RHINONET: UNSUPERVISED LEARNING FOR PHENOSCAPING AND GENERATION OF HIGH-DIMENSIONAL POINT CLOUDS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present RhinoNet, a simplicial complex-based point cloud neural network, that creates a geometry-preserving embedding of high-dimensional point clouds. RhinoNet takes collections of datasets, also called data cohorts, and processes each dataset (point cloud) within the cohort as a simplicial complex. These simplicial complexes have datapoints as vertices and edges learned by Vietoris-Rips filtration. Using simplicial wavelets and MLP layers, RhinoNet computes a vector representation of each dataset in a latent space produced by an autoencoder. The latent space is optimized such that an optimal transport distance between pairs of point clouds is preserved between the latent representations. We apply RhinoNet to single-cell datasets collected on patients—a common source of such high-dimensional data. RhinoNet creates a patient level representation, known in biological contexts as a phenoscape. We establish that RhinoNet generates organized phenoscapes, whereas other methods such as static maps do not. Next we show that we can generate entire point clouds by sampling and decoding this phenoscape. We show two modes of generation via 1) latent flow matching and 2) geodesic interpolation in latent space, followed by a decoding back into the point cloud space. RhinoNet achieves state-of-the-art performance on single-cell and spatial data cohorts relative to tested baselines. Overall, RhinoNet lifts the concepts of representation learning and generative modeling to the point cloud or population level and defines a neural model for phenoscaping.

1 INTRODUCTION

With the advent of high-throughput, high-dimensional data, particularly in biology, a new class of problems has emerged for representation learning. Instead of the classic problem of learning representations of a set of points in a dataset, these problems involve learning representations of entire collections of point clouds, $\mathcal{C} = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(M)}\}$, in which each $\mathcal{X}^{(m)} = \{\mathbf{x}_i\}_{i=1}^{n_m} \subseteq \mathbb{R}^d$ is an unordered dataset of samples from a distribution. Situations where this arises includes single-cell data, or clinical data on cohorts of patients representing disease or conditions. To learn an unsupervised representation in this scenario, an encoder has to map an entire point cloud into a single latent vector, and the organization of the latent space has to reflect population level relationships between datasets, e.g., optimal transport distances. In this paper, we address this problem by proposing a representation learning and generative model called **RhinoNet**.

RhinoNet is a simplicial complex-based point cloud autoencoder that lifts representation learning and generation from points to point clouds (Figure 1). We encode point clouds as simplicial complexes, similar to the recently developed HiPoNet framework (Viswanath et al., 2025). However, in RhinoNet, we are interested in preserving geometric relationships between different point clouds as quantified by optimal transport distances, analogous to geometric autoencoders for datapoints (Sun et al., 2025). RhinoNet begins by taking wavelet transforms of node signals over the simplicial complex utilizing a geometric Hodge Laplacian and an associated diffusion operator in order to preserve more of the signal geometry. It learns its final embedding by further transforming the wavelet representation using an MLP that is penalized with a reconstruction penalty, along with a geodesic Sinkhorn penalty. This construction effectively creates geometry-preserving embeddings that are

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.

referred to as phenoscapes, where each point encodes to an entire dataset and distances between points are interpretable as optimal transport distances.

Further, RhinoNet is a generative model with two modes of generation. It can use latent flow matching to generate new point clouds from noise, and it can use geodesic interpolation in latent space to generate new point clouds which are thought of as lying in between two existing point clouds. For example, in a cancer cohort, the first method would generate new point clouds from a similar distribution and the latter would generate new point clouds in between two existing states.

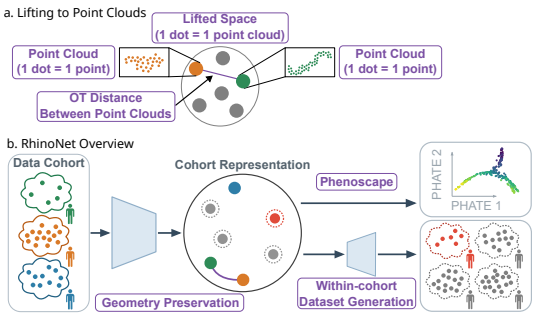


Figure 1: RhinoNet

We apply RhinoNet to the problem of phenoscaping cancer patients from single-cell proteomic and transcriptomic datasets collected on each patient. Our results show that RhinoNet most faithfully preserves optimal transport geometry between these patient datasets, and generates accurate point clouds on withheld points. Historically, most generative models for single-cell data do not generate whole point clouds from a latent space, but rather generate new points within a point cloud. Examples include scGen (Lotfollahi et al., 2019), CPA (Lotfollahi et al., 2023), and CellOT (Bunne et al., 2023), so we cannot compare to them. We compare RhinoNet to methods of phenoscaping and point cloud generation. A few methods can both phenoscape and generate whole point clouds. Meta Flow Matching (Atanackovic et al., 2025) generates a population from a matched control population, and Wasserstein Wormhole both embeds and decodes point clouds. We discuss additional related works in Appendix A.

Our main contributions include:

- **Geometry-preserving point cloud autoencoder:** We model each point cloud as a simplicial complex and embed it with wavelets built from a geometric Hodge Laplacian, which enables signal processing on the complex in a manner that captures the underlying geometry.
- **Geodesic Sinkhorn latent regularization:** A novel regularization that jointly enforces a geodesic sinkhorn distance called PhenoGS between point clouds in the latent space.
- **Generative model for point clouds:** We generate whole point clouds by sampling in the latent space. Because latent distances track optimal transport distances, interpolating or extrapolating between the embeddings of observed point clouds and decoding the result yields point clouds for unobserved states of a high-dimensional process.

2 Preliminaries

2.1 Simplicial Complexes and Simplicial Scattering

Simplicial complexes. Intuitively, an (abstract) simplicial complex is a higher-order graph that satisfies certain properties. In addition to edges (1-simplices) between two objects, a simplicial complex can use higher-order simplices to describe relationships between 3 or more objects. Simplicial complexes are often used to model hierarchical, relational, and geometric data.

Formally, we let $\mathcal{V} = \{v_0, v_1, \ldots, v_{N_0}\}$ be a finite set of nodes (also known as vertices). A k-simplex, sometimes called a k-th order simplex, $\sigma_k = \{v_{i_0}, v_{i_1}, \ldots, v_{i_k}\}$ is a subset of $\mathcal{V}$ with $k+1$ elements. We interpret 0-, 1-, and 2-simplices as nodes, edges, and triangles. A simplicial complex $\mathcal{S}$ is a finite collection of simplices closed under inclusion, meaning every non-empty subset of $\sigma_k \in \mathcal{S}$ is also in $\mathcal{S}$ (Schaub et al., 2020; Barbarossa & Sardellitti, 2020). We will let N_k denote the number of k-simplices in $\mathcal{S}$ and define the order K of the complex $\mathcal{S}$ to be the order of its largest simplex. We will assume that we are given D_k features for each k-simplex and will organize these features into a feature matrix $\mathbf{X}_k \in \mathbb{R}^{N_k \times D_k}$. We let $\mathbf{X} = \{\mathbf{X}_0, \ldots, \mathbf{X}_K\}$ denote the set of all feature matrices. We note that when $K = 1$, $\mathcal{S}$ is equivalent to an unweighted graph with node features and edge features stored in the matrices $\mathbf{X}_0$ and $\mathbf{X}_1$, respectively.

Boundary matrices. Connections between simplices of order k and $k-1$ are stored in the boundary matrix $\mathbf{B}_k \in \mathbb{R}^{N_{k-1} \times N_k}$, whose (i,j)-th entry is given by $(\mathbf{B}_k)_{i,j} = 1$ if the i-th $(k-1)$-simplex

is a subset of the j-th k-simplex and $(\mathbf{B}_k)_{i,j} = 0$ otherwise. We denote the set of all the boundary matrices by $\mathbf{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_K\}$. The combinatorial Hodge Laplacian is defined by $\Delta_k = \mathbf{L}_k^\ell + \mathbf{L}_k^u$, where $\mathbf{L}_k^\ell$ and $\mathbf{L}_k^u$ are the lower and upper Laplacians $\mathbf{L}_k^\ell = \mathbf{B}_k^\top \mathbf{B}_k$ and $\mathbf{L}_k^u = \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top$. By convention, $\mathbf{L}_0^\ell = \mathbf{L}_K^u = 0$. We observe that in the case $k = 0$, the combinatorial Hodge Laplacian Δ_0 is merely the standard combinatorial graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

Simplicial random walks and scattering transforms. We will extract multiscale information by letting the node features $\mathbf{X}$ diffuse (i.e., propagate) over $\mathcal{S}$. In each step of propagation, the features of a simplex σ_k are updated using the features of its neighbors. Notably, on a simplicial complex, each k-simplex σ_k can have four types of neighbors: its boundary adjacent neighbors $\mathcal{N}_\mathcal{B}(\sigma_k)$ are the $(k-1)$-simplices obtained by removing one node from σ_k; its coboundary adjacent neighbors $\mathcal{N}_\mathcal{C}(\sigma_k)$ are the $(k+1)$-simplices obtained by adding one node to σ_k. Its lower adjacent neighbors $\mathcal{N}_\mathcal{L}(\sigma_k)$ are the k-simplices that $\sigma_k' \in \mathcal{S}$ that share a common face, $|\sigma_k \cap \sigma_k'| = k$. Its upper adjacent neighbors $\mathcal{N}_\mathcal{U}(\sigma_k)$ are the k-simplices $\sigma_k' \in \mathcal{S}$ that are both faces of a common $k+1$ simplex, $\sigma_k \cup \sigma_k' \in \mathcal{S}$ and $|\sigma_k \cup \sigma_k'| = |\sigma_k| + 1 = k + 2$.

Thus, similar to Madhu et al. (2024), we will introduce five matrices, one corresponding to each of the four types of neighbors $\mathcal{C}, \mathcal{B}, \mathcal{L}$, and $\mathcal{U}$, and a fifth corresponding to self-loops:

$$\begin{gathered}\mathbf{P}_{k+1}^\mathcal{C} = \tfrac{1}{5}(\mathbf{D}_{k+1}^\mathcal{B})^{-1}\mathbf{B}_{k+1}, \qquad \mathbf{P}_k^\mathcal{B} = \tfrac{1}{5}\mathbf{B}_k^\top(\mathbf{D}_k^\mathcal{B})^{-1}, \qquad \mathbf{P}_k^\mathcal{I} = \tfrac{1}{5}\mathbf{I}_{N_k}, \\ \mathbf{P}_k^\mathcal{L} = \tfrac{1}{5}\mathbf{B}_k^\top\mathbf{B}_k(\mathbf{D}_k^\mathcal{L})^{-1}, \qquad \text{and} \qquad \mathbf{P}_k^\mathcal{U} = \tfrac{1}{5}\mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top(\mathbf{D}_k^\mathcal{U})^{-1},\end{gathered} \tag{1}$$

where $\mathbf{D}_k^\mathcal{B} = \operatorname{diag}(\mathbf{B}_k\mathbf{1})$, $\mathbf{D}_k^\mathcal{L} = \operatorname{diag}(\mathbf{B}_k^\top\mathbf{B}_k\mathbf{1})$, and $\mathbf{D}_k^\mathcal{U} = \operatorname{diag}(\mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top\mathbf{1})$ are degree matrices corresponding to different types of neighbors. We note that the normalizing constant $\frac{1}{5}$ is meant to highlight the connection between these matrices and the transition probabilities of a lazy random walker on $\mathcal{S}$. We define the diffused node features recursively by first computing

$$\begin{gathered}\mathbf{X}_{\mathcal{B},k}^{(j)} = \mathbf{P}_k^\mathcal{B}\mathbf{X}_{k-1}^{(j-1)}, \qquad \mathbf{X}_{\mathcal{C},k}^{(j)} = \mathbf{P}_{k+1}^\mathcal{C}\mathbf{X}_{k+1}^{(j-1)}, \qquad \mathbf{X}_{\mathcal{I},k}^{(j)} = \mathbf{P}_k^\mathcal{I}\mathbf{X}_k^{(j-1)}, \\ \mathbf{X}_{\mathcal{L},k}^{(j)} = \mathbf{P}_k^\mathcal{L}\mathbf{X}_k^{(j-1)}, \qquad \text{and} \qquad \mathbf{X}_{\mathcal{U},k}^{(j)} = \mathbf{P}_k^\mathcal{U}\mathbf{X}_k^{(j-1)},\end{gathered} \tag{2}$$

and then using a dimension-preserving aggregation:

$$\mathbf{X}_k^{(j)} = \text{AGGR1}\left(\mathbf{X}_{\mathcal{I},k}^{(j)}, \mathbf{X}_{\mathcal{B},k}^{(j)}, \mathbf{X}_{\mathcal{C},k}^{(j)}, \mathbf{X}_{\mathcal{L},k}^{(j)}, \mathbf{X}_{\mathcal{U},k}^{(j)}\right), \tag{3}$$

where $\mathbf{X}_k^{(0)} = \mathbf{X}_k$ and, in our implementation, we take AGGR1 to be a simple summation.

