VISUALIZING MANIFOLD EVOLUTION WITH IRIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Manifold Learning algorithms, such as t-SNE and UMAP, are widely used to elucidate structure in high-dimensional data, yet cannot incorporate temporal information. Though variants like Dynamic t-SNE and Aligned UMAP can jointly optimize discrete time slices, they require each item to be tracked across these slices, precluding their use for event-stream data, like news items, papers in citation networks, or cellular-level gene expression. Here, we present IRIS, a new Manifold Learning algorithm that reveals the evolving structure of high-dimensional data, even when each item only has a single observation. The core advance of IRIS is to reparameterize a Cartesian objective through the complex plane, defining radius a monotonic function of time and angle as the optimized variable. This creates an intuitive polar structure in which the manifold grows outward from the origin. Using synthetic data with known topology, we show that IRIS excels at recovering time-dependent branching, outperforming a range of baselines. Then, for real data of multiple modalities, we demonstrate that IRIS reveals temporal structure that existing algorithms cannot, while keeping classes more contiguous. IRIS unlocks temporal Manifold Learning for event-stream data, allowing exploration of population-level phenomena—like the appearance, disappearance, splitting, or merging of submanifolds—in a single, unified layout, and with no requirement to track items across discrete time slices.

1 INTRODUCTION

Manifold Learning algorithms, such as t-distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten & Hinton, 2008) and Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018), have become indispensable tools for exploring many types of high-dimensional data. In biology, learned low-dimensional layouts help to understand cellular progression and states based on high-dimensional gene expression levels measured with single-cell RNA-sequencing (scRNA-seq) (Becht et al., 2019; Kobak & Berens, 2019; Moon et al., 2018). In Text Mining and Natural Language Processing, Manifold Learning can cluster documents by semantic content, helping to reveal interactions of this content and metadata (González-Márquez et al., 2024; Tang et al., 2016). In Machine Learning, Manifold Learning is widely used to characterize intrinsic structure, identify outliers, and interpret and troubleshoot models by visualizing latent activation spaces (Smilkov et al., 2016; Carter et al., 2019).

Many high-dimensional data, however, have manifolds defined in part over the dimension of time. For example, cellular differentiation in a developing embryo will exhibit temporal branching structures, or news articles on similar topics may narrow, widen, appear, disappear, and reappear as events spur coverage. As standard Manifold Learning algorithms cannot incorporate time into their layout coordinates, this structure is "flattened," providing only a retrospective summary.

If data are *longitudinal*—that is, each item is observed in each of some fixed number of discrete time slices—variants such as Aligned UMAP (Dadu et al., 2023) and Dynamic t-SNE (Rauber et al., 2016) can be used to stabilize embedded coordinates of these items across multiple low-dimensional layouts (Fig. 1, left). However, this approach creates n separate plots for n time steps, restricting it to a small number of time points and making interpretation difficult. Perhaps more importantly, such variants cannot handle *event-stream* data, in which each item only has a single observation, associated with a continuous timestamp. This excludes entire categories of real-world datasets, like citation networks, news articles, or temporal scRNA-seq (which destroys each cell it queries).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these problems, we instead propose structuring a *single* low-dimensional layout both chronologically *and* by high-dimensional relationships (Fig. 1, right). This allows learning a unified temporal manifold, even when each item only has a single observation with an arbitrary timestamp. To achieve this, we propose IRIS, a new Manifold Learning algorithm that reparameterizes a Cartesian objective through a the complex plane, allowing radius to be defined by a monotonic function of time and angle to be optimized according to high-dimensional relationships (Fig. 2). The result is a disc-shaped layout where time radiates outward in concentric rings as the manifold evolves. This lets IRIS go beyond visually aligning snapshots of manifolds, to learning a population-level temporal manifold.

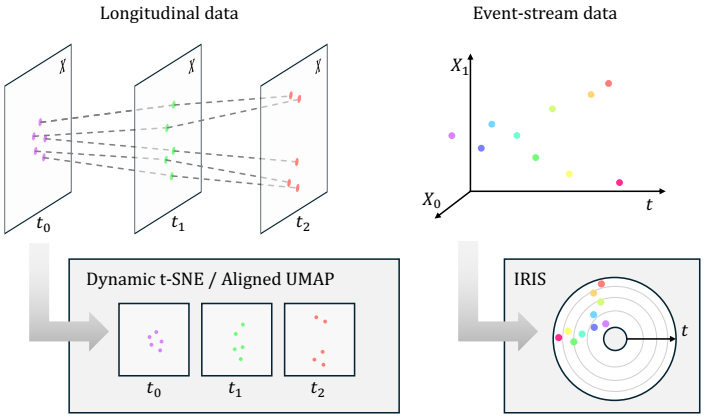

Figure 1: Longitudinal vs. event-stream Manifold Learning.

As event-stream Manifold Learning is not well-studied, we propose strong baselines based on both (1) supervised Manifold Learning and (2) including time as an auxiliary spatial variable for unsupervised methods. Using synthetic data to ensure a ground truth branching structure, we first show that IRIS recovers topology more accurately than these methods. We then explore real datasets of three modalities, finding that IRIS illuminates topology that would otherwise remain obscured, while retain strong metrics for class purity and unity. IRIS unlocks temporal Manifold Learning for event-stream data, removing the need for linked items, and providing unified, intuitive views of the evolution of high-dimensional data. We provide an open-source implementation at [ANONYMIZED].

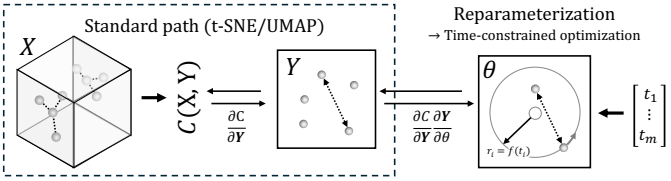

Figure 2: IRIS achieves time-structuring via reparameterization.

2 Background

Manifold Learning assumes there is some structure (the manifold) in the data that makes its local intrinsic dimensionality lower than its observational dimensionality, which is typically the case for data generated by some natural phenomenon. Owing to the myriad cases where high-dimensional data need to be visualized, Manifold Learning is a well-studied problem, with a wide variety of algorithms, including Isomap, Locally Linear Embedding (LLE), and Multi-Dimensional Scaling (MDS) (see the review by Meilă & Zhang (2024) for a more thorough treatment). Here, however, we will focus on several methods that are widely used in practice: t-SNE, UMAP, and PHATE.

t-Distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten & Hinton, 2008) is variant of Stochastic Neighbor Embedding (SNE) (Hinton & Roweis, 2002), which models the chance of each point being a given point's neighbor as a Gaussian in both the original and target spaces and optimizes point positions in the target space by minimizing the Kullback-Leibler divergence between these distributions for all points. In t-SNE, a student's t distribution is used in place of the Gaussian in the target space. This addresses the 'crowding problem' by increasing the influence of more distant points, owing to the longer tail of the student's t distribution.

Uniform Manifold Approximation and Projection (UMAP) McInnes et al. (2018) instead seeks to minimize the difference between the fuzzy simplicial sets of the high- and low-dimensional spaces,

ensuring that local neighborhoods remain similar. Algorithmically, UMAP first constructs a k-neighbors graph from high-dimensional distances, then performs a number of iterations in which attractive forces are applied to pairs of vertices with edges and repulsive forces are applied to negatively sampled pairs (with no edges).

Potential of Heat diffusion for Affinity-based Transition Embedding (PHATE) (Moon et al., 2019) computes diffusion probabilities from affinities to reduce noise and learn long-distance relationships along the manifold, transforms them with an informational distance, and embeds with Multi-Dimensional Scaling (MDS). PHATE was designed to better handle the common branching trajectories of single-cell RNA-sequencing data without shattering them into compact clusters, as t-SNE does for this type of data.

Temporal Manifold Learning is less well-studied. As we discuss in §1, Dynamic t-SNE (Rauber et al., 2016) and Aligned UMAP (Dadu et al., 2023) require the same items at each time step, making them inapplicable to event-stream data. Though simply coloring plots by time or plotting layout coordinates versus time are sometimes used to convey high-level trends for event-stream data (González-Márquez et al., 2024; Qiu et al., 2024; Rhodes et al., 2026), these strategies cannot represent more complex topology, such as branching or periodicity. To our knowledge, no existing algorithms jointly structure a single learned layout by both time and high-dimensional relationships.

3 Description of the IRIS Algorithm

3.1 Desiderata

To motivate our methods, we first define desiderata for structuring a *single* learned manifold by timestamps of data points, as this goal departs from both standard algorithms and existing time-aware variants like Dynamic t-SNE and Aligned UMAP. We note that these may not be universal and could depend on downstream uses of the layouts. However, we deem these to be reasonably broad enough to be axiomatic for defining a specific method.

I. *Nearby points in the layout should come from the same high-dimensional neighborhood*. Here we must depart slightly from the standard Manifold Learning desideratum to preserve distance orderings in the two spaces (Meilă & Zhang, 2024), because in a learned *temporal* manifold, distance in the layout will depend on time as well as high-dimensional space.

II. *High-dimensional neighborhoods should be contiguous in the layout.* This complements Desideratum I, which on its own could be fulfilled while allowing a neighborhood to "shatter" throughout the layout.

III. *Temporal branching structure should be preserved.* As tree-like trajectories commonly appear in natural phenomena (such as like cell differentiation), we wish to capture these.

IV. *The timestamp of each point should be estimable from layout position alone.* For an observer to make sense of a temporal manifold's dynamic structure, time must follow a simple pattern throughout the layout. We will formalize this as time being *estimable*.

3.2 Preliminaries

We next define the Manifold Learning problem and introduce notation. Given a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ of high-dimensional data (m observations of n variables) and a vector $\mathbf{t} \in \mathbb{R}^m$ of corresponding timestamps normalized to $[0, 1]$, we produce a low-dimensional layout $\mathbf{Y} \in \mathbb{R}^{m \times d}$. For the purposes of our method, we will assume d is 2, though the overall concepts could, in theory, be generalized to 3 dimensions. We assume some cost function $\mathcal{C}(\mathbf{X}, \mathbf{Y})$ that captures discrepancy between high- and low-dimensional pairwise distance orderings. Setting aside approximations for given implementations, we can generally define the optimization problem as:

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y} \in \mathbb{R}^{m \times d}} \mathcal{C}(\mathbf{X}, \mathbf{Y}).$$

3.3 REPARAMETERIZING THROUGH AN ANGULAR VARIABLE

To achieve temporal Manifold Learning at the population level (without requiring items to be tracked over time slices), we will explicitly structure a single, unified layout by time—that is, time will map to layout coordinates by some simple, monotonic function. A naïve, though fairly intuitive, approach to inducing this structure might be to embed directly to a single dimension (ignoring time), then simply plot that dimension against time to project back up to two dimensions. However, results from metric tree embeddings tell us that such a bottleneck would distort temporal branching structures (Indyk et al., 2004), in conflict with Desideratum III. To overcome this bottleneck, the low-dimensional embeddings must be influenced by distances over more than one dimension (e.g. the full two-dimensional space of the final layout).

