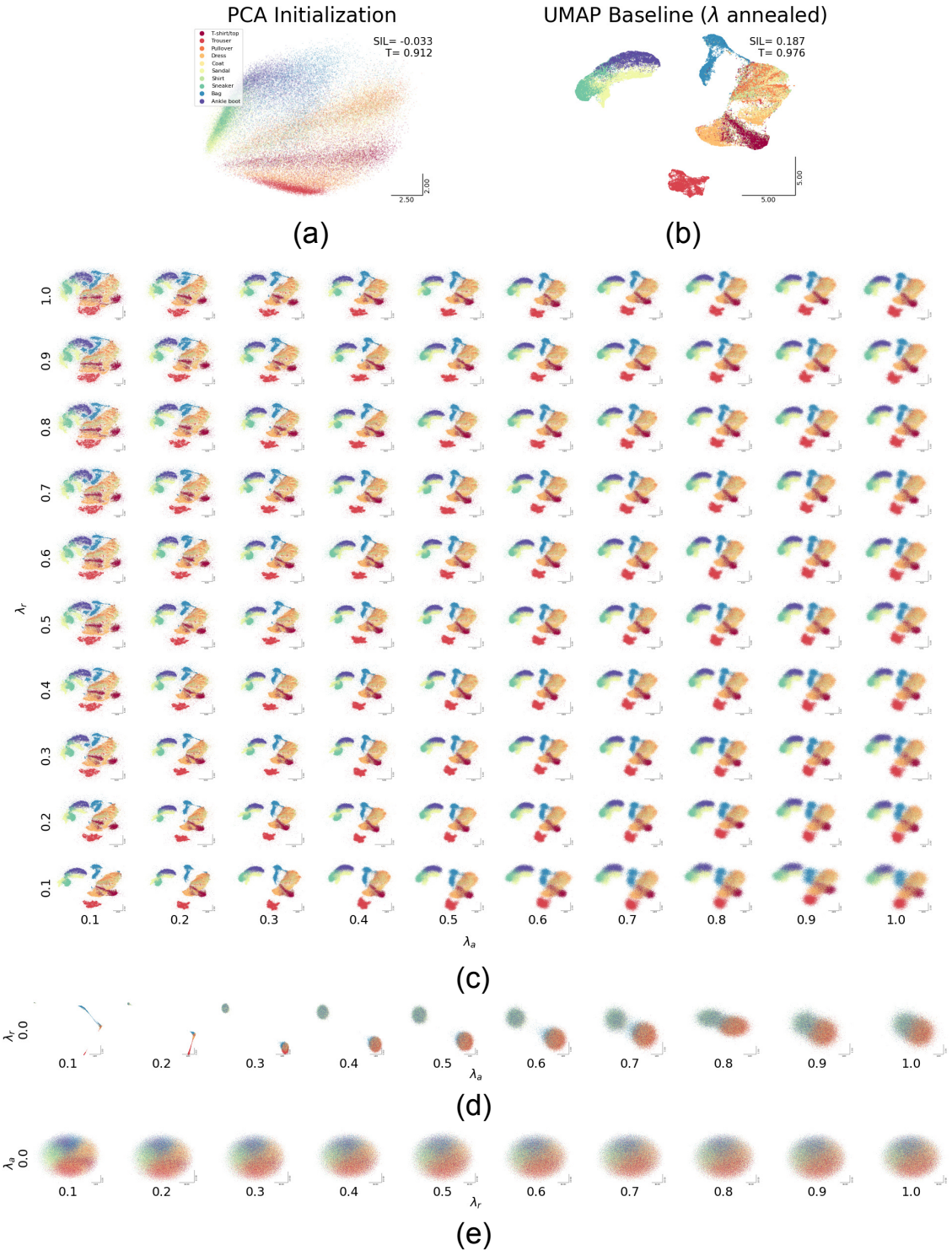


Figure 36: Varying λ_a and λ_r for the FMNIST dataset using UMAP's attraction and repulsion shapes. (a) Initialization for the embeddings. (b) Baseline when λ is annealed from 1. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