These features $\mathbf{X}_k^{(j)}$ can be interpreted as local averages of the original node features at scale j (or as low-pass filterings from the perspective of graph signal processing (GSP), see e.g., Ortega et al. (2018)). To track changes across scales, we consider wavelets of the form $\psi_j^{k,\mathcal{N}}\mathbf{X} = \mathbf{X}_{\mathcal{N},k}^{(j-1)} - \mathbf{X}_{\mathcal{N},k}^{(j)}$ for each $\mathcal{N} \in \{\mathcal{I}, \mathcal{B}, \mathcal{C}, \mathcal{L}, \mathcal{U}\}$ (band-pass filtering from the GSP perspective), as developed in Madhu et al. (2024). We then aggregate the wavelet coefficients from different neighbor types to compute simplicial coefficients at scale j as $|\Psi_j^k\mathbf{X}| = |\text{AGGR1}(\psi_j^{k,\mathcal{SL}}, \psi_j^{k,\mathcal{B}}, \psi_j^{k,\mathcal{C}}, \psi_j^{k,\mathcal{L}}, \psi_j^{k,\mathcal{U}})|$ by applying the modulus (absolute value). We denote the set of embeddings of all orders $|\Psi_j\mathbf{X}| = \{|\Psi_j^0\mathbf{X}|, |\Psi_j^1\mathbf{X}|, \ldots, |\Psi_j^K\mathbf{X}|\}$, which has the same form as $\mathbf{X}$. Applying the same wavelet operators on $|\Psi_j\mathbf{X}|$ instead of $\mathbf{X}$ computes the deep simplicial scattering transform $|\Psi_{j'}||\Psi_j\mathbf{X}|| = \left\{|\Psi_{j'}^0||\Psi_j\mathbf{X}||, \ldots, |\Psi_{j'}^K||\Psi_j\mathbf{X}||\right\}$ where $1 \le j < j' \le J$. Finally, these coefficients can be combined to compute the embedding of k-simplices $S_k(\mathbf{X}, \mathbf{B}) = \text{COMBINE}(\mathbf{X}_k, |\Psi_j^k\mathbf{X}|, |\Psi_{j'}||\Psi_j\mathbf{X}||, \ldots)$ where COMBINE is the concatenation function and the set of embeddings of all orders can be denoted as $S(\mathbf{X}, \mathbf{B}) = \{S_k(\mathbf{X}, \mathbf{B}), \ldots, S_K(\mathbf{X}, \mathbf{B})\}$.

Importantly, we observe that every operator here is assembled directly from the matrices $\mathbf{B}_k$ whose entries are all either 0 or 1. This means that these simplicial scattering coefficients are only able to capture information from the incidence structure, and are unable to capture other types of geometric information. Our methods introduced in Section 3.1 will address this limitation allowing us to obtain a more powerful representation of the underlying data geometry.

Simplicial complexes derived from point clouds. Our data consists of point clouds, each of the form $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$. We use Vietoris-Rips filtration to simplicial complexes from these

point clouds using the. As per Viswanath et al. (2025), we first learn weight vectors $\alpha_\nu \in \mathbb{R}^d$, $1 \leq \nu \leq V$, and rescale the clouds to give different views $\tilde{\mathcal{X}}_\nu^{(m)} = \{\alpha_\nu \odot \mathbf{x}_i\}_{i=1}^n$ that emphasize different subsets of features, e.g., different cellular processes in single-cell data. Then, we build a Vietoris–Rips complex $\mathcal{S}^\nu$ on each view through a differentiable filtration, yielding boundary matrices $\tilde{\mathbf{B}}^\nu$ and simplex features $\tilde{\mathcal{X}}_\nu^{(m)}$. Finally, we apply the simplicial scattering transform discussed above to each complex and aggregate to obtain $\Phi = \text{AGGR2}\{S(\tilde{\mathbf{X}}^{(1)}, \tilde{\mathbf{B}}^{(1)}), \ldots, S(\tilde{\mathbf{X}}^{(V)}, \tilde{\mathbf{B}}^{(V)})\}$, where AGGR2 is concatenation across views.

2.2 Smooth Hodge Laplacian

In the smooth setting, the Hodge Laplacian depends on the Riemannian metric g. This motivates our method, which we will introduce below, where we use mass matrices $\mathbf{M}_k$ to approximate this metric. We will then use the $\mathbf{M}_k$ to compute a geometric (discrete) Hodge Laplacian.

Formally, for an oriented n-dimensional Riemannian manifold $(\mathcal{M}^n, g)$, we denote the (metric-free) exterior derivative on the space of smooth differential k-forms by $d : \Omega^{k-1} \to \Omega^k$, and denote its L^2 adjoint, the co-differential, $\delta : \Omega^k \to \Omega^{k-1}$. The smooth Hodge Laplacian is then defined by $\Delta_k^{\text{smooth}} = d\delta + \delta d$ (note the similarity with the combinatorial Hodge Laplacian for simplicial complexes introduced in Section 2.1, $\Delta_k = \mathbf{B}_k^\top \mathbf{B}_k + \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top$).

By the definition of the adjoint, δ satisfies $\langle d\eta, \omega \rangle_{L^2(\Omega^k)} = \langle \eta, \delta\omega \rangle_{L^2(\Omega^{k-1})}$ for all $\eta \in \Omega^{k-1}, \omega \in \Omega^k$, where, the L^2 inner products (and hence δ), depends on the metric g. In order to compute these inner products, we will use the Hodge star map $\star_g : \Omega^k \to \Omega^{n-k}$, which is characterized by $\omega \wedge \star_g \eta = \langle \omega, \eta \rangle_g \, d\text{vol}_g$, where $\langle \cdot, \cdot \rangle_g$ denotes the metric-induced pointwise inner product on k-forms. This allows us to calculate L^2 inner products via $\langle \omega, \eta \rangle_{L^2} = \int_{\mathcal{M}} \omega \wedge \star_g \eta$. We also can use $\star_g$ to characterize the co-differential via $\delta = (-1)^{k(n+1)+1} \star_g d \star_g$.

2.3 The Geometric Hodge Laplacian

As in Viswanath et al. (2025), our datasets will consist of point clouds of the form $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$, and we will build simplicial complexes and subsequently Hodge Laplacians from these point clouds. However, we will use methods based on a geometric Hodge Laplacian $\widetilde{\Delta}_k$, taking inspiration from Discrete Exterior Calculus (DEC) Desbrun et al. (2005) and Finite Element Exterior Calculus (FEEC) (Arnold, 2018). This is in contrast to Viswanath et al. (2025) which use methods based on the combinatorial Hodge Laplacian Δ_k (see Section 2.1). This change is motivated by the observation that Δ_k does not depend on any metric and therefore does not in general converge to the smooth Hodge Laplacian as $n \to \infty$.

To construct the discrete geometric Hodge Laplacian, we let C^k denote the set of functions defined on the k-simplices in $\mathcal{S}$. We will identify each $\mathbf{x} \in C^k$ with a vector of length N_k and interpret it as a k-cochain, i.e., as the discrete integrals of a k-form over k-dimensional (geometric) simplices. The coboundary matrix $\mathbf{B}_k^\top$ can now be interpreted as a map $\mathbf{B}_k^\top : C^{k-1} \to C^k$ that plays the role analogous to the differential d in the smooth setting. As the discrete counterpart of the codifferential δ, we will consider the adjoint $\left(\mathbf{B}_k^\top\right)^* = \mathbf{M}_{k-1}^{-1}\mathbf{B}_k\mathbf{M}_k$ with respect to the requisite inner product $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{M}_k} = \mathbf{x}^\top \mathbf{M}_k \mathbf{y}$, where $\mathbf{M}_k$ is a symmetric positive definite matrix to be specified below. We now define the geometric Hodge Laplacian by

$$\widetilde{\Delta}_k = \mathbf{B}_k^\top (\mathbf{B}_k^\top)^* + (\mathbf{B}_{k+1}^\top)^* \mathbf{B}_{k+1}^\top = \mathbf{B}_k^\top \, \mathbf{M}_{k-1}^{-1} \, \mathbf{B}_k \, \mathbf{M}_k + \mathbf{M}_k^{-1} \, \mathbf{B}_{k+1} \, \mathbf{M}_{k+1} \, \mathbf{B}_{k+1}^\top$$

mirroring the definition $\Delta_k^{\text{smooth}} = d\delta + \delta d$ from Section 2.2. We observe that, by construction, $\widetilde{\Delta}_k$ is self-adjoint with respect to the inner product $\langle \cdot, \cdot \rangle_{\mathbf{M}_k}$ and that it reduces to the combinatorial Hodge Laplacian if $\mathbf{M}_k = \mathbf{I}$.

We will choose the matrix $\mathbf{M}_k \in \mathbb{R}^{N_k \times N_k}$ to be a diagonal mass matrix inspired by DEC with entries corresponding to the volumes of the associated simplices. Recent work has proven that the DEC Hodge Laplacian with suitable mass matrices converge to the smooth Hodge Laplacian under mesh refinement when the simplicial mesh and underlying manifold satisfy centering or regularity conditions (Zhu et al., 2025; Guzmán & Potu, 2025). These results connect the DEC constructions to well-established FEEC convergence results (Arnold et al., 2010; Arnold, 2018). Numerical experiments have also shown that the DEC Hodge Laplacian constructions converge to the smooth

Hodge Laplacian, even for simplicial meshes that do not satisfy regularity conditions (Mohamed et al., 2016; 2018). These results help motivate our use of a geometric Hodge Laplacian approach to encode metric information.

3 METHODS

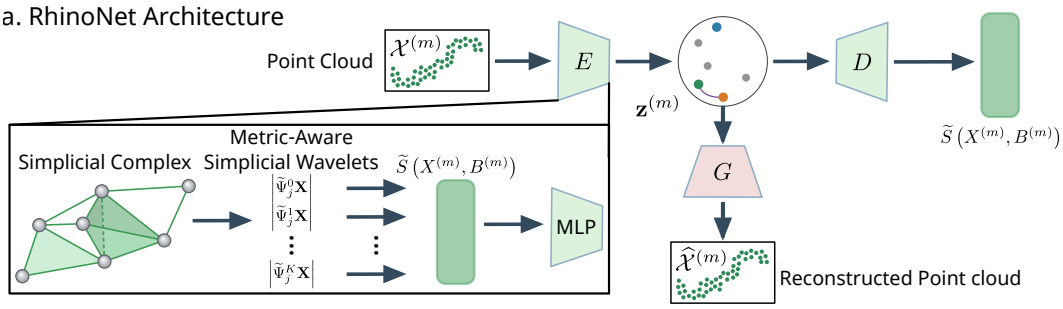


Figure 2: The RhinoNet autoencoder builds a simplicial complex for each point cloud, computes metric-aware simplicial wavelets, and embeds them with an MLP to a latent space. The generation head G is trained to reconstruct point clouds.