Our solution is thus to instead *reparameterize* a cost defined in a 2-dimensional layout space through a single, latent dimension. This will have the effect of *constraining* a 2-dimensional optimization problem, rather than forcing it through a single-dimensional bottleneck. That is, for some injective function $g : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ mapping layout coordinate to time, the optimization will solve:

$$\mathbf{Y}^* = \arg\min_{\mathbf{Y} \in \mathbb{R}^{m \times d}} \mathcal{C}(\mathbf{X}, \mathbf{Y}) \quad \text{s.t.} \quad \forall\, \mathbf{Y}_{i,:},\ g(Y_{i,1}, Y_{i,2}) = t_i.$$

A particularly elegant way to define $g(\cdot)$, at least implicitly, is via a change of coordinates through the complex plane. Specifically, we reparameterize the cost function by the latent complex variable z, allowing optimization of the single variable θ:

$$\theta_i \in S^1 \mapsto \left[z_i = f_r(t_i; \rho) e^{\mathrm{i}\theta_i} \in \mathbb{C}\right] \cong \left[(Y_{i,1}, Y_{i,2}) \in \mathbb{R}^2\right] \to \mathcal{C}_i.$$

Here S^1 is the unit circle and $f_r(t_i; \rho)$ is a monotonic function mapping time to radius, parameterized by ρ, which we can choose *a priori* for each layout (we define f in §3.6; here we can simply assume some constant r_i for each θ_i). Our Manifold Learning optimization problem then becomes simply:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in (S^1)^m} \mathcal{C}(\mathbf{X}, \mathbf{Y}).$$

The gradient through the Jacobian of the reparameterization also takes a convenient form, making the partial derivative of θ agnostic to the particular cost function, and its Cartesian gradient:

$$\frac{\partial \mathcal{C}_i}{\partial \theta_i} = \underbrace{\begin{bmatrix} -r_i \sin\theta_i & r_i \cos\theta_i \end{bmatrix}}_{J_{f_r}(\theta_i)^\top} \begin{bmatrix} \dfrac{\partial \mathcal{C}_i}{\partial Y_{i,1}} \\ \dfrac{\partial \mathcal{C}_i}{\partial Y_{i,2}} \end{bmatrix}, \quad r_i = f_r(t_i; \rho).$$

The result of this approach is a polar layout, such that the manifold will intuitively seem to "grow" outward from the center of the layout, similar to the rings of a tree. We note that the use of an angular variable departs from typical Manifold Learning algorithms in that we are embedding into a quotient (periodic) space, in this case $\mathcal{M} = S^1 \times \mathbb{R}_{\geq 0}$, where $S^1 \cong \mathbb{R}/2\pi\mathbb{Z}$. Though this requires care if we compute distances in the angular space (see §3.4), a polar layout has several attractive qualities: (1) it creates a clean conceptual distinction between time and similarity when viewing the plot, (2) it keeps layouts compact, (3) the periodic nature of S^1 can better embed cyclical high-dimensional distance relationships than $\mathbb{R}$, and (4) it creates more space as time progresses, naturally accommodating the common phenomenon of dataset growth over time.

3.4 DEFINING THE COST FUNCTION

Though Manifold Learning algorithms based on stochastic gradient descent (including t-SNE and UMAP) boil down to attractive and repulsive forces computed from gradients, they vary in how they define their cost function C. In this case, we will adopt the probabilistic framework of Tang et al. (2016), as this formulation is particularly amenable to negative sampling approximations based on

a k-neighbors graph (McInnes et al., 2018). Under this framework, we build a k-neighbors graph in the original space, then define the probability of an edge existing between points in the layout:

$$P(e_{i,j} = 1) = f(\|Y_{i,:} - Y_{j,:}\|), \quad f(x) = \frac{1}{1 + x^2}.$$

The log probability is maximized for K near neighbors, and its inverse is maximized for sampled negative edges. Since we will optimize with stochastic gradient descent for pairs, we can define the positive objective C^+ and the negative objective C^- for pairs of embeddings $(\mathbf{y_i}, \mathbf{y_j})$ (positive pairs) and $(\mathbf{y_i}, \mathbf{y_k})$ (negative pairs) (Eq. 1). Gradients are defined in Appendix B

$$C_{i,j} = C^+_{i,j} + \gamma \sum_{k=1}^{K} C^-_{i,k}, \quad C^+_{i,j} = \log P(e_{i,j} = 1), \quad C^-_{i,k} = \log(1 - P(e_{i,k} = 1)) \tag{1}$$

3.5 Incorporating Angular Distances

Though a purely angular cost would make optimization effectively one-dimensional (causing a "bottleneck"), in practice it is helpful to incorporate angular cost in addition to the reparameterized Cartesian cost. This speeds convergence by avoiding "dead zones" (Fig A.8), and keeps classes more stably in "lanes," making periodicity more apparent in the layout. We thus introduce angular cost $\mathcal{C}_\theta$, where $P(e_{i,j} = 1) = f(\text{atan2}(\sin(\theta_2 - \theta_1), \cos(\theta_2 - \theta_1))$, to account for periodicity. We weight this by a hyperparameter, which we will call β (Eq. 2), which we discuss further in Appendix A.

$$\mathcal{C}_i = (1 - \beta)\mathcal{C}(\mathbf{X_{i,:}}, \mathbf{Y_{i,:}}) + \beta \cdot \mathcal{C}_\theta(\mathbf{X_{i,:}}, \theta_i) \tag{2}$$

3.6 Parameterizing the Radial Function

Ideally the radial function f_r should "warp" time to account for unevenly distributed timestamps, and to account for higher angular density near the center of the disc. To fulfill desideratum IV (estimability of time), however, we cannot use an arbitrarily complex function. Though this involves some subjectivity, the ubiquity of log transformations in data visualizations (Tufte, 1974) suggests that a power transformation should allow viewers to estimate timestamps from layout position, despite the density of time smoothly changing throughout the plot. We thus define radius as:

$$r_i = f_r(t_i; \rho) = \zeta + (1 - \zeta)t_i^{e^\rho}. \tag{3}$$

The hyperparameter $\zeta \in (0, 1)$ controls the ratio of inner to outer radius, creating an annulus. This avoids instability approaching the origin, where movement becomes limited. The hyperparameter ρ determines the amount of time warping. The composited exponential powering conveniently symmetrizes values of ρ, such that 0 is a linear mapping, positive values compress time rings toward the origin, and negative values toward the perimeter. Though ρ may be set manually, we provide a straightforward, data-driven way to choose it based on consistent layout density in Appendix C.

3.7 The IRIS Algorithm

We now have all the components to define the complete IRIS algorithm. First, the vector of time values is used to optimize the global radial parameter ρ via grid search. Then, a k-neighbors graph is constructed from the high-dimensional points $\mathbf{X}$. Angular parameters $\boldsymbol{\theta}$ are randomly initialized. Stochastic gradient descent is then performed over a number of steps, each step sampling an edge from graph, and a number of negative samples (pairs of vertices with no edge). For these samples, Cartesian coordinates are computed from their latent angular parameters and the globally parameterized function of time $f(t; \rho)$. Costs are then computed in both angular and Cartesian space. Gradients of these costs are then used to adjust angular parameters, with the Cartesian component of the cost backpropagating through the reparameterization. Finally, the optimized angles θ are converted to Cartesian coordinates for the final layout. The IRIS computation graph is depicted in Figure I.12.

4 Examining Branching Structure with Synthetic Data

4.1 Data

We first wish to examine whether IRIS can recover time-dependent topology. As this requires a known ground truth, we create a noisy, 3-dimensional, tree-like point cloud by updating a queue of gaussian generators (Fig. 3; Alg. I.1). Branch indices are kept as ground truth labels, and step number is used as the time variable for input to IRIS and time-structured baselines. To ensure a nontrivial temporal Manifold Learning problem, the parameters are tuned to create a "tumbleweed" shape, such that time is not monotone in any basis.

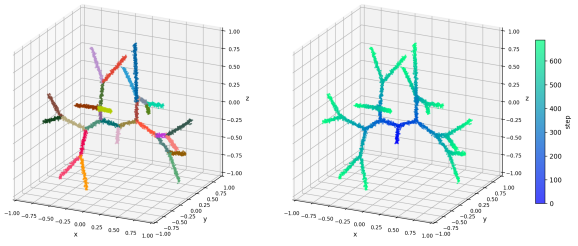

Figure 3: The synthetic tree dataset, shown colored by branch index (left) and time step (right).

4.2 Baseline Approaches

As our task is not well studied, for strong baselines, we consider how time could reasonably be incorporated into existing Manifold Learning algorithms, resulting in two overall approaches.

Supervision: Though most supervised Manifold Learning algorithms support only classification (discrete labels), the recently introduced algorithm RF-PHATE allows supervision of learning for both categorical and continuous variables, by combining Random Forest Geometry and Accuracy-Preserving proximities (RF-GAP) (Rhodes et al., 2026). This enables a straightforward baseline, by using time as the regression target.

Time as an Auxiliary Feature: Though standard, unsupervised Manifold Learning algorithms cannot explicitly incorporate time, we can include it as an auxiliary variable. To control relative weight, we z-scale each original variable, then then scale the auxiliary variable by $\lambda\sqrt{n}$, where n is the number of original dimensions and λ is a hyperparameter, with $\lambda = 1$ meaning time contributes approximately equally to squared distances as all other dimensions combined (Eq. 4). We apply this method to three standard algorithms: t-SNE, UMAP, and PHATE. We tune λ over $\{0.1, 0.3, 1, 3, 10\}$, covers each algorithms transition from original behaviour to degeneration (Fig. I.16—I.18).

$$\mathbf{X}'_i \in \mathbb{R}^{m\times(n+1)} = \left(\frac{X_{i,1}-\mu_1}{\sigma_1}, \ldots, \frac{X_{i,n}-\mu_n}{\sigma_n}, \lambda\sqrt{n}\,\frac{t_i-\mu_t}{\sigma_t}\right) \tag{4}$$

4.3 Metrics

Evaluation of time-structured Manifold Learning is complicated by the fact that near neighbors in the original space may be (properly) placed farther away in the layout due to a distal timestamp. Conversely, near neighbors in the layout may (properly) not be among the closest k in the high-dimensional neighborhood. This makes typical metrics relying on preservation of distance orderings—including trustworthiness and continuity (Venna & Kaski, 2001), metrics based on the co-ranking matrix Q (Lee & Verleysen, 2010)—inappropriate for assessing time-structured layouts.

Geodesic distances: Since we know that time will explicitly correlate with distance along the manifold from the root of our synthetic tree, we can leverage geodesic distances. Correlation of pairwise geodesic distances in the high-dimensional space with Euclidean distances in the layout is used to more accurately reflect "unrolling" of complex structures (Moon et al., 2019; Tsai, 2012). However, here we also expect a branching manifold within the low-dimensional layout, rather than a flattened sheet. We thus will use geodesic distances in both the high- and low-dimensional spaces.

Graph-based distances: We can also leverage ground truth branch labels and tree to measure how well a layout supports topological reconstruction, by building a Minimum Spanning Tree (MST) on branch centroids in the layout, and computing Graph Edit Distance (GED) (see Appendix D).

Pseudotime: In analyzing scRNA-seq data, computational biologists often use low-dimensional layouts of gene expression to compute 'pseudotime' (Street et al., 2018). We will compute this by projecting points to principle curves along tree paths (Appendix E).

Time-structuring: We use a layout to perform a Support Vector Regression (SVR) using a two-degree, non-homogeneous ($c = 1$) polynomial kernel, with five-fold cross-validation. This choice

of kernel accounts for nonlinear time-structuring (such as IRIS's radial encoding), but requires the structuring to be global and relatively simple, which mimics estimation by a human observer.

Table 1: Metrics for synthetic data experiments. All metrics are computed such that they lie in [0,1] or (0,1] and higher is better. We perform five replicates for all conditions and report the mean and Standard Error of the Mean (SEM). GM5 is computed over means and thus does not have SEM.

Metric	Type	Description	Question answered
SVR	Time-structuring	R^2 goodness of fit predicting time from layout coordinate (cross-validated)	*Is time estimable from layout position?*
GEO_w	Topology	Within-branch geodesic distance correlation	*How well are individual branches are reconstructed?*
GEO_b	Topology	Between-branch geodesic distance correlation	*How accurately are branches joined?*
GES	Topology	Graph Edit Similarity (normalized GED of MST to truth, subtracted from 1)	*Can the tree be reconstructed from layout coordinates?*
PTC	Topology / time-structuring	Pseudo-Time Correlation (computed from principle curves on the MST)	*Is time estimable from topology?*
GM5	**Aggregate**	Geometric mean of all five metrics (SVR, GEO_w, GEO_b, GES, PTC)	*How well are all metrics balanced?*

4.4 Settings

For all baseline algorithms (t-SNE, UMAP, PHATE and RF-PHATE), we use default parameters, except setting the random state to the integer replicate number (from 0), and specifying random initialization to ensure diversity across replicates. For auxiliary time variable baselines, we use λ with the highest GM5 for final experiments, resulting in (0.1, 3, 3) for (PHATE, UMAP, t-SNE), respectively. For IRIS, we set $\beta = 0$ (only the reparameterized Cartesian objective), as we are seeking to represent branching, rather intermittence of submanifolds. As an ablation for reparameterization, we also test with $\beta = 1$ (purely angular objective; no reparameterized influence). We set ζ to 0.1, making the inner radius 10% of the outer radius.