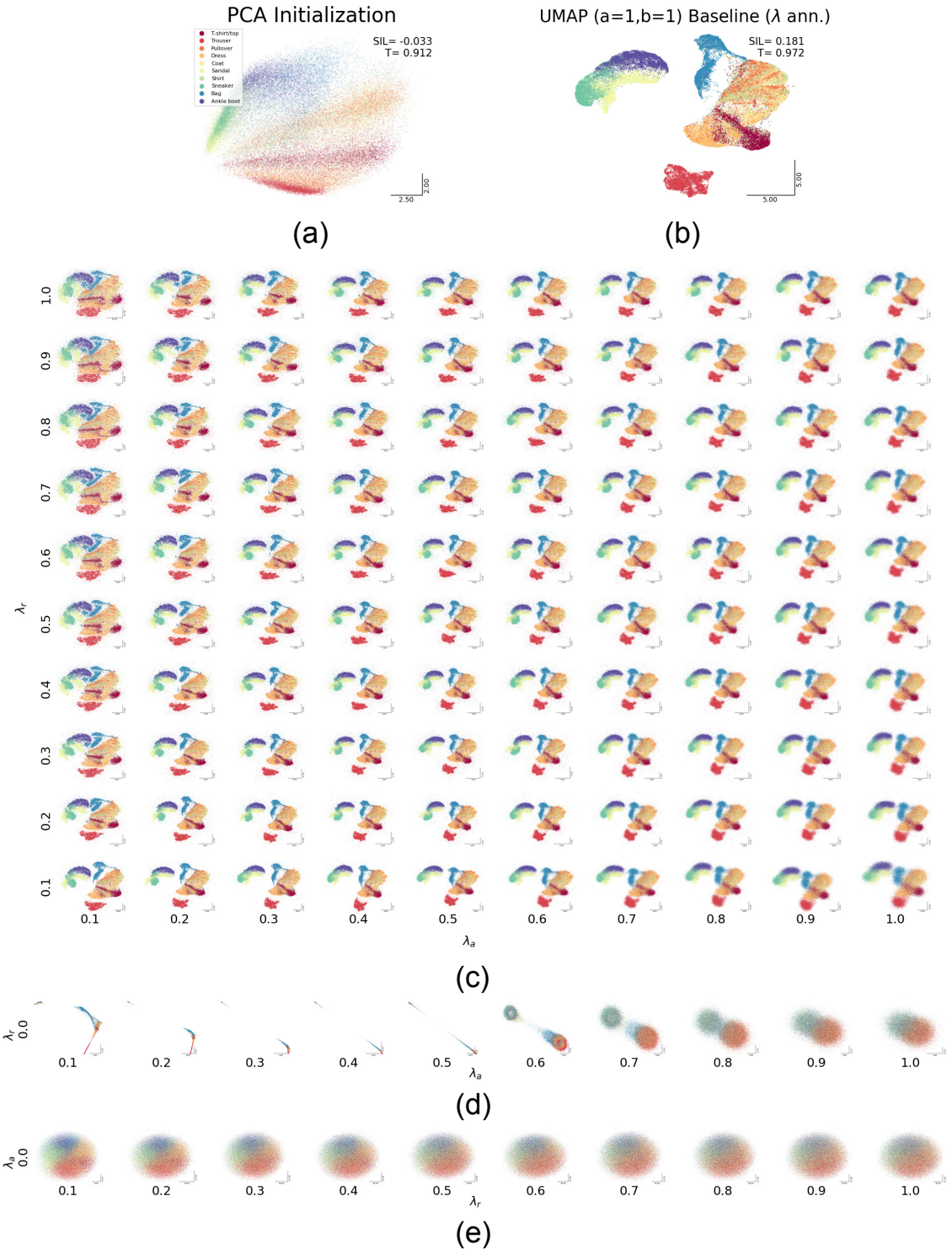


Figure 37: Varying λ_a and λ_r for the FMNIST dataset using UMAP's attraction and repulsion shapes (by setting $a = 1$ and $b = 1$). (a) Initialization for the embeddings. (b) Baseline when λ is annealed from 1. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