This section describes **RhinoNet** in two parts: We first introduce an autoencoder that maps each point cloud to a latent vector using a geometric Hodge Laplacian-based simplicial scattering transform. The latent space is regularized by a geodesic Sinkhorn distance between point clouds. We then build on this latent space to generate new point clouds, through latent flow matching and through interpolation. The architecture is depicted in Figure 2.

3.1 GEOMETRIC HODGE LAPLACIAN-BASED SIMPLICIAL WAVELET AUTOENCODER

Here we describe the **RhinoNet** autoencoder, whose encoder maps a point cloud $\mathcal{X}^{(m)}$ to a multiscale geometry-aware latent vector $\mathbf{z}^{(m)}$. First, we construct a simplicial complex $\mathcal{S}^{(m)}$ from the point cloud via a Vietoris-Rips filtration. Then, we apply the geometric Hodge Laplacian-based simplicial scattering transform, described below, over this structure to obtain a multiscale representation $\Phi^{(m)}$, which an MLP maps to the latent vector $\mathbf{z}^{(m)}$. The latent space is regularized to enforce a geodesic Sinkhorn distance between point clouds. The decoder then reconstructs $\Phi^{(m)}$ from $\mathbf{z}^{(m)}$.

Geometric Hodge Laplacian-based simplicial scattering. We introduce a novel form of the simplicial scattering transform based on the geometric Hodge Laplacian $\widetilde{\Delta}_k$ (see Section 2.3) building on previous work (Madhu et al., 2024; Viswanath et al., 2025) which implicitly used the combinatorial Hodge Laplacian Δ_k.

In our experiments, we will use simplices of order at most two. Therefore, we need to specify volumes for 0-, 1-, and 2-simplices in order to define the diagonal mass matrices $\mathbf{M}_0$, $\mathbf{M}_1$, and $\mathbf{M}_2$. We give each 0-simplex (i.e., vertex) the uniform volume $\frac{1}{N_0}$. We define the volume of a 1-simplex (edge) between $\mathbf{x}_i$ and $\mathbf{x}_j$ to be its (Euclidean) length $\|\mathbf{x}_i - \mathbf{x}_j\|_2$, and let the volume of each 2-simplex, σ_ℓ^2, be the area of the triangle it spans, $|\sigma_\ell^2|$. For higher order simplices, Cayley-Menger volumes can be used. With these choices, we obtain $\mathbf{M}_0 = \frac{1}{N_0}\mathbf{I}$, $(\mathbf{M}_1)_{\ell,\ell} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ when σ_ℓ^1 is the edge between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $(\mathbf{M}_2)_{\ell,\ell} = |\sigma_\ell^2|$, when σ_ℓ^2 is a 2-simplex.

We modify the transition matrices introduced in Section 2.1 to incorporate the mass matrices $\mathbf{M}_k$. This yields metric-aware transition matrices:

$$\begin{aligned} \widetilde{\mathbf{P}}_{k+1}^{\mathcal{C}} &= \tfrac{1}{5}(\widetilde{\mathbf{D}}_{k+1}^{\mathcal{B}})^{-1}\mathbf{M}_{k-1}^{-1}\mathbf{B}_{k+1}\mathbf{M}_k, \qquad \widetilde{\mathbf{P}}_k^{\mathcal{B}} = \tfrac{1}{5}\mathbf{B}_k^\top(\widetilde{\mathbf{D}}_k^{\mathcal{B}})^{-1}, \qquad \widetilde{\mathbf{P}}_k^{\mathcal{I}} = \tfrac{1}{5}\mathbf{I}_{N_k}, \\ \widetilde{\mathbf{P}}_k^{\mathcal{L}} &= \tfrac{1}{5}\mathbf{B}_k^\top\mathbf{M}_{k-1}^{-1}\mathbf{B}_k\mathbf{M}_k(\widetilde{\mathbf{D}}_k^{\mathcal{L}})^{-1}, \qquad \widetilde{\mathbf{P}}_k^{\mathcal{U}} = \tfrac{1}{5}\mathbf{M}_k^{-1}\mathbf{B}_{k+1}\mathbf{M}_{k+1}\mathbf{B}_{k+1}^\top\left(\widetilde{\mathbf{D}}_k^{\mathcal{U}}\right)^{-1}, \end{aligned} \tag{4}$$

with $\widetilde{\mathbf{D}}_{k+1}^{\mathcal{B}} = \mathrm{diag}(\mathbf{M}_{k-1}^{-1}\mathbf{B}_{k+1}\mathbf{M}_k\mathbf{1})$, $\widetilde{\mathbf{D}}_k^{\mathcal{B}} = \mathrm{diag}(\mathbf{B}_k^\top\mathbf{1})$, $\widetilde{\mathbf{D}}_k^{\mathcal{L}} = \mathrm{diag}(\mathbf{B}_k^\top\mathbf{M}_{k-1}^{-1}\mathbf{B}_k\mathbf{M}_k\mathbf{1})$, and $\widetilde{\mathbf{D}}_k^{\mathcal{U}} = \mathrm{diag}(\mathbf{M}_k^{-1}\mathbf{B}_{k+1}\mathbf{M}_{k+1}\mathbf{B}_{k+1}^\top\mathbf{1})$. We use these transition matrices to compute $\Phi^{(m)} = \widetilde{S}(\mathbf{X}^{(m)}, \mathbf{B}^{(m)})$, where $\widetilde{S}$ is the wavelet transform defined in Section 2.1 with each $\mathbf{P}_k^{\mathcal{N}}$ replaced by $\widetilde{\mathbf{P}}_k^{\mathcal{N}}$.

Geodesic Sinkhorn regularization. The multiscale representation described above is an initial vector encoding of the entire point cloud. We then use an MLP to further encode $\Phi^{(m)}$ (the vector encoding of the m-th point cloud) in a latent space as $\mathbf{z}^{(m)} = \text{MLP}(\Phi^{(m)}) = E(\mathcal{X}^{(m)})$. In order to produce a structured, low-dimensional representation, we regularize the RhinoNet encoder so that the Euclidean distance between latent points $\mathbf{z}^{(m)}$ and $\mathbf{z}^{(m')}$ matches the PhenoGS (Wilkinson et al., 2026) distance W between the point clouds $\mathcal{X}^{(m)}$ and $\mathcal{X}^{(m')}$, which uses the geodesic Sinkhorn distance. To this effect, we minimize:

$$\mathcal{L}_{GS} = \sum_{m,m'} \left(\|\mathbf{z}^{(m)} - \mathbf{z}^{(m')}\|_2 - W(\mathcal{X}^{(m)}, \mathcal{X}^{(m')}) \right)^2.$$

We pair the RhinoNet encoder with a decoder D_θ, chosen to be an MLP, that reconstructs the wavelet representation Φ from the latent $\mathbf{z}^{(m)}$. Our decoder is penalized by the reconstruction loss in order to preserve structural information in the latent space: $\mathcal{L}_{rec} = \|D_\theta(\mathbf{z}^{(m)}) - \Phi\|_2^2$. The encoder and decoder are trained jointly by minimizing $\mathcal{L}_{AE} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{GS}$, where λ is a hyperparameter, so that the resulting latent space preserves structural information and the optimal transport geometry between the underlying point clouds. Such an organization of the latent space enables the generation of high-dimensional point clouds, which we describe in the next section.

3.2 Data Generation

Generation head. Here we describe the generation of high-dimensional point clouds from the learned latent space. After training the encoder E to map each point cloud $\mathcal{X}^{(m)}$ to a latent $\mathbf{z}^{(m)} = E(\mathcal{X}^{(m)}) \in \mathbb{R}^D$, we consider two modes of generation, as shown in Figure 3: sampling new point clouds via latent flow matching, and interpolating between conditions along the latent space. In both cases, a generation head $G_\theta : \mathbb{R}^D \to \mathbb{R}^d$ maps the latent point $\mathbf{z}^{(m)}$ to a point cloud $\hat{\mathcal{X}}^{(m)} = G_\theta(\mathbf{z}^{(m)})$. We train the generation head by minimizing the Earth mover's distance between the generated point cloud $\hat{\mathcal{X}}^{(m)}$ and the true point cloud $\mathcal{X}^{(m)}$, approximated with entropic optimal transport $\mathcal{W}_\epsilon$ (Benamou et al., 2015):

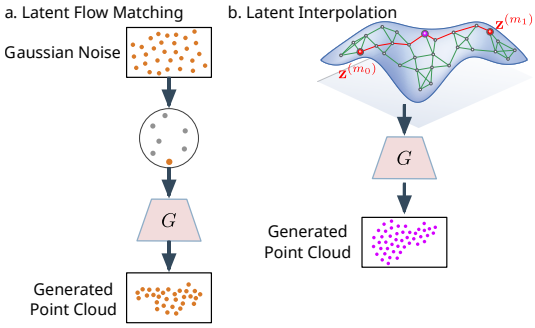


Figure 3: **Point cloud generation with RhinoNet.** (a) Using latent flow matching from noise. (b) Interpolating in latent space along discrete geodesics built from a k-nearest neighbors graph.

$$\mathcal{L}_{gen} = \mathcal{W}_\epsilon(\hat{\mathcal{X}}^{(m)}, \mathcal{X}^{(m)}),$$

where ϵ is the entropic regularization strength of the Sinkhorn approximation.

Latent flow matching. To generate new point clouds conditioned on a set of labels (e.g., patient, treatment or recurrence), we adopt flow matching (Lipman et al., 2023), which learns a velocity field that transports a standard Gaussian source to the distribution of the observed latents for a given combination of labels and is simulation-free during training. Conditioning the velocity field on the labels allows us to sample latents for combinations that were never directly observed during training. At inference time, we integrate the learned velocity starting from Gaussian noise to obtain a sampled latent for a target combination of labels, which we then decode into a point cloud through the generation head G_θ.

Interpolation. We now consider the setting where want to be able to move between two points in the latent space, which may, for example, correspond to two different conditions in a patient. We consider a path between the two latent points and interpret points along the path as intermediate states so that the latent space forms a continuous phenoscape. Since the true trajectory between two conditions need not be straight in latent space, we construct the path as a discrete geodesic over the graph of observed latent points. Recall that the encoder is trained so that Euclidean distances between latent points approximate optimal transport distances between the corresponding point clouds. The shortest path between two latent points should therefore approximate the shortest path between their corresponding point clouds under optimal transport. We build a symmetrized k-nearest neighbor graph over all the N latents $\{\mathbf{z}^{(m)}\}_{m=1}^M$, where each vertex corresponds to an observed point cloud $\mathcal{X}^{(m)}$. The graph distance between two point clouds is then the shortest path between them

in this graph computed via Dijsktra's algorithm (Dijkstra, 1959).We select two point clouds $\mathcal{X}^{(m_0)}$ and $\mathcal{X}^{(m_1)}$, drawn from two different conditions to serve as the path endpoints. Let $c(m)$ denote a label of the point cloud $\mathcal{X}^{(m)}$, so that $c(m_0) \neq c(m_1)$. The discrete geodesic between $\mathbf{z}^{(m_0)}$ and $\mathbf{z}^{(m_1)}$ then gives a sequence of point clouds connecting the two conditions, with interior point clouds serving as intermediate states. To evaluate whether these intermediate point clouds capture meaningful structure, we hold them out and evaluate whether the decoder can reconstruct each one from its corresponding latent.