4.5 Results

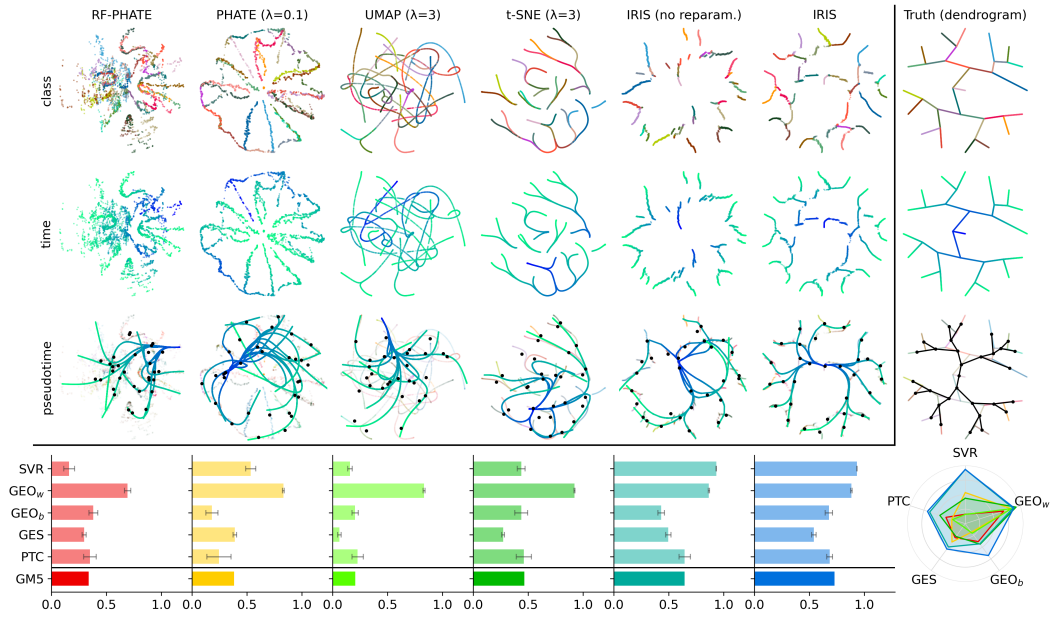

Figure 4: Synthetic data results.

IRIS outperforms all other methods for all metrics except within-branch geodesic distance correlation, which is topped by t-SNE with soft structuring via time as an auxiliary variable, with $\lambda = 3$ (Fig. 4, Table I.4). In other words, soft-structured t-SNE does a better job at keeping each branch contiguous, though this comes at the expense of accurate branch topology and the layout being arranged by time. RF-PHATE badly commingles branches, and fails to structure by its supervised variable (time). PHATE with an auxiliary time variable preserves individual branches well, but inverts the tree, placing leaves on the inside, which hurts topological inference. UMAP with an auxiliary time variable tangles branches, and still fails to structure by time. Interestingly, IRIS without reparameterizing the Cartesian cost function (that is, with a purely angular cost) fractures the topology of the tree, as prior metric tree embedding results and theory would predict (Indyk et al., 2004). This cleanly illustrates why reparameterization is needed to avoid a single-dimensional "bottleneck."

5 Evaluation with Real Data

5.1 Data

For experiments with real data, we choose three datasets of diverse modalities (Table 2). Additional details for preprocessing can be found in Appendix G.

Table 2: Summary of real datasets.

Dataset	**Modality**	**Samples**	**Features**	m	n	**Timestamps**
Embryo	Genes	Single cells from developing mouse embryos	First 300 principle components of normalized gene expression matrix	28,604	300	Developmental day
HuffPost	Text	News articles from Huffington Post Politics	Concatenated title and description embedded via transformer	26,104	1,536	Article date
Physics	Network	Nodes in a citation graph of physics papers	Fist 128 components from truncated SVD of adjacency matrix	10,284	128	Publication date

5.2 Metrics & Settings

As detailed in §4.3, standard Manifold Learning metrics are inappropriate for time-structured layouts. However, we can leverage class labels, which are not provided to any algorithms. If a manifold has been learned well, we would expect (1) nearby points in the low-dimensional layout to be of the same class Belkin & Niyogi (2004), and (2) classes to be contiguous throughout the layout (conveniently, this is true even if they are "stretched out" by time). We operationalize these as KNN and UTY (Table 1, Appendix F). We can also continue to use SVR to measure time-structuring. For baselines, we again use default settings, retuning λ for each dataset for auxiliary time variable baselines. For IRIS, we use $\beta = 0.95$, balancing representation of branching and periodicity.

Table 3: Metrics for experiments with real data. All metrics are computed such that they lie in [0,1] or (0,1] and higher is better. We perform five replicates for all conditions and report the mean and Standard Error of the Mean (SEM). GM5 is computed over means and thus does not have SEM.

Metric	**Type**	**Calculation**	**Question answered**
SVR	Time-structuring	R^2 goodness of fit predicting time from layout coordinate (see §4.3)	*Is time estimable from layout position?*
KNN	Class purity	Accuracy of a K-Nearest Neighbors classifier predicting class from layout coordinate, averaged over five folds	*Do layout neighbors come from the same high-dimensional neighborhood?*
UTY	Class unity	Macro-averaged inverse of connected components of class-induced neighborhood graph (1 means all classes have a single connected component)	*Are high-dimensional neighborhoods contiguous in the layout?*
GM3	**Aggregate**	Geometric mean of all other metrics (KNN, UTY, SVR)	*How well are all metrics balanced?*

5.3 RESULTS

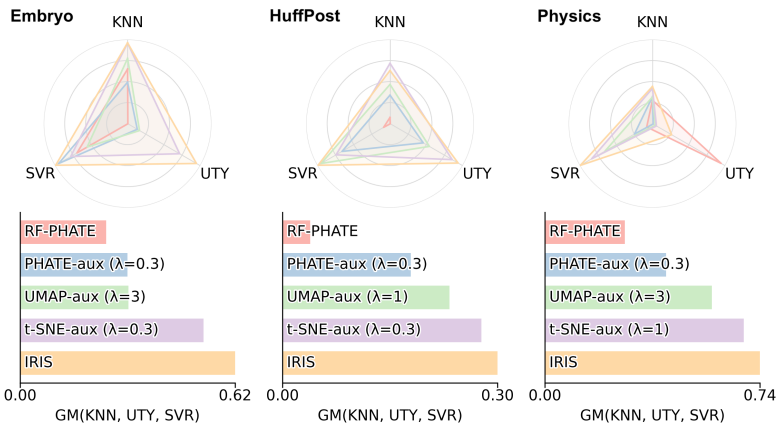

Figure 5: Results on real data.

As seen in Figure 5 (source data in Table I.5), IRIS outperforms all baselines in aggregate score. For individual metrics, t-SNE has better KNN (class purity) for HuffPost and Physics, though IRIS stays competitive while drastically improving SVR (time-structuring). IRIS is also close behind t-SNE and RF-PHATE in UTY (class unity) for the HuffPost and Physics datasets. All layouts can be seen in Figure I.13. As seen for the Embryo dataset in Figure 6, IRIS elucidates many dynamic phenomena that are not readily apparent from even the best performing alternative, especially with regards to the branching of epithelial cells, which t-SNE-aux shatters (see Fig. I.14). The HuffPost dataset, on the other hand, illustrates periodicity in news topics (Fig. I.15).

6 CONCLUSION

By incorporating both time and high-dimensional relationships directly into a single, low-dimensional layouts, IRIS unlocks temporal Manifold Learning at the population level. This allows intuitive visualization of myriad datasets that existing methods could not accommodate. As IRIS represents a new class of Manifold Learning algorithm, there is still work to be done around optimizing implementation and hyperparameters, and theoretical guarantees. Future work may also include jointly optimizing radial and angular parameters and extending to 3 dimensions using quaternions.

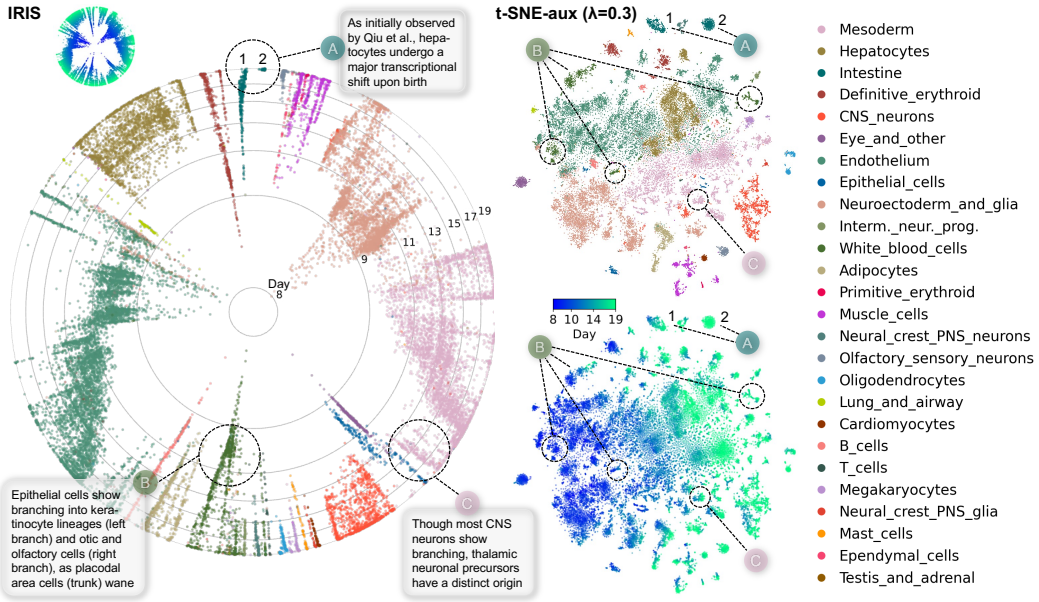

Figure 6: Comparison of IRIS and t-SNE with an auxiliary time variable for the Embryo dataset.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, design or provide feedback on research methodology or experiments, implement methods, create or edit software code, draft parts of a research paper, summarize or analyse existing literature, brainstorming, sourcing/searching for information, and identify relevant literature. Only supplemental methods sections in the appendix were drafted by generative AI, and these were thoroughly reviewed by the authors. Generative AI aided in choosing appropriate metrics, and brainstorming new metrics, and implementing metrics. Soundness of methods and code were checked by the authors. It also aided in correct notation and terminology for Gradients were derived manually. When generative AI was used for summarize areas in the literature, all references were followed to original papers to verify content. Generative AI was used to write code to specification to create the synthetic dataset and plots, but did not directly output any data or images in this paper.

We have not used generative AI tools to generate synthetic data sets, assist in the writing of proofs, propose or refine hypotheses, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results, provide critical ingredients for proving mathematical claims, formulate questions for surveys or interviews, create or modify scientific figures or images, suggest experimental parameters, creation of artifacts, transcribe recordings of research material, discover research topics or identify gaps, edit a research paper to improve readability, format references, suggest a structure for a research paper, propose a title or keywords for a research paper.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Source code for our IRIS implementation, code for running experiments, analyzing outputs, and creating figures is included in a supplemental file, and will be released publicly should this paper be published. We do not include *APS Datasets for Research* due to licensing; these may be obtained by contacting the American Physical Society at https://journals.aps.org/datasets to request access.

ACKNOWLEDGMENTS

This work was funded by [ANONYMIZED].

REFERENCES

American Physical Society. APS Data Sets for Research: Citing Article Pairs and Basic Bibliographic Metadata. `https://journals.aps.org/datasets`, 2010. Accessed: 2026-08-13.

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. Semi-supervised learning on riemannian manifolds. *Machine learning*, 56(1):209–239, 2004.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, October 2008. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/10/P10008.

Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 2019. doi: 10.23915/distill.00015. https://distill.pub/2019/activation-atlas.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Anant Dadu, Vipul K Satone, Rachneet Kaur, Mathew J Koretsky, Hirotaka Iwaki, Yue A Qi, Daniel M Ramos, Brian Avants, Jacob Hesterman, Roger Gunn, et al. Application of aligned-umap to longitudinal biomedical studies. *Patterns*, 4(6), 2023.

Rita González-Márquez, Luca Schmidt, Benjamin M Schmidt, Philipp Berens, and Dmitry Kobak. The landscape of biomedical research. *Patterns*, 5(6), 2024.

Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

John Hopcroft and Robert Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.

Piotr Indyk, Jirı Matoušek, and Anastasios Sidiropoulos. Low-distortion embeddings of finite metric spaces. *Handbook of discrete and computational geometry*, 37:46, 2004.

Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *Nature communications*, 10(1):5416, 2019.

John A Lee and Michel Verleysen. Scale-independent quality criteria for dimensionality reduction. *Pattern Recognition Letters*, 31(14):2248–2257, 2010.

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

Marina Meilă and Hanyu Zhang. Manifold learning: What, how, and why. *Annual Review of Statistics and Its Application*, 11, 2024.

Rishabh Misra. News category dataset. *arXiv preprint arXiv:2209.11429*, 2022.

Kevin R Moon, Jay S Stanley III, Daniel Burkhardt, David van Dijk, Guy Wolf, and Smita Krishnaswamy. Manifold learning-based methods for analyzing single-cell rna-sequencing data. *Current Opinion in Systems Biology*, 7:36–46, 2018.

Kevin R Moon, David Van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature biotechnology*, 37(12):1482–1492, 2019.

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. *BioRxiv*, pp. 731877, 2019.

Chengxiang Qiu, Beth K Martin, Ian C Welsh, Riza M Daza, Truc-Mai Le, Xingfan Huang, Eva K Nichols, Megan L Taylor, Olivia Fulton, Diana R O'Day, et al. A single-cell time-lapse of mouse prenatal development from gastrula to birth. *Nature*, 626(8001):1084–1093, 2024.

Paulo E Rauber, Alexandre X Falcão, and Alexandru C Telea. Visualizing time-dependent data using dynamic t-sne. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers*, pp. 73–77, 2016.

Jake S Rhodes, Adrien Aumon, Sacha Morin, Marc Girard, Catherine Larochelle, Elsa Brunet-Ratnasingham, Amélie Pagliuzza, Lorie Marchitto, Wei Zhang, Adele Cutler, et al. Gaining biological insights through supervised data visualization. *Nature Computational Science*, pp. 1–21, 2026.

Alberto Sanfeliu and King-Sun Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (3):353–362, 1983.

Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B. Viégas, and Martin Wattenberg. Embedding projector: Interactive visualization and interpretation of embeddings, 2016. URL https://arxiv.org/abs/1611.05469.

Jian Song and Xue Chen. Research on hot news topic discovery and communication mechanism by fusion clustering algorithm. 2026.

Kelly Street, Davide Risso, Russell B Fletcher, Diya Das, John Ngai, Nir Yosef, Elizabeth Purdom, and Sandrine Dudoit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC genomics*, 19(1):477, 2018.

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.

Flora S Tsai. A visualization metric for dimensionality reduction. *Expert Systems with Applications*, 39(2):1747–1752, 2012.

Edward R Tufte. *Data analysis for politics and policy*. Prentice-Hall Englewood Cliffs, NJ, 1974.

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.

A Balancing Angular and Cartesian Costs with β

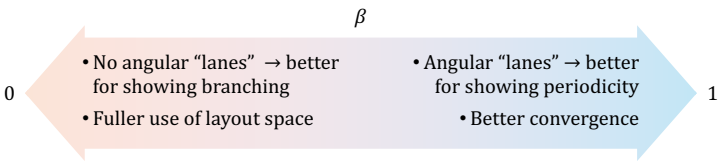

Figure A.7: Tradeoffs when setting β.

Our reparameterization of a Cartesian objective via a latent angular variable overcomes the "bottleneck" problem. However, an objective purely defined in Cartesian space introduces its own problem for our polar formulation: the gradient of any such objective with respect to an angular variable shrinks to zero for points at the same angle (where repulsive forces for non-neighbors are most needed) and on opposite sides of the circle (where attractive forces for neighbors are most needed). This results in undesirable "dead zones" that may cause inefficient or unstable optimization (Fig A.8).

An intuitive geometric interpretation of β is as a scaling factor for radii, where 1 is infinite scaling (making the angular contribution to cost 0) and 0 flattens the annulus to a circle (making Cartesian contribution to cost 0). The practical implication of this framing is that it makes sense to sweep β asymptotically towards 1, rather than linearly. Example layouts at various β values can be seen in Figure I.22.

B Derivation of Cost Functions and Gradients

Our expanded cost functions are:

$$\mathcal{C}^{+}_{i,j} = (1-\beta)\log\frac{1}{1+(||\mathbf{y}_i-\mathbf{y}_j||^2)} + \beta\log\frac{1}{1+(\theta_i-\theta_j)^2}$$

$$\mathcal{C}^{-}_{i,k} = (1-\beta)\log\left[1-\frac{1}{1+(||\mathbf{y}_i-\mathbf{y}_k||^2)}\right] + \beta\log\left[1-\frac{1}{1+(\theta_i-\theta_k)^2}\right]$$

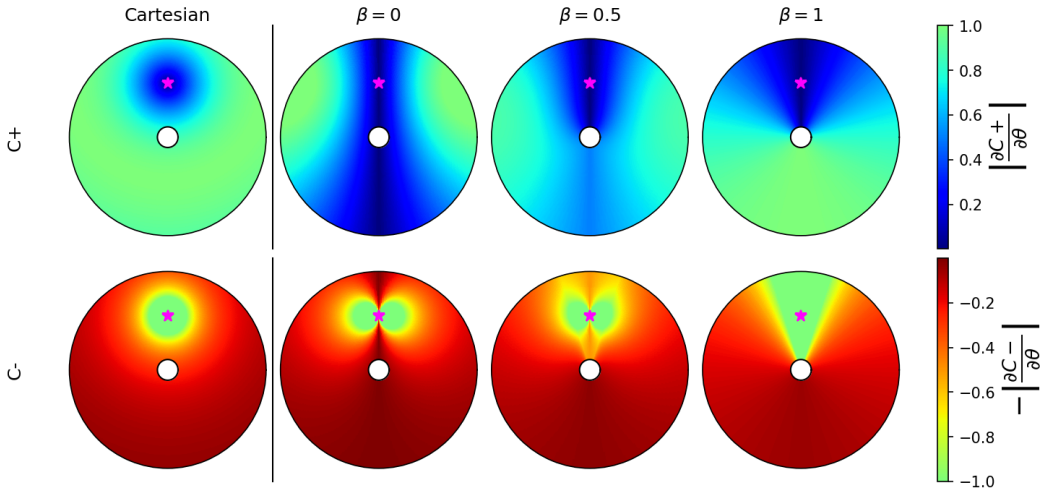

Figure A.8: Analysis of Gradients. Top, magnitudes of the positive cost C^+ with respect to θ at the starred position. Bottom the same for negative costs C^-. . For reference, the first column shows norms of two-dimensional gradients for standard Manifold Learning optimization. As seen for $\beta = 0$, positive points opposite the starred reference have no attractive force (a "dead zone"). Similarly, the negative cost is 0 directly above and below the reference point. Though $\beta = 1$, solves the issue for attractive forces (top), it creates "hot zones" in the negative gradient above and below the reference point, forming a "lane" where only high-dimensional neighbors can go, and hurting representation of complex branching structures. Here, $\beta = 0.5$ illustrates mitigating both effects and offering a balance.

To derive the gradient of these costs with respect to θ, we first define the gradient of the cost C for a pair of embeddings $(\mathbf{y}_i, \mathbf{y}_j)$ with respect to a single embeddding $\mathbf{y}_i$:

$$\nabla_{\mathbf{y}} \mathcal{C}^+_{i,j} = (1-\beta)\frac{-2(\mathbf{y}_i - \mathbf{y}_j)}{1 + ||\mathbf{y}_i - \mathbf{y}_j||^2} \tag{5}$$

$$\nabla_{\mathbf{y}} \mathcal{C}^-_{i,k} = (1-\beta)\frac{2(\mathbf{y}_i - \mathbf{y}_k)}{||\mathbf{y}_i - \mathbf{y}_k||^2(1 + ||\mathbf{y}_i - \mathbf{y}_k||^2)} \tag{6}$$

The partial derivatives of $\mathcal{C}^+_{i,j}$ and $\mathcal{C}^-_{i,k}$ with respect to θ_i (the angle used to define $\mathbf{y}_i$) are then:

$$\frac{\partial \mathcal{C}^+_{i,j}}{\partial \theta_i} = r_i(1-\beta)\left(\frac{\partial \mathcal{C}^+_{i,j}}{\partial y_i^{(2)}}\cos\theta_i - \frac{\partial \mathcal{C}^+_{i,j}}{\partial y_i^{(1)}}\sin\theta_i\right) + \frac{-2\beta(\theta_i - \theta_j)}{1 + (\theta_i - \theta_j)^2} \tag{7}$$

and

$$\frac{\partial \mathcal{C}^-_{i,k}}{\partial \theta_i} = r_i(1-\beta)\left(\frac{\partial \mathcal{C}^-_{i,k}}{\partial y_i^{(2)}}\cos\theta_i - \frac{\partial \mathcal{C}^-_{i,k}}{\partial y_i^{(1)}}\sin\theta_i\right) + \frac{2\beta(\theta_i - \theta_k)}{(\theta_i - \theta_k)^2(1 + (\theta_i - \theta_k)^2)}. \tag{8}$$

Due to symmetry, $\frac{\partial \mathcal{C}^+_{i,j}}{\partial \theta_j} = -\frac{\partial \mathcal{C}^+_{i,j}}{\partial \theta_i}$ and $\frac{\partial \mathcal{C}^-_{i,k}}{\partial \theta_k} = -\frac{\partial \mathcal{C}^-_{i,k}}{\partial \theta_i}$. For each sampled pair $(\mathbf{y}_i, \mathbf{y}_j)$ and $(\mathbf{y}_i, \mathbf{y}_k)$, we can thus gain efficiency by simultaneously performing updates to $\mathbf{y}_i$, $\mathbf{y}_j$ and $\mathbf{y}_k$.

C Setting the Hyperparameter ρ

To determine an optimal ρ for a given dataset, we will first define an ideal distribution $P(X)$ that we would like radii of our embedded points to exhibit. As an objective, we will aim to make density of the final layout approximately equal for any given time value.

For an increase in radius δr, the area a of a circle increases proportionally ($\delta a \propto \delta r$), so $P(X)$ must increase linearly from the inner radius of ζ to the outer radius of 1. This means the PDF will form a right trapezoid (Fig. C.9). As the integral of a PDF must be 1, and the base has length $1-\zeta$, we can solve for the parallel sides. The probability density function to evenly distribute a random variable throughout a unit annulus of fractional inner radius ζ is thus:

$$p(x) = \begin{cases} 0, & x < 0 \\ \frac{2\zeta}{1+\zeta} + \frac{2-2\zeta}{1+\zeta}x, & 0 \le x \le 1 \\ 0, & x > 1 \end{cases} \tag{9}$$

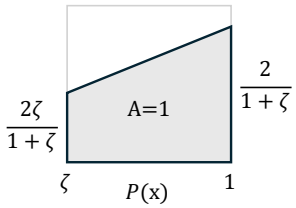

Figure C.9: The PDF producing even density for a unit annulus of inner radius ζ.