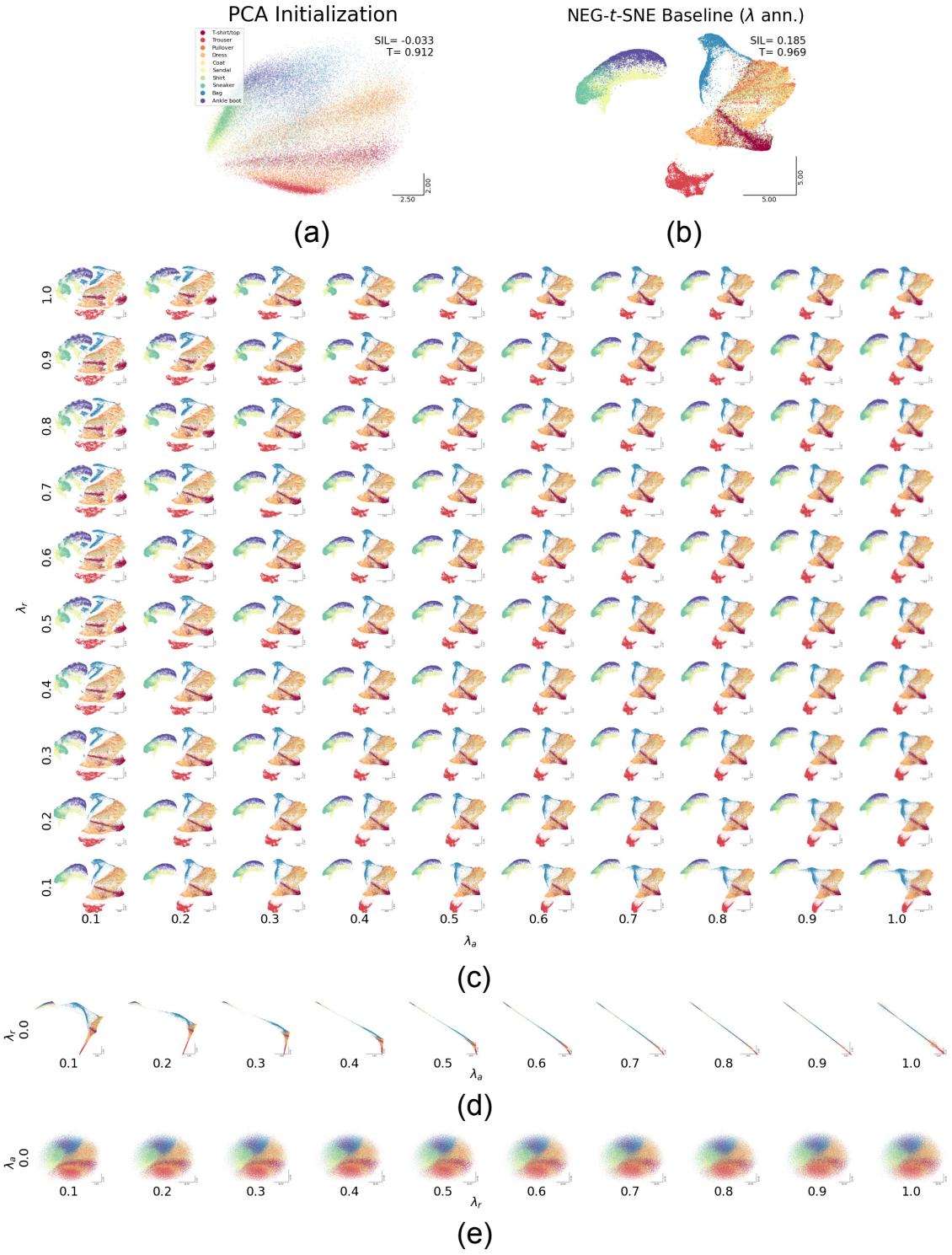


Figure 38: Varying λ_a and λ_r for the FMNIST dataset using NEG-t-SNE's attraction and repulsion shapes. (a) Initialization for the embeddings. (b) Baseline when λ is annealed from 1. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