RhinoNet is trained in two stages. We first train the autoencoder by minimizing $\mathcal{L}_{AE}$ to obtain a latent space organized by geodesic Sinkhorn distance between point clouds. We then train the generation head G_θ and the components used for latent flow matching and interpolation on top of this latent space. Computational complexity of the method is discussed in Appendix E.

4 Results

In this section, we describe the datasets used to train RhinoNet, the tasks we evaluate on, and the baselines we compare against, followed by results and ablation studies. Because our latent space is organized geometrically, we evaluate it using metrics that directly compare point cloud geometry. For the interpolation task, where a ground-truth intermediate point cloud is available, we measure the Earth Mover's Distance (EMD) between the generated point cloud and this ground truth. For the latent flow matching task, where new point clouds are generated from noise, we instead compare the distribution of generated point clouds to the distribution of real point clouds using 1-Wasserstein distance ($\mathcal{W}_1$) a.k.a, the Earth mover's distance, 2-Wasserstein distance ($\mathcal{W}_2$), and Maximum Mean Discrepancy (MMD). Further details on the experimental setup are provided in Appendix D.

Data cohorts. We evaluate RhinoNet on a synthetic Swiss-roll dataset and on two real-world biological data cohorts. First, we use a 10-dimensional synthetic Swiss-roll with a known ground truth manifold structure as a controlled setting for evaluating neighborhood preservation in the learned latent space. We then evaluate RhinoNet on two real-world biological data cohorts: the first is a set of patient-derived organoids (PDOs) (Ramos Zapatero et al., 2023), consisting of organoid cultures grown from colorectal cancer patients, with each culture represented as a point cloud of cells characterized by a panel of 44 protein markers; the second is CODEX-based spatial transcriptomics data (Wu et al., 2022), spanning multiple patient cohorts including Charville and UPMC with head-and-neck or colorectal cancer, where each cell's protein expression is measured across 40 markers along with its spatial position in the tissue. Further details on the data cohorts, including cohort sizes, marker panels, and preprocessing are provided in Appendix B.

Baselines. We evaluate RhinoNet against a set of baselines for each task. For the phenoscape evaluation, we compare against DiffusionEMD (Tong et al., 2021), PointNet++ (Qi et al., 2017b), PointTransformer (Wu et al., 2024), and Meta Flow Matching (Atanackovic et al., 2025). For the generation of single-cell data using latent flow matching evaluation, we compare against Point-Voxel Diffusion (PVD) (Zhou et al., 2021), PointFlow (Yang et al., 2019), and LION (Zeng et al., 2022). For the interpolation task, we compare against Meta Flow Matching (Atanackovic et al., 2025).

RhinoNet can preserve the manifold geometry. We first evaluate how well the geometry learned by RhinoNet captures the structure of a manifold, comparing against DiffusionEMD, PointNet++, and PointTransformer. Each method maps a point cloud to a single latent vector, which we reduce to two dimensions with PHATE (Moon et al., 2019), so that every dot in Figure 4 is one entire point cloud, colored by a known property: the generating sheet coordinates for the Swiss roll, and sample metadata or mean marker expression for the biological cohorts. The Swiss roll cohort consists of 1,000 point clouds whose centers lie on a two-dimensional sheet rolled up in three dimensions and embedded in $\mathbb{R}^{10}$, with ground-truth distance measured along the unrolled sheet. RhinoNet recovers a single band ordered by position along the roll and retains height as a second axis. PointNet++ folds the roll onto itself, placing point clouds

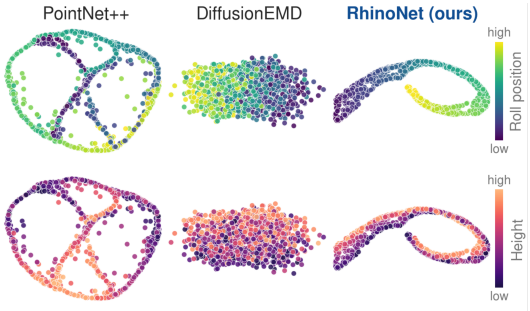


Figure 4: **RhinoNet preserves 10D Swiss roll geometry.** 2D PHATE of each method's latent space, one dot per point cloud, colored by position along the roll (top) and height (bottom).

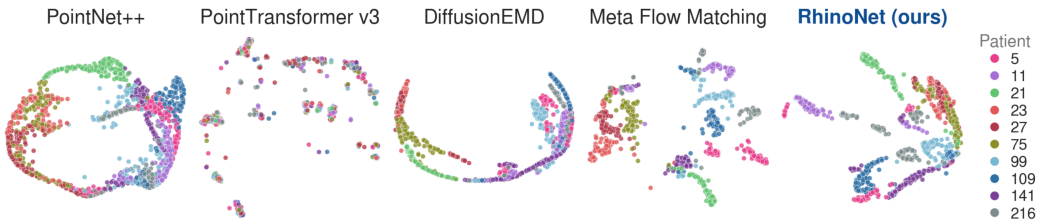


Figure 5: **RhinoNet organizes patient-derived organoids by patient.** Phenoscapes of 1678 colorectal cancer organoid samples from 10 patients (2M cells, 44 protein markers). Each dot is one dataset, visualized using PHATE of that method's latent space and colored by patient of origin.

from distant parts of the sheet side by side, and DiffusionEMD keeps only a coarse ordering with no local structure. Neither PointNet++ nor DiffusionEMD preserves height.

Phenoscape. We next construct phenoscapes of the patient-derived organoid PDO cohort (Ramos Zapatero et al., 2023). Each sample is a point cloud of roughly two thousand cells measured across patients. Here, we ask whether the phenoscape recovers the dominant source of biological variation: the patient of origin. Figure 5 shows that RhinoNet places most patients on distinct branches of the phenoscape, including held-out patients, which were embedded without being seen during training. PointNet++ arranges samples along a single ring on which several patients overlap. DiffusionEMD orders samples along one curve and only partially separates patients. PointTransformer collapses the cohort into a few isolated clusters that mix patients. While MFM separeted the patients well, it places each patient in a compact cluster, whereas RhinoNet arranges each patient's samples along a continuous branch. This difference likely reflects that Meta Flow Matching trains its point cloud embedding to condition a flow, with no term relating distances between embeddings to distances between point clouds. Together with the Swiss roll results, this shows that the geodesic Sinkhorn regularisation produces a latent space whose large-scale organisation reflects the geometric structure of the cohort. We show a similar phenoscape for Charville in Figure A1 and UPMC A2 colored by the gene expression levels showing a gradient in expression level across the phenoscape while other methods fail to encode this kind of structure.

Generating single-cell datasets using Latent Flow Matching. We next assess RhinoNet's ability to generate single-cell point clouds using the latent flow-matching procedure described in Section 3.2. For the PDO data cohort, generation is conditioned on patient identity and treatment, enabling the model to generate a cellular point cloud associated with a specified patient–treatment combination. For the Charville and UPMC cohorts, generation is conditioned on patient identity and observed recurrence status. As shown in Table 1, RhinoNet achieves the lowest values for all three metrics on PDO and Charville. Overall, these results indicate that RhinoNet produces point clouds with strong marker-space and distributional agreement across the three cohorts, while PointFlow remains competitive on UPMC.

Table 1: Comparison of generated point clouds across the PDO, Charville, and UPMC data cohorts using 1-Wasserstein distance, 2-Wasserstein distance, and Maximum Mean Discrepancy (MMD) between the generated and real point cloud distributions. Lower values indicate better generation quality. Bolded score indicates best overall performance, and an underlined score indicates second-best performance.

Model	PDO			Charville			UPMC		
	EMD ($\mathcal{W}_1$) ↓	$\mathcal{W}_2$ ↓	MMD ↓	EMD ($\mathcal{W}_1$) ↓	$\mathcal{W}_2$ ↓	MMD ↓	EMD ($\mathcal{W}_1$) ↓	$\mathcal{W}_2$ ↓	MMD ↓
PVD	36.8437	119.8368	0.249434	35.3825	50.1669	0.286170	12.5308	12.6588	0.263725
PointFlow	<u>6.1260</u>	<u>6.2154</u>	<u>0.024944</u>	<u>4.7665</u>	<u>4.9663</u>	<u>0.037828</u>	**2.8926**	<u>3.2072</u>	**0.014311**
LION	66.3286	259.9126	0.198157	240.1520	806.4914	0.264518	26.7287	35.8295	0.371724
RhinoNet	**4.2305**	**4.3241**	**0.021284**	**3.7432**	**3.9447**	**0.025637**	<u>2.9567</u>	**3.1816**	<u>0.027461</u>

Interpolation. Finally, we evaluate RhinoNet's ability to interpolate between point clouds in its learned latent space. For each cohort we construct a discrete geodesic between two endpoint point clouds (Section 3.2), hold out the intermediate point clouds on the path from decoder training, and reconstruct them from their latent codes. On PDO, the path runs within a single patient from a control (DMSO) culture to a drug-treated culture (treatment F), with three intermediate cultures held out. On Charville and UPMC, the path runs from a non-recurrent to a recurrent sample of the same patient, testing whether the latent space captures a continuous transition associated with disease recurrence; two intermediate samples are held out on Charville and three on UPMC, where

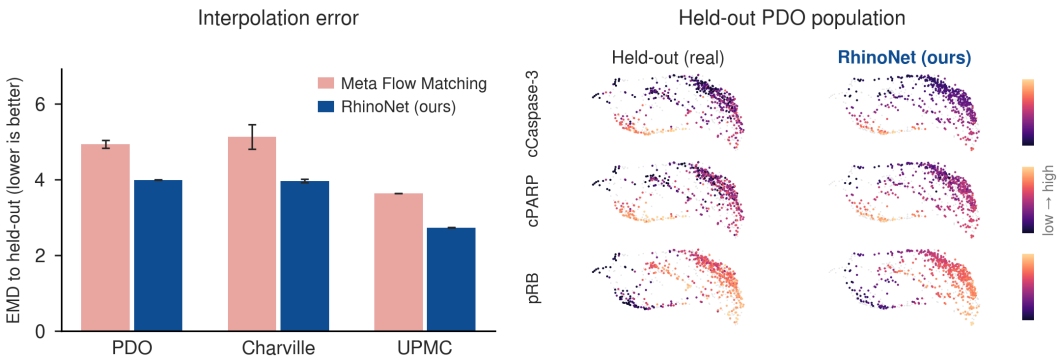


Figure 6: **Interpolating held-out point clouds. Left:** Earth Mover's distance (EMD) between generated and held-out point clouds (lower is better). Bars show the mean ± s.d. over three seeds of the average EMD across each dataset's held-out point clouds. **Right:** a held-out PDO point cloud from patient 21, lying on the phenoscape path between DMSO and treatment 5-FU. Real cells (left) and cells generated by RhinoNet (right) are embedded jointly with PHATE and colored by cleaved caspase-3, cleaved PARP and phosphorylated RB. Each point in the plot is a cell.

the intermediate samples come from three patients. Across all three cohorts, RhinoNet reconstructs held-out intermediate point clouds more accurately than Meta Flow Matching, Figure 6, left), reducing EMD by 19% on PDO (3.98 vs. 4.93), 23% on Charville (3.96 vs. 5.13) and 25% on UPMC (2.73 vs. 3.63). RhinoNet is also more stable across seeds (s.d. ≤ 0.05 vs. up to 0.32 for Meta Flow Matching). Beyond matching the overall distribution, the generated cells recover how signaling varies within a held-out organoid point cloud: apoptosis markers (cleaved caspase-3, cleaved PARP) and a proliferation marker (pRB) follow the same gradients across the PHATE embedding as in the real cells (Figure 6, right). Figure A4 (Appendix F.3) shows each path in RhinoNet's phenoscape. In all three cohorts, the geodesic between the two conditions passes through a short run of neighboring point clouds rather than a single long jump. The held-out point clouds are therefore intermediate states between the endpoints, and RhinoNet decodes each one from its latent position on the path.