We wish to find the value ρ^* that makes the distribution of radii Q most similar to this ideal using Kullback-Leibler divergence:

$$\rho^* = \arg\min_{\rho} D_{\mathrm{KL}}(P \| Q) \tag{10}$$

As a closed form of Q is not available to us, we compute a probability mass function over b equally sized bins for a given value of ρ. To avoid 0-frequency bins (which would make the KL divergence undefined), we apply Laplace smoothing with $\alpha = 1$. We then accordingly discretize Eq. 9 by integrating over each bin to obtain P. We can then use the discrete form of the KL divergence:

$$D_{KL}(P \parallel Q) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{11}$$

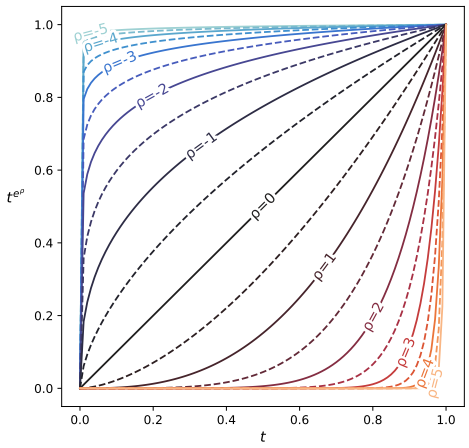

Figure C.10: The influence of ρ on warping time.

Though the objective in Eq. 10 is non-convex, in practice it can be easily optimized via grid search (Fig. C.11). This is due to the facts that (1) it is tractable to compute, (2) we optimize ρ independently of other parameters, and (3) the composited power function in Eq. 3 ensures that a linear search of a small range of potential ρ values, such as $[-5, 5]$ in intervals of 0.01, will cover both extremes while providing adequate granularity for more central values (Fig. C.10).

D Graph Edit Similarity

To measure whether a two-dimensional layout recovers the known branching topology, we compare a ground-truth cluster tree to a tree inferred from the embedding. Let $T^\star = (V^\star, E^\star)$ denote the ground-truth tree read from the dataset metadata: each point carries a cluster (branch) label, and each cluster records its children. The resulting graph is an undirected tree on these cluster labels; its unique source (the unique node that is never listed as a child) is treated as the root $r^\star$.

Inferred tree. Given a layout $Y \in \mathbb{R}^{m \times 2}$ and the same cluster labels, we compute a minimum spanning tree on the cluster centroids of Y.

Labeled edit distance. Because both trees are labeled by the same cluster identities, a node is identified with its label and no label-substitution search is required. We therefore use the unit-cost graph edit distance that inserts or deletes unmatched nodes and edges Sanfeliu & Fu (1983):

$$d(T^\star, T) = |V^\star \triangle V| + |E^\star \triangle E|, \tag{12}$$

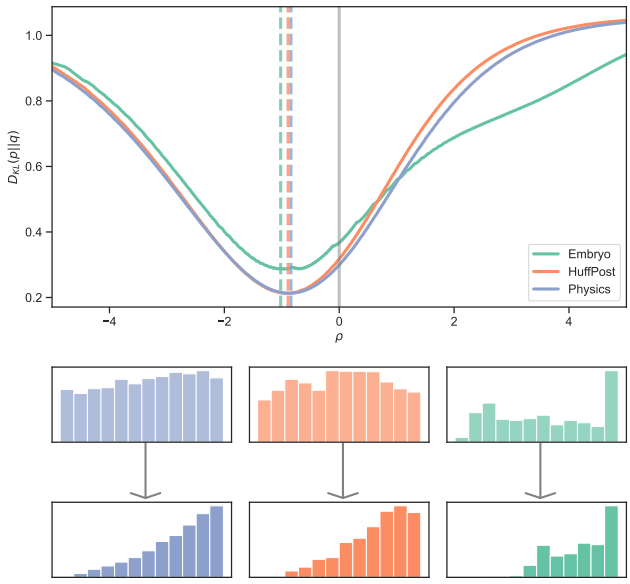

Figure C.11: Cost Curves for Optimizing ρ. Bottom, temporal histograms before and after optimization.

where $\triangle$ denotes symmetric difference. Each missing or extra cluster contributes one node edit, and each missing or extra adjacency contributes one edge edit. This is exactly the size of the labeled graph difference: $d(T^\star, T) = 0$ if and only if the inferred topology matches the ground truth.

Normalization. Raw edit counts grow with tree size, so we report a size-normalized similarity. Deleting an entire graph $G = (V, E)$ costs $|V| + |E|$. We normalize by the larger of the two trees,

$$\mathrm{GES}(T^\star, T) = 1 - \frac{d(T^\star, T)}{\max\big(|V^\star| + |E^\star|,\ |V| + |E|\big)},$$

and define $\mathrm{GES} = 1$ when both graphs are empty. Larger values indicate closer topological agreement.

E Pseudotime Correlation

To measure whether a two-dimensional layout recovers the continuous temporal order of the data, we compare ground-truth timestamps to the pseudotime of a trajectory inferred from the embedding. As in Section D, we run Slingshot Street et al. (2018) on the layout $Y \in \mathbb{R}^{m\times 2}$ with the ground-truth cluster labels and starting cluster $r^\star$. Slingshot fits a principal curve to each lineage $k = 1, \ldots, K$ and assigns every point i a lineage-specific pseudotime τ_{ik}, left undefined when i is not assigned to lineage k Street et al. (2018). When a point belongs to several lineages—typically near a branch point—we replace the lineage-specific values by their average,

$$\bar{\tau}_i = \frac{1}{|\mathcal{K}_i|} \sum_{k\in\mathcal{K}_i} \tau_{ik}, \qquad \mathcal{K}_i = \{k : \tau_{ik} \text{ is defined}\}. \tag{13}$$

Points with $\mathcal{K}_i = \emptyset$, and any point whose ground-truth time t_i is missing, are discarded. Let I be the remaining index set, and write R_i (resp. S_i) for the rank of t_i (resp. $\bar{\tau}_i$) among $\{t_j\}_{j\in I}$ (resp. $\{\bar{\tau}_j\}_{j\in I}$), using midranks in the event of ties. We report Spearman's rank correlation.

$$\mathrm{PTC} = \frac{\sum_{i\in I}(R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_{i\in I}(R_i - \bar{R})^2}\,\sqrt{\sum_{i\in I}(S_i - \bar{S})^2}}, \tag{14}$$

where $\bar{R}$ and $\bar{S}$ are the mean ranks. The statistic equals $+1$ when the inferred trajectory is a strictly monotone reparameterization of true time, 0 when the two sequences are unassociated, and -1 when

the ordering is reversed. If Slingshot cannot be run, no pseudotime is available and the metric is left undefined. For the synthetic tree dataset, t_i is the known simulation time of each point.

F CLASS UNITY

To measure whether a two-dimensional layout keeps each labeled class in a single contiguous region, we examine connectivity on a neighborhood graph of the embedding. Given a layout $Y \in \mathbb{R}^{m\times 2}$ and class labels $y_i \in \mathcal{C}$, we form the undirected k-nearest-neighbor graph $G = (V, E)$ with $k = 50$ (Cover & Hart, 1967): vertices are the embedded points, and $\{i, j\} \in E$ whenever i is among the k Euclidean nearest neighbors of j in the plane, or vice versa. For each class $c \in \mathcal{C}$, let G_c be the subgraph induced by $V_c = \{i : y_i = c\}$, and write $\kappa(G_c)$ for the number of connected components of G_c (Hopcroft & Tarjan, 1973). Class-wise unity is the reciprocal of this count, and we report the macro-average

$$\mathrm{UTY} = \frac{1}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} \frac{1}{\kappa(G_c)}. \tag{15}$$

The statistic equals 1 when every class occupies a single connected region of the neighborhood graph, and falls toward 0 as classes fragment into many isolated pieces. Unlike local classification accuracy, which can remain high when a class is split into several internally pure islands, UTY penalizes that fragmentation.

G DATA SOURCES AND PREPARATION

Embryo (scRNA-seq): A single-cell RNA-seq dataset capturing mouse embryonic development from gastrula to birth (Qiu et al., 2024). Each sample (i) is a cell, and each dimension (j) is a gene, with each value $X_{i,j}$ representing a normalized expression level. We use "normalized subset 1" from Cell X Gene,[1] comprising 2,860,465 cells, and further sample %1 of these, resulting in 28,604 cells. Samples are timestamped, for the most part, in quarter-day intervals during embryonic development, resulting in 42 discrete day values in $[8.00, 18.75]$, and at birth, which we assign a day value of 19.00. Due to the small number of discrete values relative to sample count, and the heterogeneous intervals, we apply temporal resampling:

$$t'_i \sim U(t_i, \min_{j;\, t_j > t_i} t_j).$$

As these data tend to be highly sparse (having mostly zero values) we perform Principal Component Analysis (PCA) and use the 300 components of highest variance as input to Manifold Learning algorithms. The first 300 PCA components, in sum, explain 30.00% of the variance of the original 45,525 dimensions. We use the provided `author_major_cell_cluster` values as class labels for post-hoc analysis, and use `author_cell_type` values for further exploration of cell type arrangement within clusters. Class labels are cell types as reported in Cell X Gene,[2] from which the data were downloaded.

HuffPost (News): Semantic embeddings of concatenated titles and short descriptions of Huffington Post politics articles. We begin with the News Category Dataset, which comprises 200,853 Huffington Post articles spanning from 2012 to 2018 Misra (2022). Each entry includes the title, date and a short description of a news article. The dataset is originally organized into 42 categories, such as politics, wellness, and entertainment. However, these are news sections rather than topics, and so are inappropriate as labels. We thus extract articles in "politics" (the largest category), and run topic modeling using BERTopic Grootendorst (2022). As this dataset has known inconsistencies due to the history of how the Huffington Post produced articles and provided data (Song & Chen, 2026), we further filter by date, selecting articles from July 1st, 2015 (after the Verizon acquisition) to May 31st (before major website infrastructure and data sharing changes). We encode the text data using `text-embedding-3-small` into 1,536-dimensional embeddings.

[1] `https://cellxgene.cziscience.com/e/a91d5064-aba3-4688-9fcd-6187b2f435a4.cxg/`

[2] `https://cellxgene.cziscience.com/e/37a17b78-4864-4a42-b67b-31c00962795a.cxg/`

American Physical Society (APS) Gravitational Physics Articles Dataset A subset of the corpus of 450,000 articles from APS journals Physical Review Letters, Physical Review, and Reviews of modern physics published between the years of 1893 and 2021 American Physical Society (2010). The *APS Datasets for Research* comprises two datasets: the first is a collection of citing article pairs which determines the structure of the citation network of journal articles within the dataset, and the second is a collection of article metadata which corresponds to the articles in the citing pairs dataset. Of the metadata, we use `date`, the article's DOI (`id`), the publication date (`date`), and the article's Physics subject Heading discipline and concepts (PhySH, *classificationSchemes.physh.discipline*,*classificationSchemes.physh.concepts*). We select articles from January 2016 on, as this is when PhySH was introduced. Article disciplines are singularly applied to each paper with rare exception. Article concepts are numerous, however authors may label their paper with a primary research area, which is singular per article, also with rare exception. There are over 2,500 unique PhySH concepts referenced as the primary concept in papers in the greater than 140,000 papers sampled after excluding papers without PhySH labels in the dataset. We focus the 'Gravitation, Cosmology and Astrophysics' disciple, and keep only articles containing the top 30 most numerous PhySH Primary concepts, yielding 10,284 articles. Labels are communities detected with the Louvain method provided in the NetworkX package of scikit-learn for community detection to determine labels for further data analysis (Pedregosa et al. (2011),Blondel et al. (2008)).

H Implementation Details

We use the originally reported UMAP implementation,[3], OpenTSNE (Poličar et al., 2019), and the originally reported implementations of PHATE and RF-PHATE. Learning for `SVR` and `KNN` metrics is performed with Scikit-learn Pedregosa et al. (2011). For experiments, we perform five replicates for each combination of dataset and algorithm. Each replicate uses different random initializations for layouts and cross-validation splits. Means and standard deviations of metrics are computed across replicates, using means from cross-validation as observations. For tractability, we compute `SVR` for 10,000 randomly chosen points (without replacement) for datasets with more than 10,000 points. Other metrics are computed from all points. For additional IRIS hyperparameters shared by LargeVis (Tang et al., 2016), we set $\gamma = 32$, the number of neighbors to 128, the number of propagations to 3, the number of steps (neighbor pairs sampled) to $m * 10^3$ for the dataset size m, the learning rate α to 1, and the number of negative samples per positive to 5 (the default) for synthetic data experiments. For real data, we use $\gamma = 128$ and the number of neighbors to 32, to account for the larger dataset sizes.