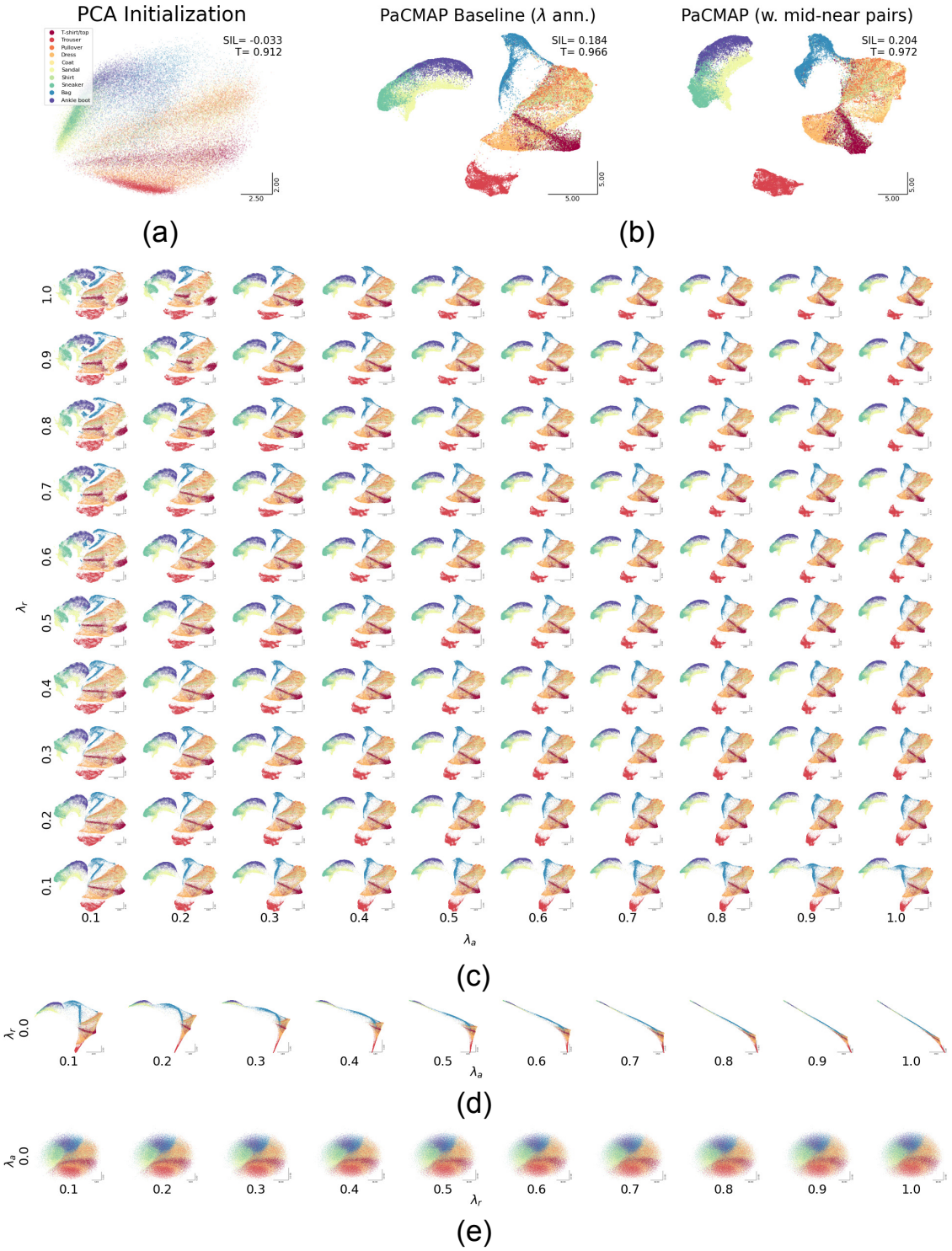


Figure 39: Varying λ_a and λ_r for the FMNIST dataset using PaCMAP's attraction and repulsion shapes. (a) Initialization for the embeddings. (b) Left: Baseline when λ is annealed from 1 (mid-near points are excluded to observe the interaction of attraction-repulsion alone), right: when mid-near points are considered. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

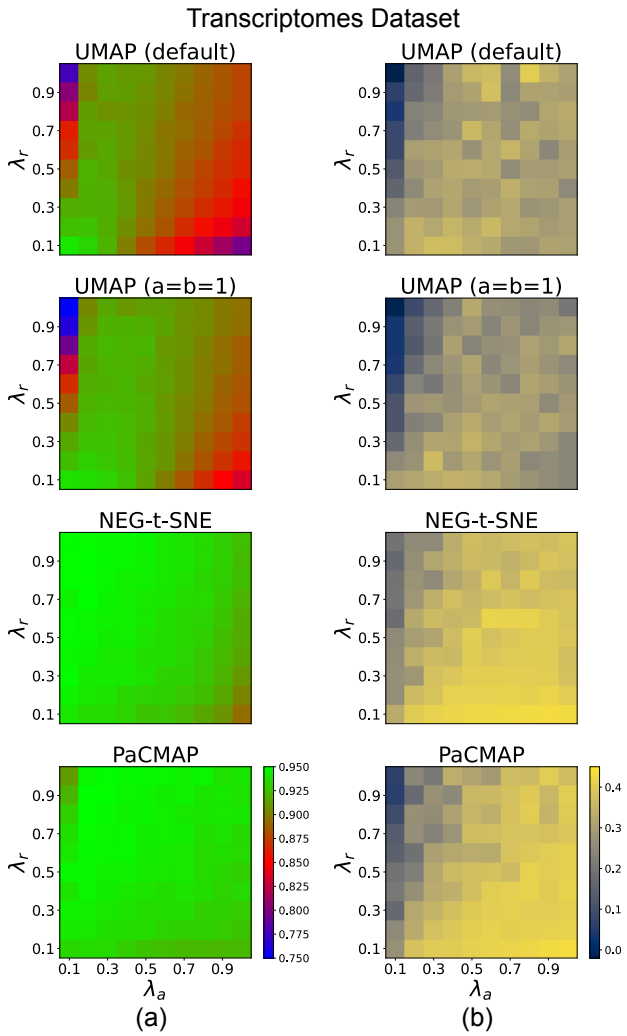


Figure 40: (a) Trustworthiness and (b) Silhouette score of different methods for the Single-cell transcriptomes dataset.