Ablations. To isolate the effect of the geometric Hodge Laplacian, we retrain RhinoNet on Charville without using the mass matrices, which corresponds to use the combinatorial Laplacian in place of the geometric Laplacian. As we can see in Figure 7, the full version of RhinoNet based on the geometric Laplacian fits the geodesic Sinkhorn distances well, and gives smoother marker gradients across the phenoscape compared to the combinatorial version. We present a detailed description and more markers in Appendix Figure A3. Furthermore, increasing the simplex order from $K = 1$ (a graph over cells) to $K = 2$ (adding triangles) improves the fit to the geodesic Sinkhorn distances on both spatial cohorts. On UPMC it also orders tissue regions along a smoother axis of immune infiltration (Figure A5a), while on Charville both orders recover the immune gradient (Figure A5b).

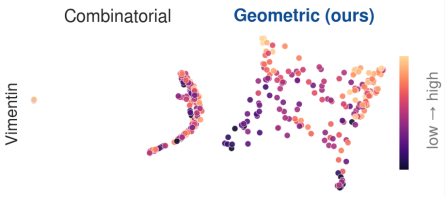


Figure 7: **Hodge Laplacian ablation.** Charville phenoscapes from RhinoNet with the combinatorial or geometric Hodge Laplacian, colored by mean Vimentin expression per region.

5 Conclusion

In this work, we present RhinoNet, a novel simplicial complex based point cloud neural network that preserves the geometry of the underlying data. RhinoNet is trained with an autoencoder such that each point in the latent space is one point cloud. RhinoNet's latent space is optimized such that the optimal transport distance between pairs of point clouds is preserved. When applied to single cell data, the resulting patient level phenoscape is organized in a biologically meaningful way. RhinoNet's geometry-aware phenoscape also enables generation of new single cell populations conditioned on patients and treatments. The geodesic interpolation in the latent space decodes back the points accurately. These generated and interpolated point clouds contain meaningful biological signals about disease progression or untried treatments, which are sparsely measured due to cost and time. Hence, reasoning at the level of populations rather than cells makes cohorts of point clouds a data type that can be compared, searched, and generated to progress biological discoveries.

AI USE STATEMENT

We used large language models (LLMs), including Claude (Anthropic) for writing and coding. Among the tasks that require disclosure, we used LLMs to implement parts of RhinoNet and the baseline adaptations and to give feedback on whether our descriptions of results matched the figures and tables; the interpretations reported in the paper were decided by the authors. We did not use LLMs to generate synthetic datasets, develop theoretical models, formulate or prove mathematical claims, propose hypotheses, design experiments, or clean datasets, and translation and qualitative data analysis are not applicable to this work. Additionally, we used LLMs to draft, edit, and structure parts of the paper, to identify and summarize relevant literature, and to format tables. The authors reviewed and revised all LLM-assisted text, checked every LLM-suggested reference against the original paper, and reviewed and tested all LLM-assisted code and computed values. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only previously published data and involves no new data collection from human subjects. The patient-derived organoid data (Ramos Zapatero et al., 2023) and the CODEX data from the Charville and UPMC cohorts (Wu et al., 2022) are both publicly available data cohorts. RhinoNet generates synthetic single-cell point clouds, including point clouds for conditions that were never measured. These are model outputs, not measurements: they should be labeled as synthetic when shared and should not inform clinical or treatment decisions without experimental validation. Our cohorts include a small number of patients with colorectal or head-and-neck cancer, so patterns learned from them may not generalize to other patient populations, cancer types, or measurement platforms.

REPRODUCIBILITY STATEMENT

We have made all code for RhinoNet, the baseline adaptations, and the evaluation available at `https://anonymous.4open.science/r/RhinoNet-B4DE/`, and it is also included in the supplementary material. All datasets used in this work are publicly available: the patient-derived organoid data (Ramos Zapatero et al., 2023) and the CODEX data from the Charville and UPMC cohorts (Wu et al., 2022) can be obtained from the original studies, and the synthetic Swiss roll can be regenerated with the provided code. Section 3 describes the architecture, the geometric Hodge Laplacian, and the training objective, and Appendix B describes the cohort sizes, marker panels, and preprocessing. Section 4 describes how each baseline was adapted to high-dimensional marker spaces, and Appendix D lists the hyperparameters and training details.

REFERENCES

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In International conference on machine learning, pp. 40–49. PMLR, 2018.

Douglas Arnold, Richard Falk, and Ragnar Winther. Finite element exterior calculus: from hodge theory to numerical stability. Bulletin of the American mathematical society, 47(2):281–354, 2010.

Douglas N Arnold. Finite element exterior calculus. SIAM, 2018.

Lazar Atanackovic, Xi Nicole Zhang, Brandon Amos, Mathieu Blanchette, Leo J Lee, Yoshua Bengio, Alexander Tong, and Kirill Neklyudov. Meta flow matching: Integrating vector fields on the wasserstein manifold. In International Conference on Learning Representations, volume 2025, pp. 94586–94610, 2025.

Sergio Barbarossa and Stefania Sardellitti. Topological signal processing over simplicial complexes. IEEE Transactions on Signal Processing, 68:2992–3007, 2020. ISSN 1941-0476. doi: 10.1109/tsp.2020.2981920. URL http://dx.doi.org/10.1109/TSP.2020.2981920.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. SIAM Journal on Scientific Computing, 37(2):A1111–A1138, 2015.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In International conference on database theory, pp. 217–235. Springer, 1999.

Cristian Bodnar, Fabrizio Frasca, Yu Guang Wang, Nina Otter, Guido Montúfar, Pietro Liò, and Michael M. Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. CoRR, abs/2103.03212, 2021. URL https://arxiv.org/abs/2103.03212.

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. Nature methods, 20(11):1759–1768, 2023.

Andrew Y Chen and Tom Zimmermann. Open source cross-sectional asset pricing. ¿ Critical Finance Review, 11(2):207–264, 2022.

Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. Advances in Neural Information Processing Systems, 36:29667–29679, 2023.

William S Chen, Nevena Zivanovic, David Van Dijk, Guy Wolf, Bernd Bodenmiller, and Smita Krishnaswamy. Uncovering axes of variation among single-cell cancer specimens. Nature methods, 17(3):302–310, 2020.

Ronald R Coifman and Mauro Maggioni. Diffusion wavelets. Applied and computational harmonic analysis, 21(1):53–94, 2006.

Nicolas Courty, Rémi Flamary, and Mélanie Ducoffe. Learning wasserstein embeddings. arXiv preprint arXiv:1710.07457, 2017.

Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. ACM Transactions on Graphics (ToG), 32(5):1–11, 2013.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems, 26, 2013.

Mathieu Desbrun, Anil N Hirani, Melvin Leok, and Jerrold E Marsden. Discrete exterior calculus. arXiv preprint math/0508341, 2005.

Adriana Di Martino, Chao-Gan Yan, Qingyang Li, Erin Denio, Francisco X Castellanos, Kaat Alaerts, Jeffrey S Anderson, Michal Assaf, Susan Y Bookheimer, Mirella Dapretto, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. Molecular psychiatry, 19(6):659–667, 2014.

EW Dijkstra. A note on two problems in connexion with graphs. numeriske mathematik, 1, 269–271, 1959.

Johnny Guzmán and Pratyush Potu. A framework for analysis of dec approximations to hodge-laplacian problems using generalized whitney forms. arXiv preprint arXiv:2505.08934, 2025.

Doron Haviv, Russell Zhang Kunes, Thomas Dougherty, Cassandra Burdziak, Tal Nawy, Anna Gilbert, and Dana Pe'Er. Wasserstein wormhole: Scalable optimal transport distance with transformer. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 17697–17718. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/haviv24a.html.

Guillaume Huguet, Alexander Tong, Edward De Brouwer, Yanlei Zhang, Guy Wolf, Ian Adelstein, and Smita Krishnaswamy. A heat diffusion perspective on geodesic preserving dimensionality reduction. In Thirty-seventh Conference on Neural Information Processing Systems, 2023a. URL https://openreview.net/forum?id=HNd4qTJxkW.

Guillaume Huguet, Alexander Tong, María Ramos Zapatero, Christopher J Tape, Guy Wolf, and Smita Krishnaswamy. Geodesic sinkhorn for fast and accurate optimal transport on manifolds. In 2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. IEEE, 2023b.

Mehdi Joodaki, Mina Shaigan, Victor Parra, Roman D Bülow, Christoph Kuppe, David L Hölscher, Mingbo Cheng, James S Nagai, Michaël Goedertier, Nassim Bouteldja, et al. Detection of patient-level distances from single cell genomics and pathomics data with optimal transport (pilot). Molecular systems biology, 20(2):57, 2023.

Michael Keller, David S Schimel, W William Hargrove, and Forrest M Hoffman. A continental strategy for the national ecological observatory network. The Ecological Society of America: 282-284, 2008.

Jinwoo Kim, Jaehoon Yoo, Juho Lee, and Seunghoon Hong. Setvae: Learning hierarchical composition for generative modeling of set-structured data. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15054–15063. IEEE, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In International conference on machine learning, pp. 3744–3753. PMLR, 2019.

Lek-Heng Lim. Hodge laplacians on graphs. Siam Review, 62(3):685–715, 2020.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. Nature methods, 16(8):715–721, 2019.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. Molecular systems biology, 19(6):MSB202211517, 2023.

Hiren Madhu, Sravanthi Gurugubelli, and Sundeep Prabhakar Chepuri. Unsupervised parameter-free simplicial representation learning with scattering transforms. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 34145–34160. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/madhu24a.html.

Mamdouh S Mohamed, Anil N Hirani, and Ravi Samtaney. Discrete exterior calculus discretization of incompressible navier–stokes equations over surface simplicial meshes. Journal of Computational Physics, 312:175–191, 2016.

Mamdouh S Mohamed, Anil N Hirani, and Ravi Samtaney. Numerical convergence of discrete exterior calculus on arbitrary surface meshes. International Journal for Computational Methods in Engineering Science and Mechanics, 19(3):194–206, 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.

Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. Proceedings of the IEEE, 106 (5):808–828, 2018.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017a. URL https://arxiv.org/abs/1612.00593.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. CoRR, abs/1706.02413, 2017b. URL http://arxiv.org/abs/1706.02413.

María Ramos Zapatero, Alexander Tong, James W. Opzoomer, Rhianna O'Sullivan, Ferran Cardoso Rodriguez, Jahangir Sufi, Petra Vlckova, Callum Nattress, Xiao Qin, Jeroen Claus, Daniel Hochhauser, Smita Krishnaswamy, and Christopher J. Tape. Trellis tree-based analysis reveals stromal regulation of patient-derived organoid drug responses. Cell, 186(25):5606–5619.e24, 2023. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2023.11.005. URL https://www.sciencedirect.com/science/article/pii/S0092867423012205.

T Mitchell Roddenberry, Nicholas Glaze, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In international conference on machine learning, pp. 9020–9029. PMLR, 2021.

Michael T Schaub, Austin R Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. Random walks on simplicial complexes and the normalized hodge 1-laplacian. SIAM Review, 62(2):353–391, 2020.