I Additional Algorithms, Tables, and Figures

[3] `https://github.com/lmcinnes/umap`

Algorithm I.1 Synthetic data generation. Each point is sampled from an isotropic Gaussian of standard deviation $\sigma = 0.5$ whose center advances at speed $s = 0.1$ with time increment $\Delta t = 1$. A live branch splits independently with probability $\tau = 0.025$ once it has taken at least $\ell_{\min} = 100$ steps, subject to a cap of $B_{\max} = 16$ concurrent branches. The initial fork is drawn at angle $\alpha_0 = 80^\circ$ with jitter $\delta_0 = 16^\circ$; after each split both the angle and the jitter of that lineage are multiplied by $\rho = 0.75$. The simulation is initialized with random seed 0.

Require: number of points m; split probability τ; Gaussian std. σ; speed s; time step Δt; refractory length $\ell_{\min}$; concurrent-branch cap $B_{\max}$; initial split angle α_0 and jitter δ_0; angle decay $\rho \in (0, 1]$; number of roots R

Ensure: coordinates $X \in \mathbb{R}^{m\times 3}$, times $t \in \mathbb{R}^m$, branch IDs $b \in \mathbb{Z}^m$, children map C

$X, t, b \leftarrow (\,),\ C \leftarrow \{\,\},\ id \leftarrow R$
$\mathcal{A} \leftarrow \{\,(\,i,\ \mathbf{0},\ u_i,\ 0,\ 0,\ \alpha_0,\ \delta_0\,) :\ i = 0, \ldots, R-1,\ u_i \sim \mathrm{Unif}(\mathbb{S}^2)\,\}$
while $|X| < m$ **do**
  $\mathcal{A}' \leftarrow \{\,\},\ B \leftarrow |\mathcal{A}|$
  for each branch $(id_j, \mu_j, d_j, t_j, \ell_j, \alpha_j, \delta_j)$ in $\mathcal{A}$ **do**
    sample $x \sim \mathcal{N}(\mu_j, \sigma^2 I_3)$; append x, t_j, id_j to X, t, b
    $\mu_j \leftarrow \mu_j + s\, d_j,\ t_j \leftarrow t_j + \Delta t,\ \ell_j \leftarrow \ell_j + 1$
    if $\ell_j \geq \ell_{\min}$ **and** $B < B_{\max}$ **and** $U \sim \mathrm{Unif}(0,1) < \tau$ **then**
      draw a unit vector $u \perp d_j$; sample $\theta, \phi \sim \alpha_j + \mathrm{Unif}(-\delta_j, \delta_j)$
      $d^+ \leftarrow \mathrm{normalize}(\cos\theta\, d_j + \sin\theta\, u)$
      $d^- \leftarrow \mathrm{normalize}(\cos\phi\, d_j - \sin\phi\, u)$
      $C(id_j) \leftarrow \{id,\ id+1\}$
      spawn two children at μ_j with directions $d^\pm$, angles $\rho\alpha_j, \rho\delta_j$, and age 0
    else
      keep the updated branch
    end if
  end for
  $\mathcal{A} \leftarrow \mathcal{A}'$
end while
independently affinely map each column of X onto $[-1, 1]$
return X, t, b, C

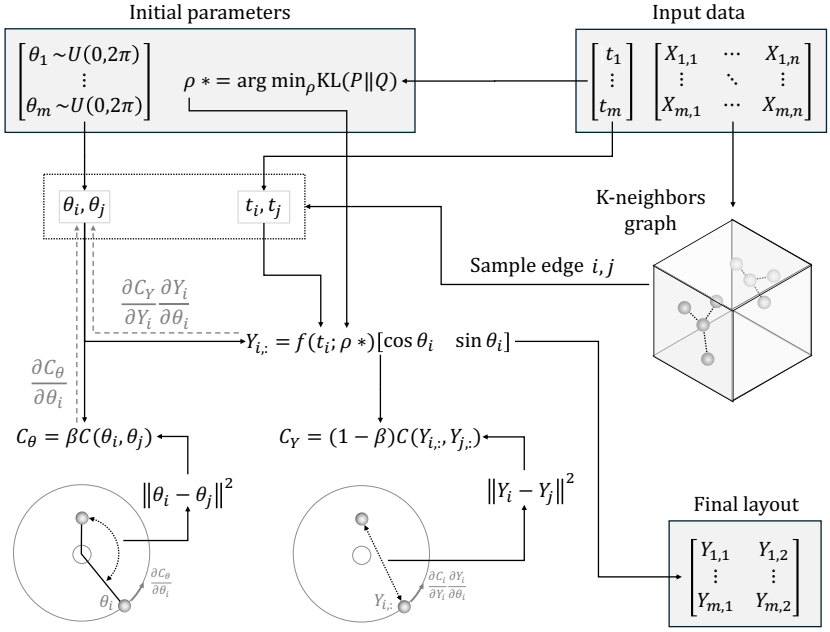

Figure I.12: The IRIS computation graph.

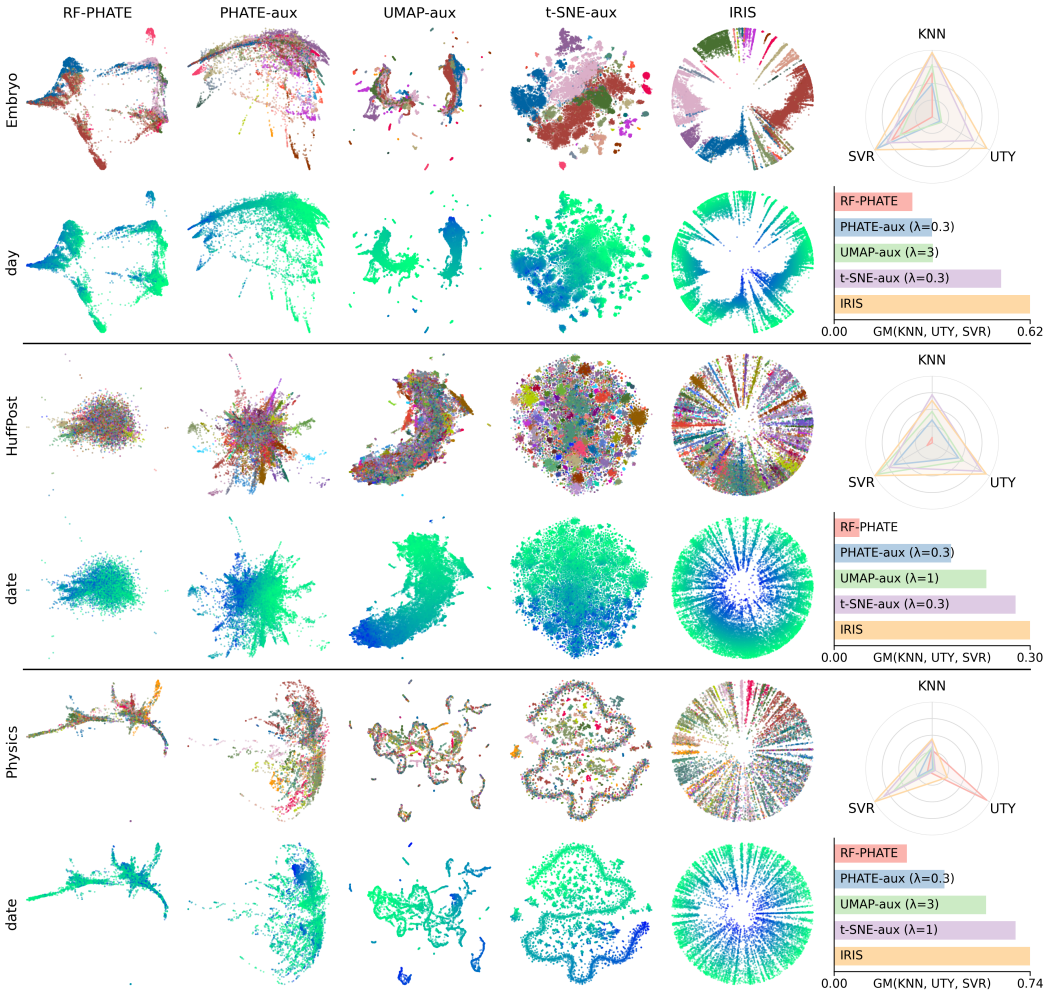
RF-PHATE
PHATE-aux
UMAP-aux
t-SNE-aux
IRIS
Embryo
day
HuffPost
date
Physics
date
KNN
SVR
UTY
RF-PHATE
PHATE-aux (λ=0.3)
UMAP-aux (λ=3)
t-SNE-aux (λ=0.3)
IRIS
0.00
GM(KNN, UTY, SVR)
0.62
KNN
SVR
UTY
RF-PHATE
PHATE-aux (λ=0.3)
UMAP-aux (λ=1)
t-SNE-aux (λ=0.3)
IRIS
0.00
GM(KNN, UTY, SVR)
0.30
KNN
SVR
UTY
RF-PHATE
PHATE-aux (λ=0.3)
UMAP-aux (λ=3)
t-SNE-aux (λ=1)
IRIS
0.00
GM(KNN, UTY, SVR)
0.74

Figure I.13

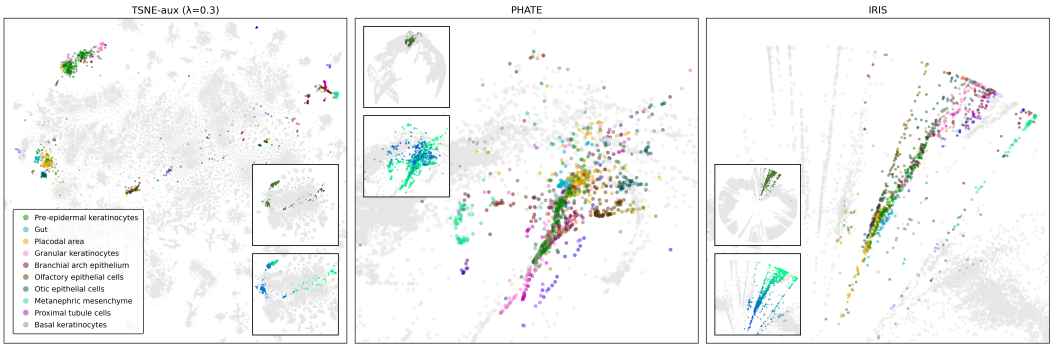

Figure I.14: Comparison of representations of branching in epithelial cells in the Embryo dataset. Legends are restricted to the 10 most frequent epithelial subtypes. Upper insets show locations of the epithelial cell cluster within the main plot; lower insets show time coloring. Though t-SNE with an auxiliary time variable ($\lambda = 0.3$) creates a clear temporal ordering within the plot, it shatters the epithelial cluster, making it impossible to reconstruct branching (left). PHATE (standard; no auxiliary variable), while developed specifically to recover branching structure in scRNA-seq data, also does not capture divergence of epithelial subtypes over time (middle). However, by incorporating both time and high-dimensional data—that is, learning the *temporal manifold*, IRIS clearly shows placodal area cells diverging into keritinocytes (the major cells of skin) and olfactory (odor-sensing) cells (right).