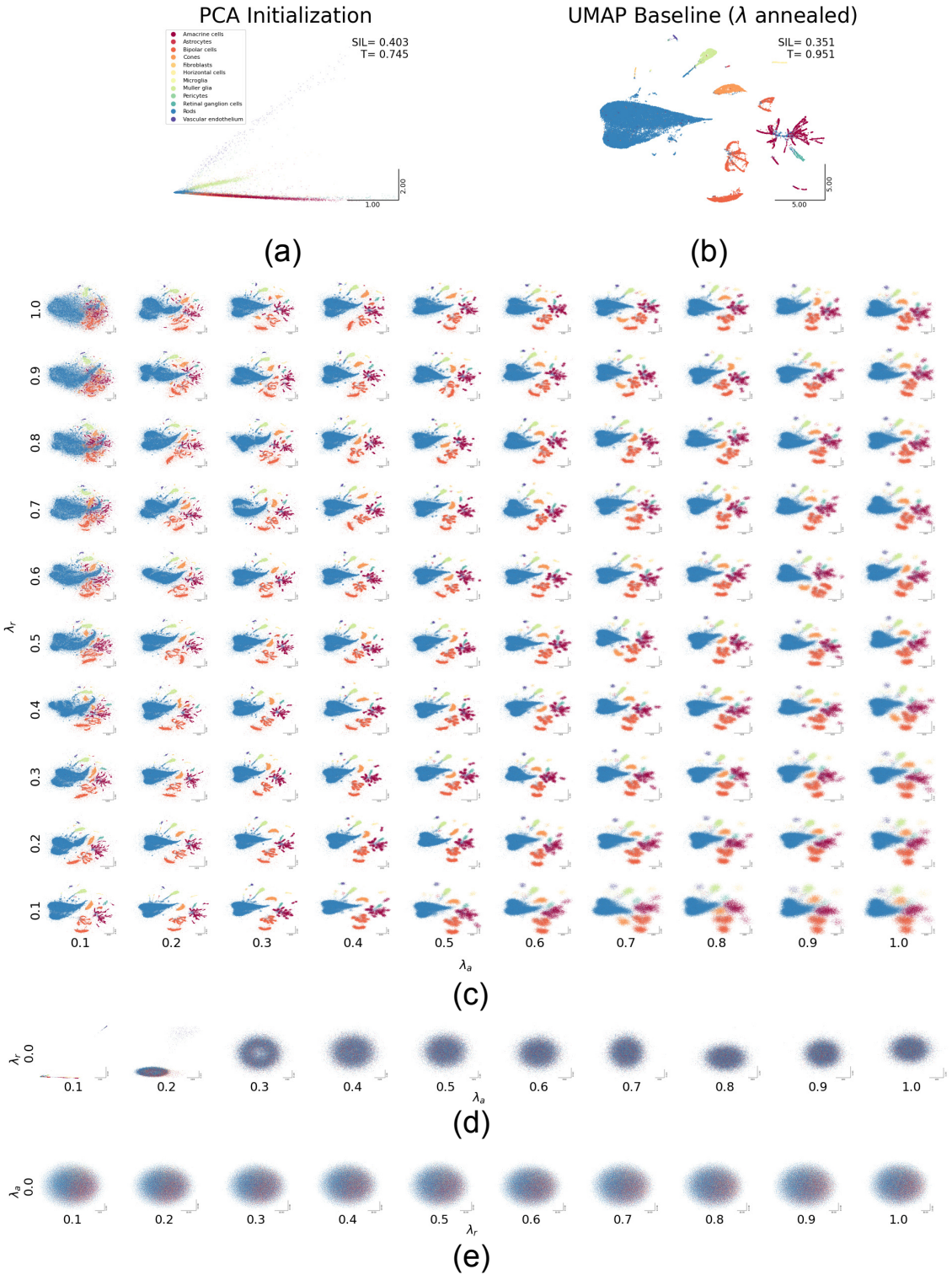


Figure 41: Varying λ_a and λ_r for the single-cell transcriptomes dataset using UMAP's attraction and repulsion shapes. (a) Initialization for the embeddings. (b) Baseline when λ is annealed from 1. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