Xingzhi Sun, Danqi Liao, Kincaid MacDonald, Yanlei Zhang, Guillaume Huguet, Guy Wolf, Ian Adelstein, Tim G. J. Rudner, and Smita Krishnaswamy. Geometry-aware generative autoencoders for warped riemannian metric learning and generative modeling on data manifolds. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), Proceedings of The 28th International Conference on Artificial Intelligence and Statistics, volume 258 of Proceedings of Machine Learning Research, pp. 1018–1026. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/sun25c.html.

Zoltán Szabó, Bharath K. Sriperumbudur, Barnabás Póczos, and Arthur Gretton. Learning theory for distribution regression. Journal of Machine Learning Research, 17(152):1–40, 2016. URL http://jmlr.org/papers/v17/14-510.html.

Alexander Y Tong, Guillaume Huguet, Amine Natik, Kincaid Macdonald, Manik Kuchroo, Ronald Coifman, Guy Wolf, and Smita Krishnaswamy. Diffusion earth mover's distance and distribution embeddings. In Marina Meila and Tong Zhang (eds.), Proceedings of the 38th International

Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pp. 10336–10346. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/tong21a.html.

Siddharth Viswanath, Hiren Madhu, Dhananjay Bhaskar, Jake Kovalic, David R Johnson, Christopher Tape, Ian Adelstein, Rex Ying, Michael Perlmutter, and Smita Krishnaswamy. Hiponet: A multi-view simplicial complex network for high dimensional point-cloud and single-cell data. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025. URL https://openreview.net/forum?id=UoKt9B1aY8.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (tog), 38(5): 1–12, 2019.

Alistair Wilkinson, Christopher J. Tape, and Smita Krishnaswamy. Pheno-gs: Phenoscape-scale geodesic sinkhorn, 2026. URL https://arxiv.org/abs/2609.27633.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4840–4851. IEEE, 2024.

Zhenqin Wu, Alexandro E. Trevino, Eric Wu, Kyle Swanson, Honesty J. Kim, H. Blaize D'Angio, Ryan Preska, Gregory W. Charville, Piero D. Dalerba, Ann Marie Egloff, Ravindra Uppaluri, Umamaheswar Duvvuri, Aaron T. Mayer, and James Zou. Space-gm: geometric deep learning of disease-associated microenvironments from multiplex spatial protein profiles. bioRxiv, 2022. doi: 10.1101/2022.05.12.491707. URL https://www.biorxiv.org/content/early/2022/05/13/2022.05.12.491707.

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4540–4549, 2019. doi: 10.1109/ICCV.2019.00464.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. Advances in neural information processing systems, 30, 2017.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In Advances in Neural Information Processing Systems (NeurIPS), 2022.

Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 5806–5815. IEEE, 2021.

Chengbin Zhu, Snorre H Christiansen, Kaibo Hu, and Anil N Hirani. Convergence and stability of discrete exterior calculus for the hodge laplace problem in two dimensions. arXiv preprint arXiv:2505.08966, 2025.

A Related Works

Point Cloud and Set Networks. Neural networks on point clouds must map a set of points of any size to a fixed-size representation. Deep Sets (Zaheer et al., 2017) and PointNet (Qi et al., 2017a) apply a shared network to each point and pool the results, and Set Transformer (Lee et al., 2019) adds attention between points before pooling. Other encoders compute features over local structure: PointNet++ (Qi et al., 2017b) groups points into local neighborhoods, DGCNN (Wang et al., 2019) applies edge convolutions over a k-nearest-neighbor graph recomputed at each layer, and Point Transformer (Wu et al., 2024) applies attention within local neighborhoods. Point cloud autoencoders, such as that of Achlioptas et al. (2018) and SetVAE (Kim et al., 2021), learn latent spaces by reconstructing each point cloud. PointNet, PointNet++, DGCNN, and Point Transformer were designed for 3D shapes, where the Euclidean distance between points is meaningful. These encoders are typically trained on supervised targets or on per-cloud reconstruction, and neither objective constrains distances between point clouds in the latent space. RhinoNet adds this constraint and targets high-dimensional point clouds.

Simplicial Networks. Simplicial complexes extend graphs with higher-order simplices such as triangles, and Hodge Laplacians extend the graph Laplacian to signals on these simplices Lim (2020). Simplicial neural networks pass messages between incident or adjacent simplices (Bodnar et al., 2021; Roddenberry et al., 2021), and wavelet constructions on graphs (Coifman & Maggioni, 2006) have been extended to simplicial complexes (Madhu et al., 2024). HiPoNet (Viswanath et al., 2025) applies these tools to point clouds: it builds several simplicial complexes per point cloud from learned feature reweightings, extracts multiscale features with a simplicial wavelet transform, and is trained for supervised tasks. These networks use the combinatorial Hodge Laplacian, which depends only on which simplices are incident to which. RhinoNet's geometric Hodge Laplacian depends on the positions of the points, and RhinoNet is trained without labels.

Optimal Transport Between Point Clouds. Optimal transport (OT) compares two point clouds by the cost of moving the points of one onto the other, lifting a ground cost between points to a distance between point clouds, and entropic regularization makes it tractable with the Sinkhorn algorithm (Cuturi, 2013). Kernel mean embeddings offer another way to compare distributions and underlie distribution regression (Szabó et al., 2016). We use OT because its ground cost gives direct control over how distances between points are measured, which we set to a distance along the data manifold. Static maps compute OT distances between all pairs of point clouds in a collection and embed them non-parametrically. PhEMD (Chen et al., 2020) applies MDS or PHATE (Moon et al., 2019) to the pairwise distance matrix, Diffusion EMD (Tong et al., 2021) computes transport distances by diffusion over a graph shared by the collection, and PILOT (Joodaki et al., 2023) computes OT distances between patients from single-cell data. These embeddings are tied to the collection they were computed on: placing a new point cloud requires new transport problems and a recomputed embedding, and no decoder maps a location back to a point cloud. Deep Wasserstein embeddings (Courty et al., 2017) and Wasserstein Wormhole (Haviv et al., 2024) train an encoder and a decoder so that Euclidean distances between latent points match OT distances, which makes the embedding inductive and invertible, but both use the Euclidean distance between points as the ground cost. RhinoNet likewise solves transport problems only during training, and it replaces the Euclidean ground cost with a geodesic one.

Manifold Geometry and the Ground Cost. Under the manifold hypothesis, high-dimensional data lie near a lower-dimensional manifold, and distances along that manifold can differ from ambient Euclidean distances. In high dimensions, Euclidean distances also concentrate under broad conditions, so that nearest and farthest neighbors become hard to tell apart (Beyer et al., 1999). Diffusion maps (Coifman & Maggioni, 2006) and heat-based geodesics Crane et al. (2013) approximate manifold distances through diffusion over a graph or mesh, and Heat-Geo (Huguet et al., 2023a) uses heat diffusion to build embeddings of individual points that preserve geodesic distances. Geodesic Sinkhorn (Huguet et al., 2023b) uses a heat-diffusion geodesic as the ground cost of entropic OT, so that the distance between point clouds follows the manifold. RhinoNet uses geodesic Sinkhorn distances as a training signal for its encoder, so they need not be computed for every pair of point clouds at test time.

Generative Models for Point Clouds. Generative models for 3D point clouds learn a distribution over point clouds. Achlioptas et al. (2018) train GANs in the latent space of a point cloud autoencoder, PointFlow (Yang et al., 2019) uses continuous normalizing flows over a shape-level latent

variable and over points, PVD (Zhou et al., 2021) generates point clouds by iterative denoising, and LION (Zeng et al., 2022) combines a hierarchical variational autoencoder with diffusion models over global and point-structured latent representations. PointGPT (Chen et al., 2023) pretrains a transformer by autoregressively generating patches of a point cloud. These models were designed for three-dimensional coordinates, and we adapt them to high-dimensional marker spaces as baselines. In single-cell biology, generative models mostly work at the level of points: scGen (Lotfollahi et al., 2019), CPA (Lotfollahi et al., 2023), and CellOT (Bunne et al., 2023) predict how individual cells respond to a perturbation, and Meta Flow Matching (Atanackovic et al., 2025) generates a population from a matched source population. RhinoNet generates whole point clouds from any location in a geometry-preserving latent space, including locations between observed point clouds.

B High Dimensional Point Cloud datasets

PDO Data. The PDO dataset originates from a colorectal cancer drug-screening study by Ramos Zapatero et al. (2023), which profiled 10 patient-derived organoid (PDO cultures using CyTOF under two culture conditions: PDO monoculture and co-culture with cancer-associated fibroblasts (CAFs). To enable high-throughput multiplexing across more than 2,500 3D cultures, the study employed TOBis (thiol-reactive organoid barcoding in situ), measuring a 44-antibody panel spanning PTM signaling, DNA damage response, apoptosis markers, and cell-state and cell-type identifiers. The drug screen covered three standard-of-care chemotherapies, SN-38 (S), 5-fluorouracil (5-FU, F), and oxaliplatin (O), alongside three targeted agents: cetuximab (C), berzosertib/VX-970 (V), and LGK-974 (L) applied as both single agents and combination treatments. For preprocessing, we used the per-cell marker profiles provided by the source study, organizing cells by patient identity, culture condition, and treatment.

Charville Data. The Stanford-CRC dataset comprises 292 CODEX multiplexed immunofluorescence samples from 161 colorectal cancer patients. They were collected across 4 batches and contain 632,280 segmented and classified cells. Each sample represents a tumor microarray core stained with a 40-plex antibody panel, with single-cell protein expression quantified across all markers and cells assigned to types via unsupervised clustering and k-nearest-neighbour label transfer. Samples are annotated with three clinical outcomes: primary outcome (alive after 60 months: 122 patients; recurred or dead within 60 months: 74), binary recurrence status (recurred: 57; not recurred: 129), and time-to-recurrence interval.

UPMC Data. The UPMC-HNC dataset covers 81 head and neck cancer patients across 308 tumor microarray samples with 7 batches. It contains over 2 million cells segmented and classified from 40-plex CODEX imaging. Cell identities were established by unsupervised clustering of per-cell protein expression profiles followed by k-nearest-neighbor label transfer across the full dataset. Clinical annotations span four axes: primary tumor outcome (NED: 197; non-NED: 111), overall survival length, disease recurrence (recurred: 31; not recurred: 253), and HPV infection status (positive: 158; negative: 150).

C Other Sources of High-Dimensional Data Cohorts

RhinoNet applies to cohorts in which each experimental unit contributes a point cloud: an unordered set of observations, possibly of different sizes, measured in a feature space shared across the cohort. The questions it addresses concern relationships between units. Suitable cohorts also carry labels for each unit, such as a patient, treatment, site, or time, which can condition generation or colour a phenoscape. The cohorts we evaluate in Section 4 contain between 186 and 1,678 point clouds, and cohorts with the same structure arise in other fields. In neuroimaging (Di Martino et al., 2014), a multi-site resting-state fMRI study could treat each scanning session as a point cloud whose points are time frames, each described by the signal in several hundred brain regions. In a phenoscape of such a cohort, sessions would lie near one another when their distributions of brain states are similar, and acquisition site and diagnostic group could be examined as separate sources of variation in the latent space. In ecology Keller et al. (2008), a network of long-term monitoring sites that records many organisms at each sampling visit, each organism annotated with environmental and spectral measurements, would yield one point cloud per site and season. Decoding latent points between consecutive seasons of the same site could then generate that site's community at times that were not sampled. In finance Chen & Zimmermann (2022), the monthly cross-section of listed firms, with each firm described by a few hundred accounting and market characteristics, forms one point

cloud per month, so a panel spanning several decades gives several hundred point clouds. Months with similar distributions of firm characteristics would lie near one another in the phenoscape, and latent flow matching could sample synthetic cross-sections conditioned on labels such as a recession indicator. We have not evaluated RhinoNet in any of these settings.