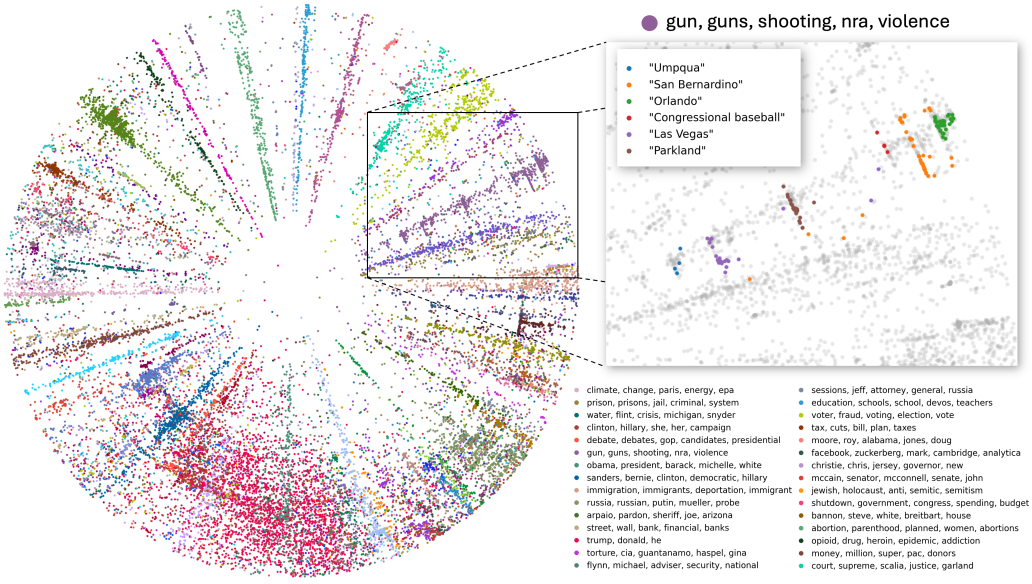

Figure I.15: Illustration of periodicity. In this case, a topic around gun violence can be seen to have periodic "studs" as related news items create activity. Zooming in and highlighting points by keywords shows these studs relate to specific shooting events.

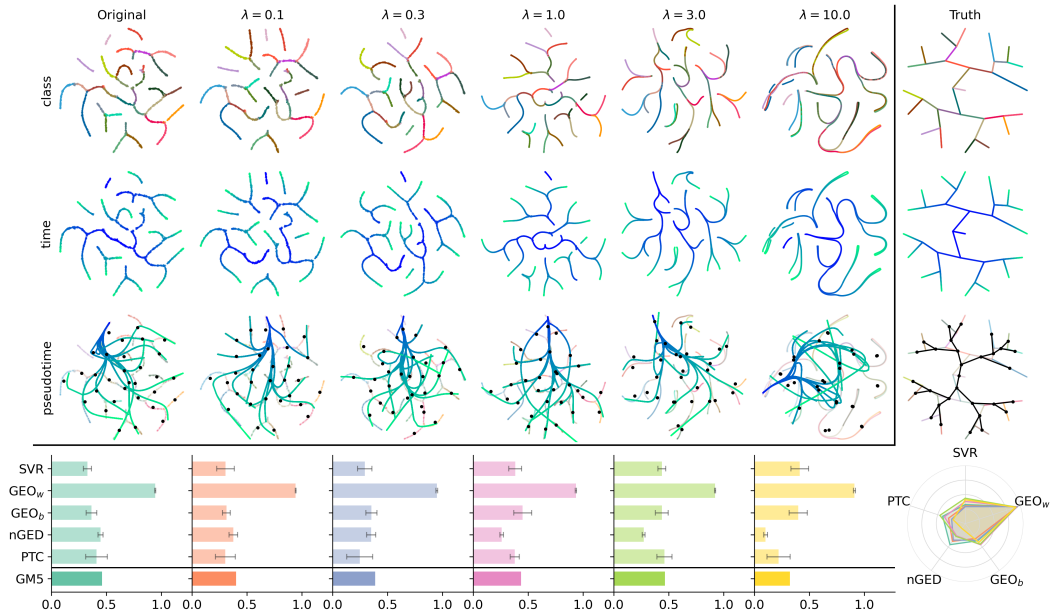

Figure I.16: Hyperparameter optimization for t-SNE with an auxiliary time variable. GM is the geometric mean of all metrics.

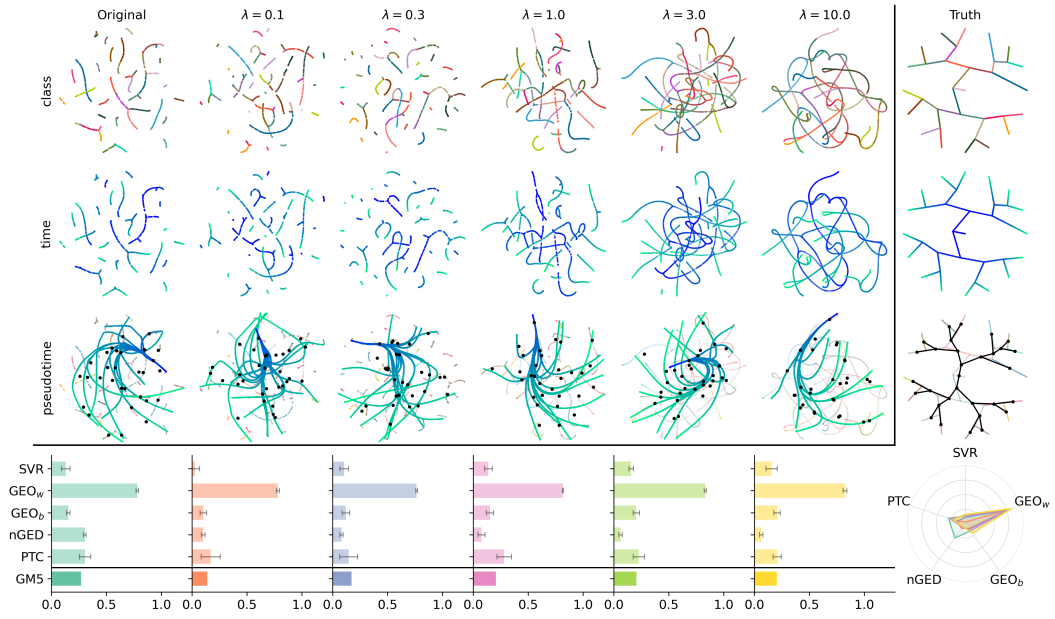

Figure I.17: Hyperparameter optimization for UMAP with an auxiliary time variable. GM is the geometric mean of all metrics.

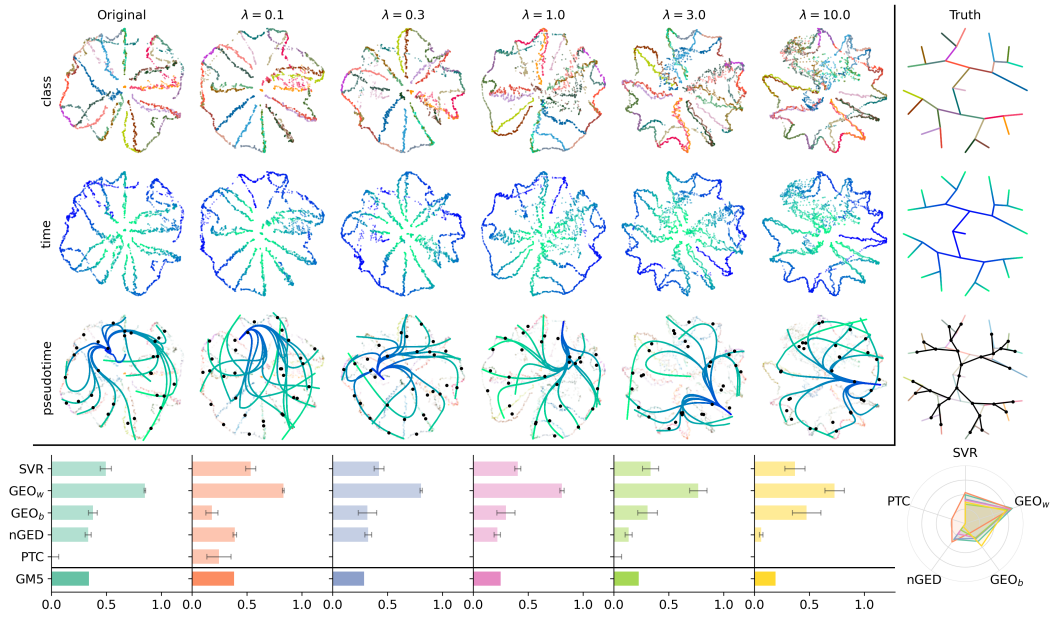

Figure I.18: Hyperparameter optimization for PHATE with an auxiliary time variable. GM is the geometric mean of all metrics.

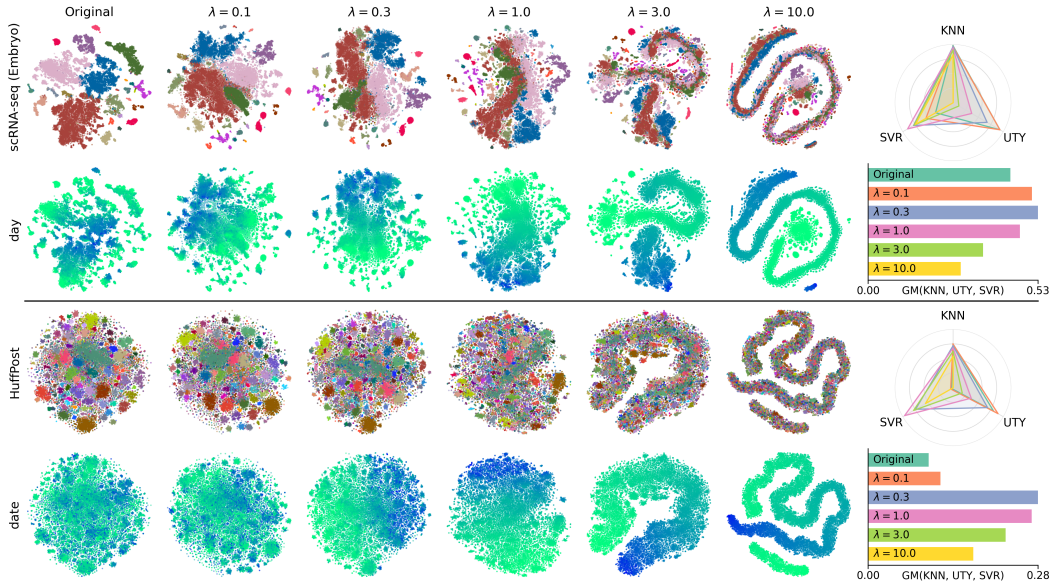

Figure I.19: Hyperparameter optimization for t-SNE with an auxiliary time variable.

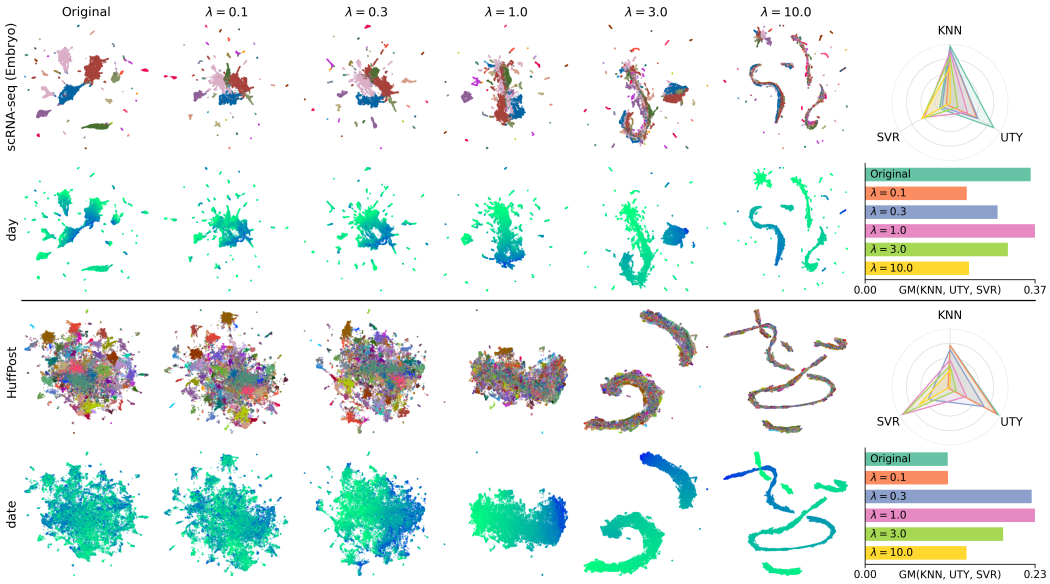

Figure I.20: Hyperparameter optimization for UMAP with an auxiliary time variable.