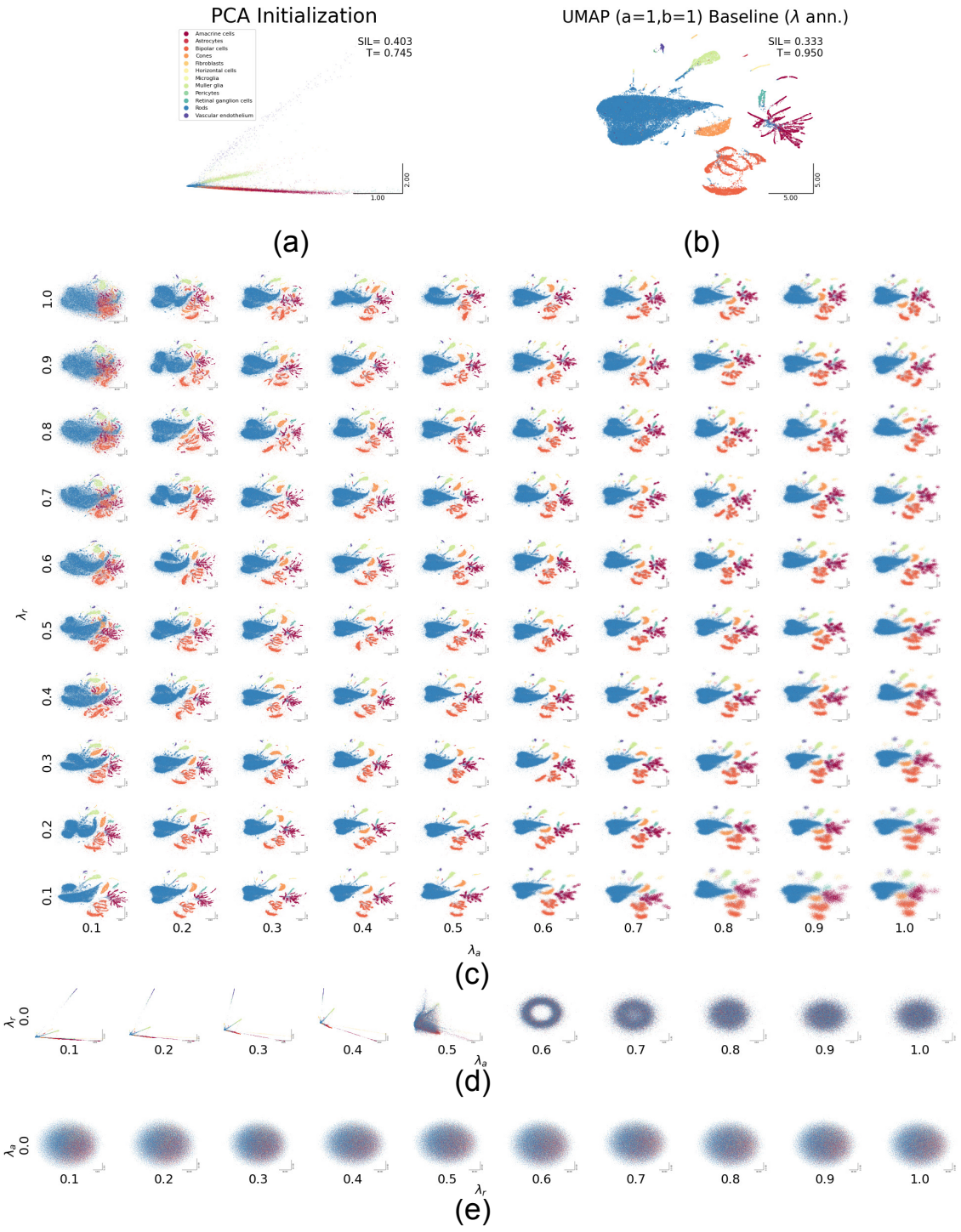


Figure 42: Varying λ_a and λ_r for the single-cell transcriptomes dataset using UMAP's attraction and repulsion shapes (by setting $a = 1$ and $b = 1$). (a) Initialization for the embeddings. (b) Baseline when λ is annealed from 1. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), repulsive force alone cannot produce any clusters.