D Experimental Setup

The parameters of RhinoNet are optimized using the AdamW optimizer with a constant learning rate of 10^{-3} and weight decay of 3×10^{-3}. We train the PDO model for at most 120 epochs and the Charville and UPMC models for 100 epochs. For PDO, we apply early stopping with a patience of 20 epochs and a minimum validation-loss improvement of 10^{-4}. In every experiment, model selection is performed using validation loss, and the checkpoint achieving the lowest validation loss is subsequently evaluated on the held-out test set. PDO and Charville populations are embedded into 32-dimensional latent spaces, whereas UPMC populations are embedded into a 16-dimensional latent space. Unless otherwise specified, we use global mean pooling, $J = 2$ wavelet scales, and a PhenoGS regularization weight of 0.1. For PDO, we use $K = 1$; for the spatial Charville and UPMC datasets, we use $K = 2$. The conditional latent-flow generator is optimized using Adam optimizer with a learning rate of 10^{-3} for 30,000 epochs, while the point-cloud decoder is trained for 400 epochs. For the interpolation experiment, we constructed a 20-nearest-neighbor graph over the learned population-level latent representations and identified a shortest path between selected endpoint populations. All representation-learning and generation experiments are performed on a Nvidia H200 GPU with four CPU workers.

D.1 Data Splits

For PDO, patients 75 and 99 are held out for validation and testing, respectively, and the remaining patients are used for training. The same patient-based split is used when estimating standardization parameters. For Charville, patients c001–c003 are used for training, while the populations from c004 are randomly divided equally between validation and testing. For UPMC, patients c001–c004 and c007 are used for training. The combined populations from c005 and c006 are shuffled and divided equally between validation and testing. All randomized operations use seed 1302 unless stated otherwise. The PDO path connected a DMSO control population to an F-treated population from patient 21 and contained three held-out intermediate populations corresponding to L-, S-, and S-treated cultures. The Charville path connected non-recurrent and recurrent populations from patient c001 and contained two held-out intermediate populations. The UPMC path connected non-recurrent and recurrent populations from patient c005 and contained three held-out intermediate populations; because the graph was constructed over the complete cohort, two intermediate populations belonged to patients c006 and c007

D.2 Metrics

We evaluate generated point clouds using Earth Mover's Distance (EMD, or W_1), the 2-Wasserstein distance (W_2), and maximum mean discrepancy (MMD). We selected these metrics because they provide complementary assessments of point-cloud generation quality. Both Wasserstein distances compare populations directly in marker space while accounting for the complete empirical distributions rather than requiring predefined cell-to-cell correspondences. In particular, EMD measures the minimum average displacement required to transport the generated population onto the real population and is comparatively robust to isolated large errors. By contrast, W_2 uses a squared ground cost and therefore penalizes large marker-space discrepancies more strongly. MMD complements these transport-based metrics by comparing distributions through kernel similarities, allowing it to detect nonlinear differences in their overall structure. Using multiple kernel bandwidths makes the MMD evaluation sensitive to discrepancies at several spatial scales.

Let $\widehat{\mathcal{X}}$ and $\mathcal{X}$ denote generated and real point clouds, respectively. EMD is the optimal transport cost under the Euclidean ground distance:

$$W_1(\widehat{\mathcal{X}}, \mathcal{X}) = \min_{\pi \in \Pi(\widehat{\mathcal{X}}, \mathcal{X})} \sum_{i,j} \pi_{ij} \left\| \widehat{\mathbf{x}}_i - \mathbf{x}_j \right\|_2 , \tag{5}$$

where $\Pi(\widehat{\mathcal{X}}, \mathcal{X})$ is the set of couplings between the uniform empirical measures defined by the two point clouds.

We compute W_2 using the squared Euclidean ground cost and report the square root of the resulting optimal transport cost:

$$W_2(\widehat{\mathcal{X}}, \mathcal{X}) = \left(\min_{\pi \in \Pi(\widehat{\mathcal{X}}, \mathcal{X})} \sum_{i,j} \pi_{ij} \left\| \widehat{\mathbf{x}}_i - \mathbf{x}_j \right\|_2^2 \right)^{1/2}. \tag{6}$$

MMD compares the generated and real empirical distributions in a reproducing-kernel Hilbert space. For generated cells $\widehat{\mathcal{X}} = \{\widehat{\mathbf{x}}_i\}_{i=1}^n$ and real cells $\mathcal{X} = \{\mathbf{x}_j\}_{j=1}^m$, we compute the biased empirical squared MMD:

$$\begin{aligned} \mathrm{MMD}_k^2(\widehat{\mathcal{X}}, \mathcal{X}) = \frac{1}{n^2} \sum_{i,i'=1}^{n} k(\widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}_{i'}) + \frac{1}{m^2} \sum_{j,j'=1}^{m} k(\mathbf{x}_j, \mathbf{x}_{j'}) \\ - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(\widehat{\mathbf{x}}_i, \mathbf{x}_j), \end{aligned} \tag{7}$$

where the radial basis function kernel is

$$k_\gamma(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma \left\| \mathbf{x} - \mathbf{y} \right\|_2^2 \right). \tag{8}$$

We report the mean of $\mathrm{MMD}_{k_\gamma}^2$ over $\gamma \in \{2, 1, 0.5, 0.1, 0.01, 0.005\}$, thereby evaluating agreement across multiple kernel scales. Lower values of all three metrics indicate greater agreement between the generated and real cell distributions.

E Computational Complexity

Let M denote the number of point clouds, D is the latent dimension, and $n^{(m)}$ the number of points in the point cloud $\mathcal{X}^{(m)}$. Training $\mathcal{L}_{GS}$ over a minibatch of B point clouds costs $O(B^2 D)$, while the target distance matrix W is precomputed once at $O(M^2)$ memory and cost dominated by diffusion over a shared point-level graph. Training the generation head costs $O((n^{(m)})^2 D)$ per point cloud under a fixed-iteration Sinkhorn approximation. Latent flow matching operates on latents $\mathbf{z}^{(m)}$ so training over M point clouds for E epochs costs $O(EMV)$, where V is the cost of one forward pass through the velocity network, and sampling J latents with S Euler steps costs $O(SJV)$, both independent of $n^{(m)}$. Decoding the point clouds then adds $O(J\bar{n}V_G)$, where $\bar{n}$ is the mean number of points across J generated point clouds and V_G is the cost of one forward pass through the generation head. Interpolation builds a k-nearest neighbor graph over the M latents and runs Dijkstra's algorithm at $O((M + Mk) \log M)$ per query.

F Additional Results

F.1 Additional Phenoscapes

We next construct phenoscapes of spatial proteomics data from the Charville cohort (Wu et al., 2022), in which each point cloud is a tissue region imaged by CODEX, and each cell is described by 40 protein markers. We colour each region by the mean expression of markers for the main tissue compartments: tumour (PanCK), stroma (Vimentin), immune cells (CD45) and T cells (CD3e). Figure A1 shows that RhinoNet arranges the regions along a continuous axis from tumour-dominated regions, with high PanCK, to regions rich in stroma and immune infiltrate, with high Vimentin, CD45 and CD3e. The marker gradients change smoothly across the phenoscape, even though RhinoNet is never given marker-level supervision. In the PointNet++, PointTransformer and Meta Flow Matching phenoscapes, regions with high and low expression are interleaved, with no consistent gradient. DiffusionEMD instead separates regions by slide rather than by tissue composition, which suggests that it is dominated by a batch effect. On the UPMC cohort, RhinoNet again orders regions along a tumour–immune axis (Figure A2), and it keeps the tumour marker smoothest across latent neighbourhoods of all methods tested. Across all 22 markers, DiffusionEMD, which operates directly on optimal transport between cell distributions, gives the smoothest gradients, while PointTransformer again collapses the cohort without a consistent gradient.

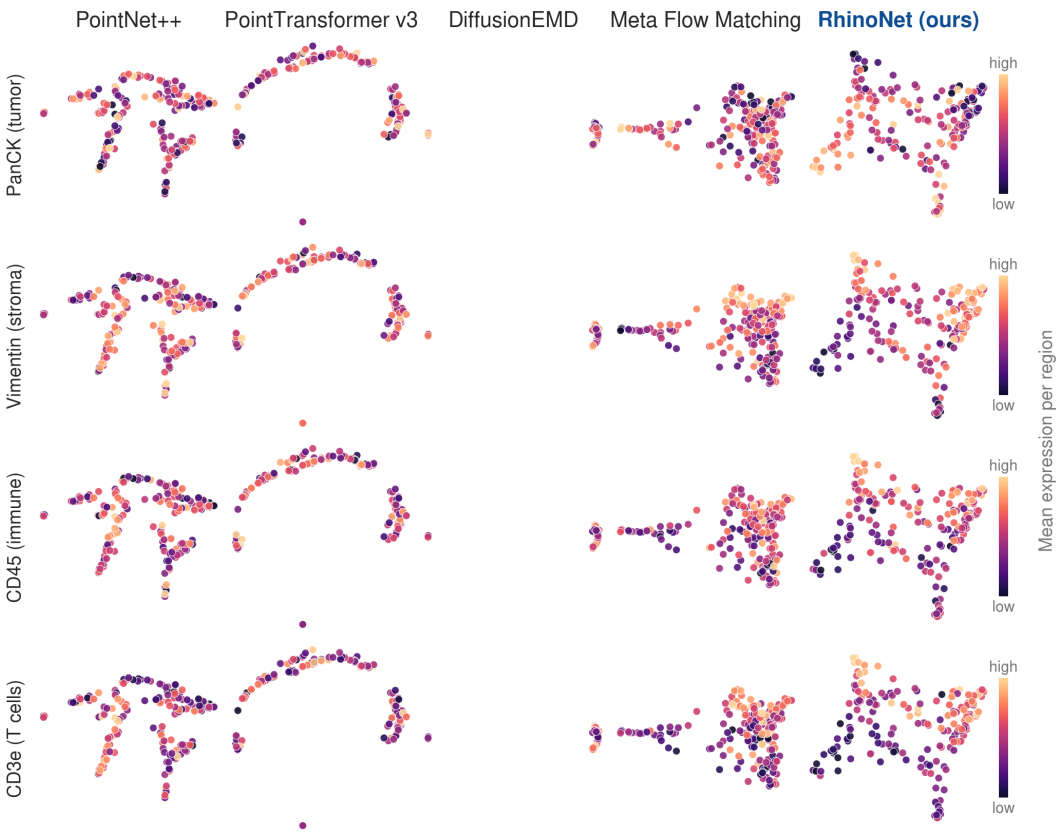


Figure A1: **RhinoNet organises tissue regions along a tumour–stroma axis.** Phenoscapes of 186 CODEX tissue regions from the Charville cohort. Each dot is one region, placed by 2D PHATE ($k = 5$) of that method's latent space and coloured by the region's mean expression of a marker for tumour (PanCK), stroma (Vimentin), immune cells (CD45) or T cells (CD3e). In RhinoNet's phenoscape, tumour-rich and stroma- and immune-rich regions lie at opposite ends of a continuous axis, while the baselines show no consistent gradient. At $k = 5$ the DiffusionEMD neighbour graph splits by slide, so PHATE places the two components far apart.

F.2 Hodge Laplacian Ablation

Figure A3 compares Charville phenoscapes from two RhinoNet models that differ only in the Hodge Laplacian used by the simplicial wavelet transform: the combinatorial Laplacian (left) and our geometric Laplacian (right). Both models use $K = 2$, the same training data, seed and hyperparameters, and are visualised with PHATE on the same 186 tissue regions. Each region is coloured by its mean expression of markers for tumour (PanCK), stroma (Vimentin), immune cells (CD45) and T cells (CD3e). With the geometric Laplacian, the phenoscape spreads regions over two dimensions, and stroma- and immune-rich regions separate from tumour-rich ones. With the combinatorial Laplacian, most regions collapse onto a single arc, and 74% of the latent variance lies along one direction.