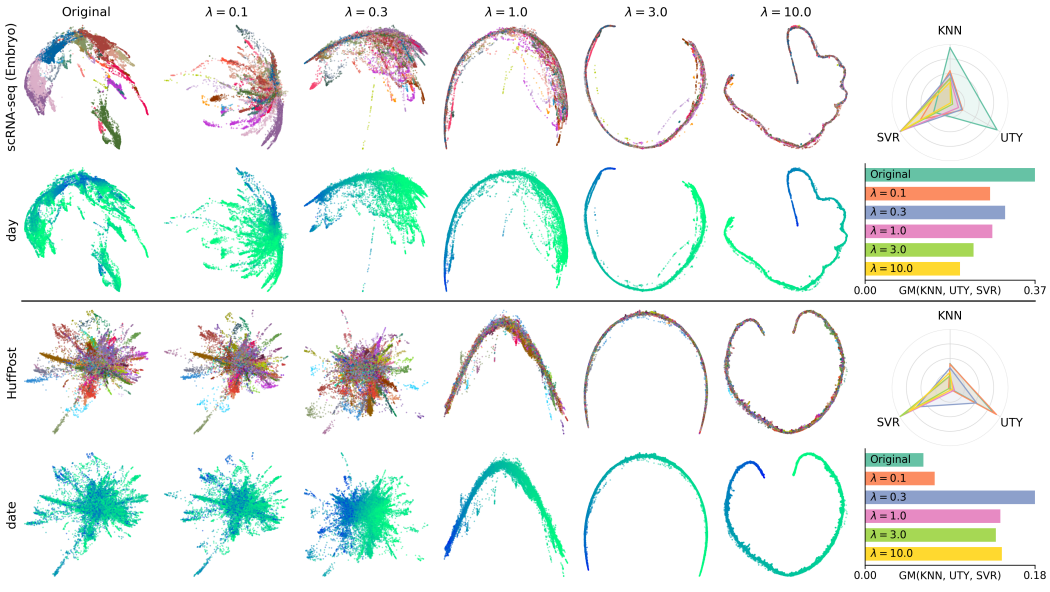

Figure I.21: Hyperparameter optimization for PHATE with an auxiliary time variable.

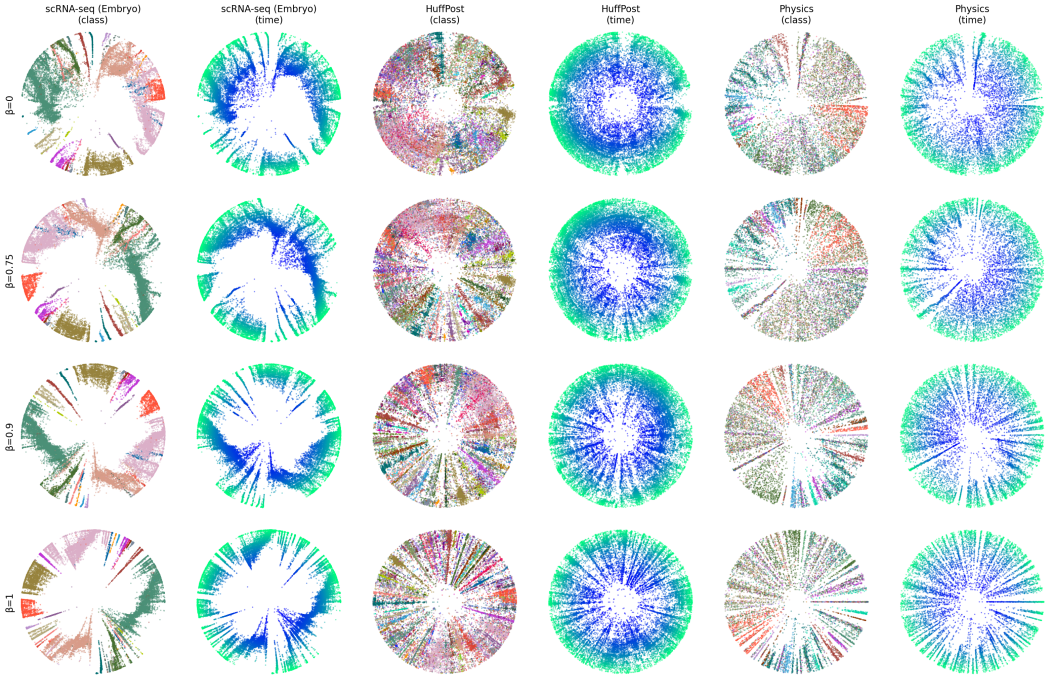

Figure I.22: Illustration of β values.

Table I.4: Synthetic tree metrics (mean $\pm$ SEM over replicates).

Method	SVR	GEO_w	GEO_b	GES	PTC	GM5
Unsupervised						
t-SNE	0.33 ± 0.04	0.94 ± 0.00	0.36 ± 0.05	0.45 ± 0.02	0.41 ± 0.10	0.46 ± 0.00
UMAP	0.13 ± 0.04	0.78 ± 0.01	0.15 ± 0.02	0.30 ± 0.01	0.31 ± 0.05	0.27 ± 0.00
PHATE	0.50 ± 0.05	0.85 ± 0.01	0.38 ± 0.04	0.33 ± 0.03	-0.06 ± 0.13	0.34 ± 0.00
Supervised						
RF-PHATE	0.16 ± 0.05	0.69 ± 0.03	0.38 ± 0.04	0.30 ± 0.02	0.35 ± 0.06	0.34 ± 0.00
Auxiliary variable (UMAP)						
UMAP ($\lambda = 0.1$)	0.03 ± 0.04	0.78 ± 0.01	0.10 ± 0.03	0.10 ± 0.02	0.17 ± 0.09	0.14 ± 0.00
UMAP ($\lambda = 0.3$)	0.11 ± 0.04	0.76 ± 0.01	0.12 ± 0.04	0.08 ± 0.01	0.15 ± 0.08	0.17 ± 0.00
UMAP ($\lambda = 1$)	0.14 ± 0.04	0.82 ± 0.00	0.15 ± 0.03	0.08 ± 0.03	0.28 ± 0.07	0.21 ± 0.00
UMAP ($\lambda = 3$)	0.16 ± 0.02	0.83 ± 0.01	0.20 ± 0.03	0.06 ± 0.02	0.23 ± 0.05	0.21 ± 0.00
UMAP ($\lambda = 10$)	0.16 ± 0.05	0.82 ± 0.02	0.21 ± 0.03	0.06 ± 0.02	0.21 ± 0.04	0.20 ± 0.00
Auxiliary variable (t-SNE)						
t-SNE ($\lambda = 0.1$)	0.30 ± 0.08	0.94 ± 0.00	0.31 ± 0.04	0.38 ± 0.04	0.30 ± 0.09	0.40 ± 0.00
t-SNE ($\lambda = 0.3$)	0.29 ± 0.07	$\mathbf{0.95} \pm 0.01$	0.35 ± 0.05	0.35 ± 0.04	0.25 ± 0.12	0.39 ± 0.00
t-SNE ($\lambda = 1$)	0.38 ± 0.06	0.94 ± 0.01	0.45 ± 0.08	0.26 ± 0.02	0.38 ± 0.04	0.44 ± 0.00
t-SNE ($\lambda = 3$)	0.44 ± 0.04	0.92 ± 0.00	0.44 ± 0.06	0.27 ± 0.01	0.46 ± 0.07	0.47 ± 0.00
t-SNE ($\lambda = 10$)	0.41 ± 0.08	0.91 ± 0.01	0.40 ± 0.08	0.10 ± 0.02	0.22 ± 0.11	0.32 ± 0.00
Auxiliary variable (PHATE)						
PHATE ($\lambda = 0.1$)	0.53 ± 0.05	0.83 ± 0.01	0.18 ± 0.05	0.39 ± 0.02	0.25 ± 0.11	0.38 ± 0.00
PHATE ($\lambda = 0.3$)	0.42 ± 0.05	0.81 ± 0.01	0.32 ± 0.08	0.32 ± 0.03	-0.17 ± 0.13	0.29 ± 0.00
PHATE ($\lambda = 1$)	0.41 ± 0.03	0.81 ± 0.02	0.30 ± 0.08	0.22 ± 0.03	-0.21 ± 0.14	0.25 ± 0.00
PHATE ($\lambda = 3$)	0.33 ± 0.07	0.77 ± 0.08	0.31 ± 0.09	0.13 ± 0.03	-0.03 ± 0.11	0.23 ± 0.00
PHATE ($\lambda = 10$)	0.37 ± 0.09	0.73 ± 0.09	0.48 ± 0.13	0.06 ± 0.02	-0.12 ± 0.11	0.19 ± 0.00
IRIS						
IRIS (β=1)	$\mathbf{0.93} \pm 0.00$	0.87 ± 0.01	0.43 ± 0.03	0.50 ± 0.02	0.65 ± 0.05	0.64 ± 0.00
IRIS (β=0)	$\mathbf{0.93} \pm 0.00$	0.88 ± 0.01	$\mathbf{0.68} \pm 0.03$	$\mathbf{0.54} \pm 0.02$	$\mathbf{0.69} \pm 0.03$	$\mathbf{0.73} \pm 0.00$

Table I.5: Metrics for real data (mean ± SEM over replicates).

Method	KNN	UTY	SVR	GM3
Embryo				
RF-PHATE	0.65 ± 0.00	0.03 ± 0.00	0.70 ± 0.00	0.25 ± 0.00
UMAP-aux ($\lambda = 3$)	0.78 ± 0.00	0.07 ± 0.00	0.56 ± 0.05	0.31 ± 0.00
t-SNE-aux ($\lambda = 0.3$)	$\mathbf{0.96} \pm 0.00$	0.19 ± 0.00	0.78 ± 0.00	0.53 ± 0.00
PHATE-aux ($\lambda = 0.3$)	0.49 ± 0.00	0.06 ± 0.00	0.95 ± 0.00	0.31 ± 0.00
IRIS	$\mathbf{0.96} \pm 0.00$	$\mathbf{0.25} \pm 0.00$	$\mathbf{0.99} \pm 0.00$	$\mathbf{0.62} \pm 0.00$
HuffPost				
RF-PHATE	0.07 ± 0.00	0.01 ± 0.00	0.09 ± 0.00	0.04 ± 0.00
UMAP-aux ($\lambda = 3$)	0.36 ± 0.00	0.02 ± 0.00	0.91 ± 0.02	0.19 ± 0.00
t-SNE-aux ($\lambda = 0.1$)	$\mathbf{0.75} \pm 0.00$	$\mathbf{0.05} \pm 0.00$	0.05 ± 0.00	0.12 ± 0.00
PHATE-aux ($\lambda = 0.3$)	0.34 ± 0.00	0.03 ± 0.00	0.66 ± 0.00	0.18 ± 0.00
IRIS	0.63 ± 0.00	0.04 ± 0.00	$\mathbf{0.99} \pm 0.00$	$\mathbf{0.30} \pm 0.00$
Physics				
RF-PHATE	0.28 ± 0.00	$\mathbf{0.93} \pm 0.00$	0.08 ± 0.00	0.27 ± 0.00
UMAP-aux ($\lambda = 3$)	0.32 ± 0.00	0.90 ± 0.00	0.65 ± 0.03	0.57 ± 0.00
t-SNE-aux ($\lambda = 0.1$)	$\mathbf{0.51} \pm 0.00$	0.91 ± 0.00	0.48 ± 0.01	0.60 ± 0.00
PHATE-aux ($\lambda = 0.3$)	0.32 ± 0.00	0.90 ± 0.00	0.25 ± 0.02	0.42 ± 0.00
IRIS	0.44 ± 0.00	0.91 ± 0.00	$\mathbf{0.99} \pm 0.00$	$\mathbf{0.74} \pm 0.00$