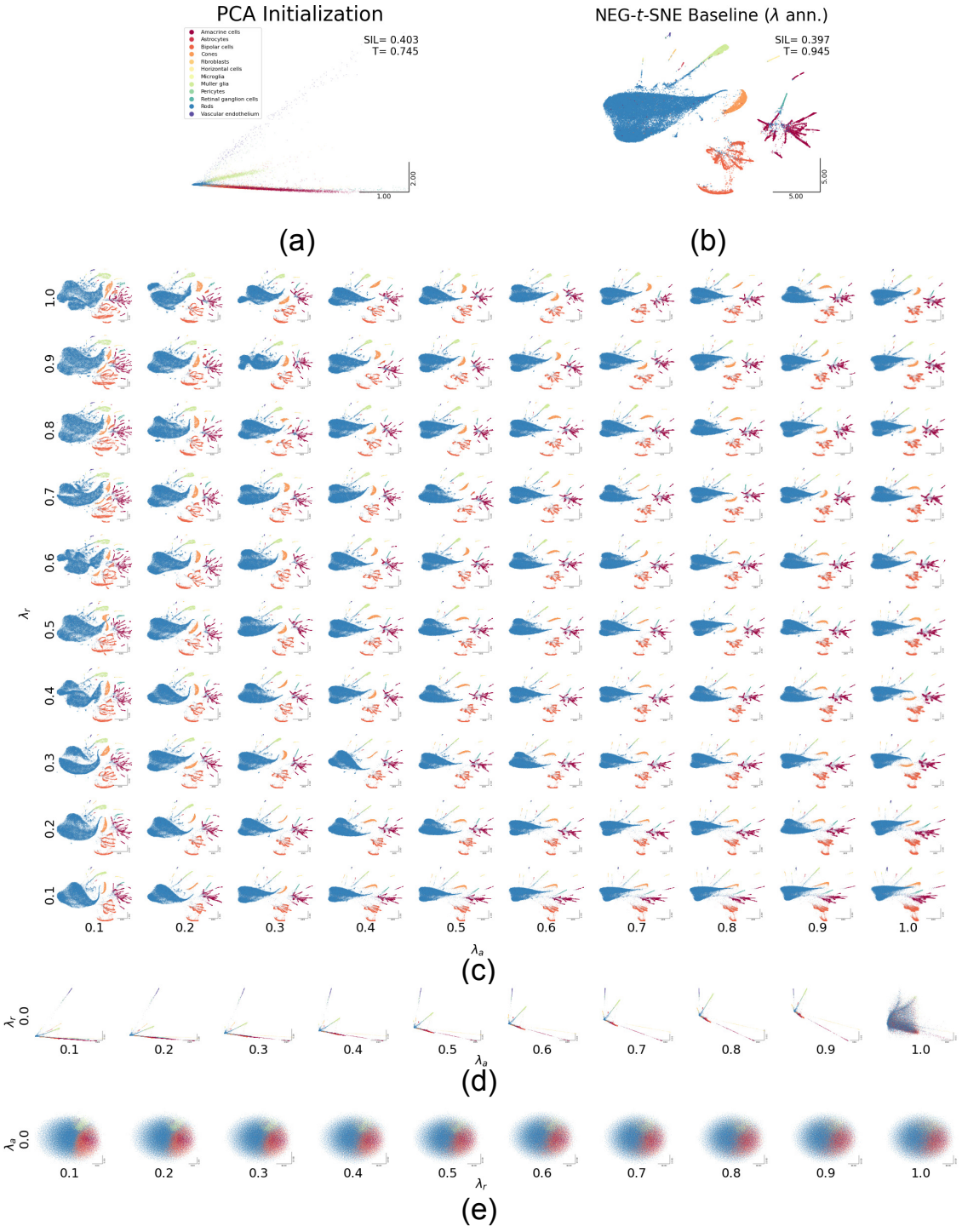


Figure 43: Varying λ_a and λ_r for the single-cell transcriptomes dataset using NEG-t-SNE's attraction and repulsion shapes. (a) Initialization for the embeddings. (b) Baseline when λ is annealed from 1. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce distinct clusters. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