F.3 Interpolation Paths

Figure A4 shows the discrete geodesics used in the interpolation experiment (Section 3.2, Figure 6). Each dot is one point cloud, placed by 2D PHATE of RhinoNet's latent space. For each cohort, the path runs through a symmetrized 20-nearest-neighbour graph over the latents, from a point cloud of the start condition to the nearest point cloud of a different condition from the same patient or batch. The intermediate point clouds on the path are held out from decoder training and reconstructed from their latent codes. On PDO, the path from a DMSO culture to a treatment-F culture of patient 21 passes through three held-out cultures of the same patient, treated with LGK-974 (L) and SN-38 (S). On Charville, the path runs from a non-recurrent to a recurrent region of batch c001, and its two held-out regions come from the same batch: the first is non-recurrent and the second is recurrent. On UPMC, the path joins a non-recurrent and a recurrent region of batch c005. Its three held-

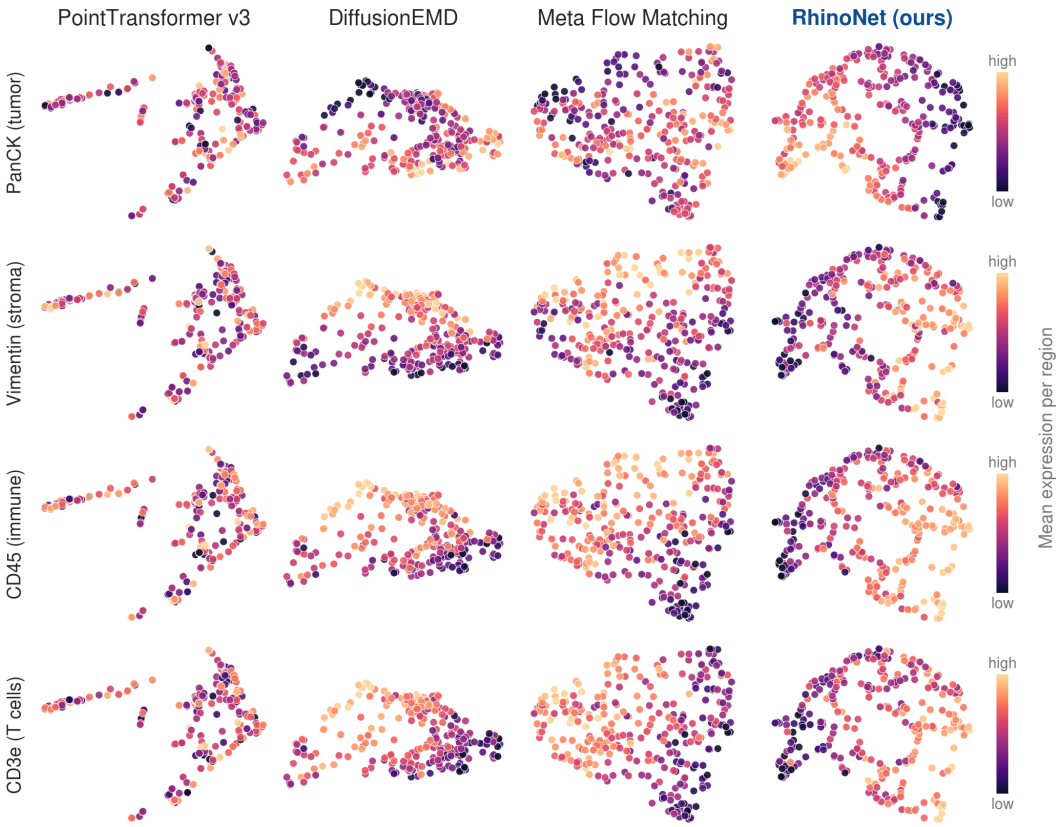


Figure A2: **UPMC phenoscapes.** 308 CODEX tissue regions from the UPMC cohort, one dot per region, placed by 2D PHATE of each method's latent space and coloured by the region's mean expression of markers for tumour (PanCK), stroma (Vimentin), immune cells (CD45) and T cells (CD3e). RhinoNet orders regions from tumour-rich (high PanCK) to stroma- and immune-rich (high Vimentin, CD45 and CD3e).

out regions come from batches c006, c007 and c005, so the geodesic passes through neighbouring batches' samples.

F.4 Simplex Order Ablations

We next ablate the order of the simplicial complex, comparing $K = 2$, which adds triangles to the complex, with $K = 1$, which reduces the encoder to a graph over cells. On both spatial cohorts, $K = 2$ fits the geodesic Sinkhorn distances between tissue regions more closely and reconstructs the wavelet features more accurately. The resulting phenoscapes (Figure A5) show that on UPMC, $K = 2$ arranges regions along a continuous axis of immune infiltration, with smoother CD3e and CD45 gradients across latent neighbourhoods than $K = 1$. On Charville, both models recover the immune gradient, with $K = 1$ producing a smoother one-dimensional ordering, while $K = 2$ gives a closer fit to the transport distances. All markers are shown in Appendix F.4.

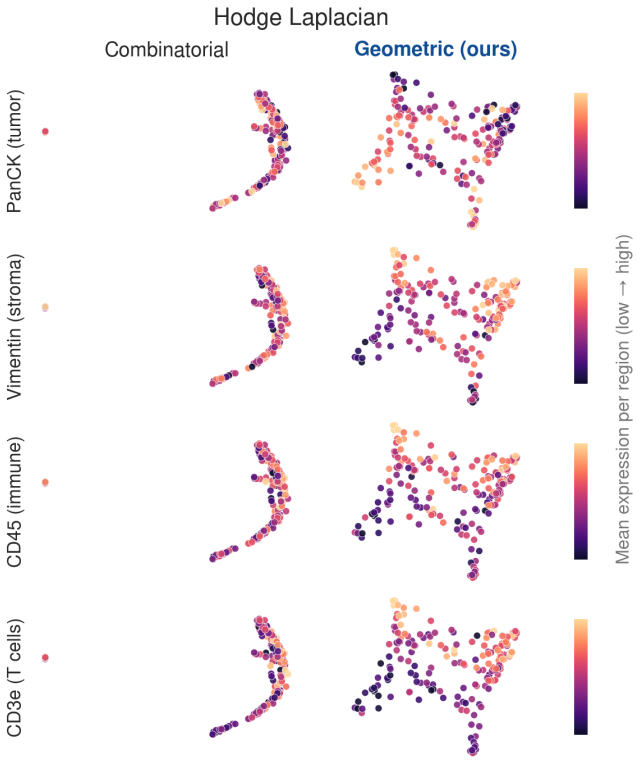


Figure A3: Hodge Laplacian ablation on Charville, coloured by PanCK, Vimentin, CD45 and CD3e.

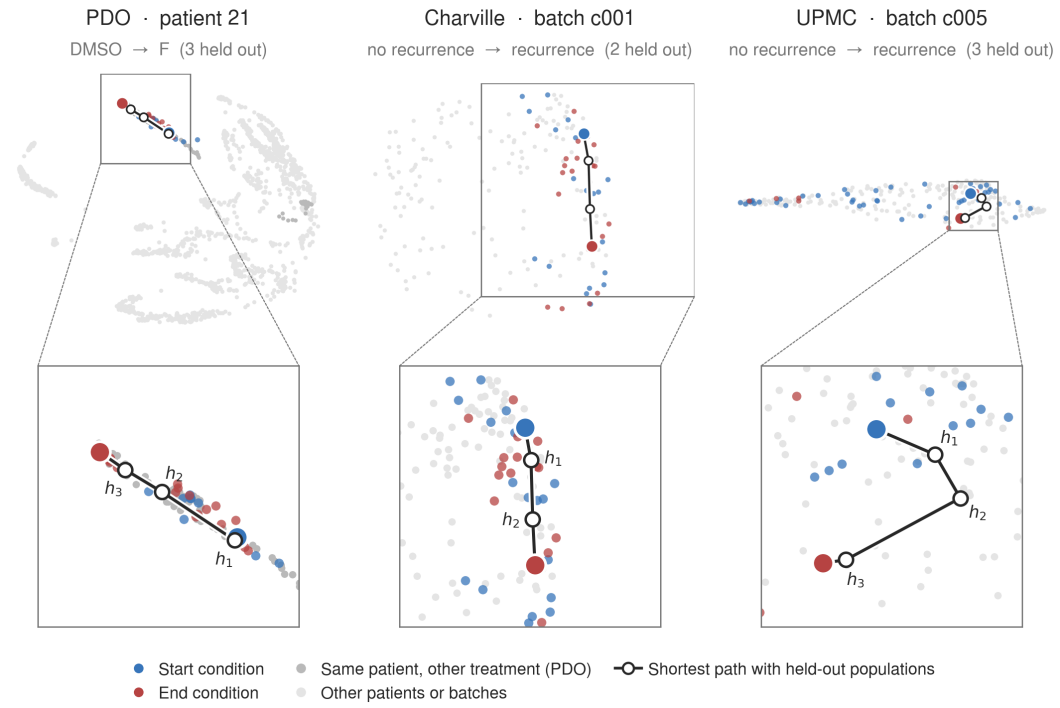
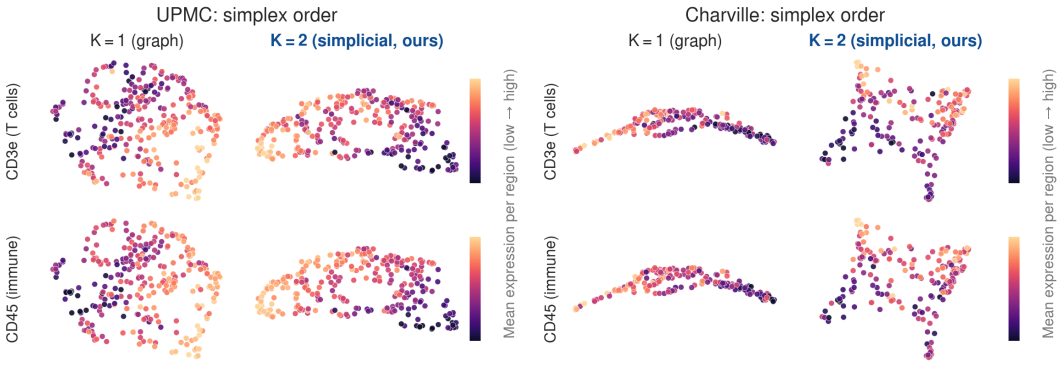


Figure A4: **Interpolation paths through RhinoNet's phenoscape. Top:** 2D PHATE of RhinoNet's latent space for PDO, Charville and UPMC, one dot per point cloud. Point clouds of the interpolated patient are coloured by condition (blue: start, red: end), the same patient's other treatments are dark grey (PDO only), and all other patients are light grey. The box marks the region enlarged below. **Bottom:** the shortest path through the k-NN graph over latents (black line) between two endpoints of different conditions (filled circles). Held-out intermediate point clouds $h_1, \ldots, h_n$ (open circles) are numbered in path order. Both rows use equal axis scales, so distances in the enlarged panels are not distorted.



(a) UPMC (b) Charville

Figure A5: **Ablation of the simplex order K.** Phenoscapes from RhinoNet with $K = 1$ (graph: nodes and edges) or $K = 2$ (simplicial: nodes, edges and triangles), coloured by mean CD3e and CD45 expression per tissue region. On UPMC, $K = 2$ orders regions along a single axis from immune-rich to immune-poor. On Charville, both models separate immune-rich from immune-poor regions.