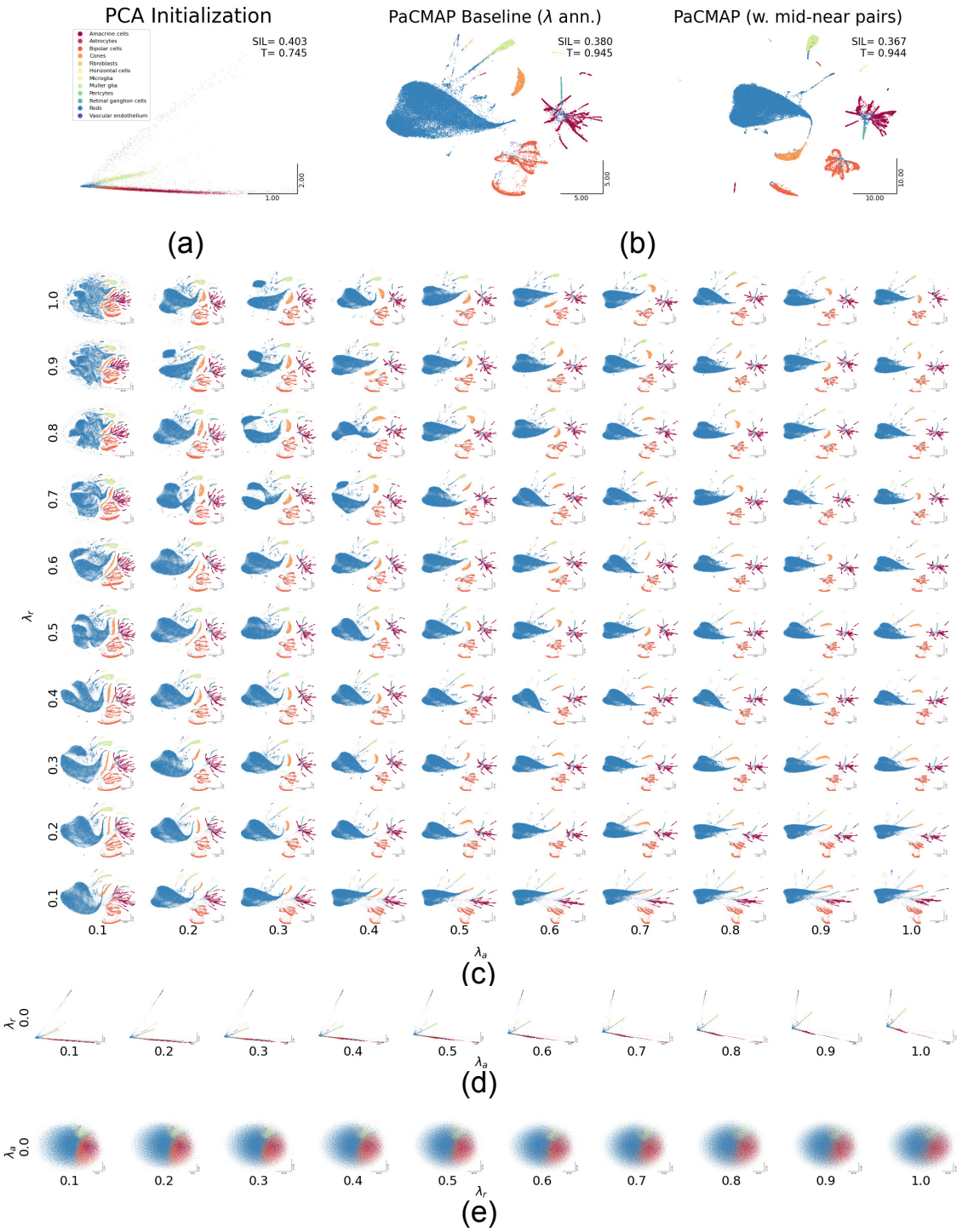


Figure 44: Varying λ_a and λ_r for the single-cell transcriptomes dataset using PaCMAP's attraction and repulsion shapes. (a) Initialization for the embeddings. (b) Left: Baseline when λ is annealed from 1 (mid-near points are excluded to observe the interaction of attraction-repulsion alone), right: when mid-near points are considered. (c) The embeddings when λ_a and λ_r vary (without any annealing). (d) When λ_r is set to 0 (no repulsion), the attractive force alone cannot produce any cluster. (e) Similarly, when λ_a is set to 0 (no attraction), the repulsive force alone cannot produce any clusters.

P Implementation Details

For analysis, we implemented our own UMAP algorithm. We used `numba` (Lam et al., 2015) to compute an exact nearest neighbor graph (instead of an approximate one) with $k = 15$ and `scikit-learn`'s (Pedregosa et al., 2011) implementation of the PCA algorithm for PCA initialization. We used this to produce and quantify the embeddings in Figs. 1, 4, 5, 6, 8, 9, 10, 11, 12, and 13. We also used the same implementation when we changed the attraction and the repulsion shapes to those of the alternative methods (Figs. 30- 44, unless otherwise stated). The trustworthiness and silhouette scores were computed using the corresponding function from the `scikit-learn` package.

The mappings shown in Figs. 4, 5, 10, 11, 12, 13, and 22 are rotated to a reference embedding ((a) for each respective Figures). To achieve this, we performed Procrustes alignment of the embeddings by normalizing them (zero mean and unit norm) and then using SciPy's (Virtanen et al., 2020) `orthogonal_procrustes` method to extract rotation and scaling parameters.

To compare with Neg-t-SNE in Fig. 18, we used the original implementation of the contrastive embedding framework for both the UMAP and the Neg-t-SNE algorithms (available at https://github.com/berenslab/contrastive-ne).

PaCMAP and LocalMAP embeddings in Figs. 22, 23, 34 (b) (right), 39 (b) (right), and 44 (b) (right) were obtained using the official PaCMAP package (available at https://github.com/YingfanWang/PaCMAP).