K-Nearest-Neighbors Induced Topological PCA for Single Cell RNA-Sequence Data Analysis

Sean Cottrell[1], Yuta Hozumi[1] and Guo-Wei Wei[1,2,3*]

[1] Department of Mathematics,
Michigan State University, East Lansing, MI 48824, USA.
[2] Department of Electrical and Computer Engineering,
Michigan State University, East Lansing, MI 48824, USA.
[3] Department of Biochemistry and Molecular Biology,
Michigan State University, East Lansing, MI 48824, USA.

October 24, 2023

Abstract

Single-cell RNA sequencing (scRNA-seq) is widely used to reveal heterogeneity in cells, which has given us insights into cell-cell communication, cell differentiation, and differential gene expression. However, analyzing scRNA-seq data is a challenge due to sparsity and the large number of genes involved. Therefore, dimensionality reduction and feature selection are important for removing spurious signals and enhancing downstream analysis. Traditional PCA, a main workhorse in dimensionality reduction, lacks the ability to capture geometrical structure information embedded in the data, and previous graph Laplacian regularizations are limited by the analysis of only a single scale. We propose a topological Principal Components Analysis (tPCA) method by the combination of persistent Laplacian (PL) technique and $L_{2,1}$ norm regularization to address multiscale and multiclass heterogeneity issues in data. We further introduce a k-Nearest-Neighbor (kNN) persistent Laplacian technique to improve the robustness of our persistent Laplacian method. The proposed kNN-PL is a new algebraic topology technique which addresses the many limitations of the traditional persistent homology. Rather than inducing filtration via the varying of a distance threshold, we introduced kNN-tPCA, where filtrations are achieved by varying the number of neighbors in a kNN network at each step, and find that this framework has significant implications for hyper-parameter tuning. We validate the efficacy of our proposed tPCA and kNN-tPCA methods on 11 diverse benchmark scRNA-seq datasets, and showcase that our methods outperform other unsupervised PCA enhancements from the literature, as well as popular Uniform Manifold Approximation (UMAP), t-Distributed Stochastic Neighbor Embedding (tSNE), and Projection Non-Negative Matrix Factorization (NMF) by significant margins. For example, tPCA provides up to 628%, 78%, and 149% improvements to UMAP, tSNE, and NMF, respectively on classification in the F1 metric, and kNN-tPCA offers 53%, 63%, and 32% improvements to UMAP, tSNE, and NMF, respectively on clustering in the ARI metric.

keywords: Topology, Persistent Homology, Persistent Laplacian, scRNA-seq, dimensionality reduction, clustering, machine learning

*Corresponding author. Email: weig@msu.edu

Contents

1 Introduction

Single cell RNA sequencing (scRNA-seq) is a relatively new method that profiles transcriptomes of individual cells, revealing vast information in the heterogeneity within cell population, which has lead to further understanding of gene expression, gene regulation, cell-cell communication, cell differentiation, spatial transcrtiptomics, signal transduction pathways, and more [1,2]. The workflow of a typical scRNA-seq analysis involves single cell isolation, RNA extraction and sequencing using a library and downstream analysis. With the technological improvements, more than 20,000 genes can be profiled, which has led to a high-dimensionality challenge. Despite the improvements in the methodology that allows for more accurate reading of genes and increasing the number of sequenced cells per experiment, analyzing the data for downstream analysis remains a hurdle. Numerous methods and procedures have been proposed to analyze the data [3–9]. Specific challenges in scRNA-seq data analysis include drop-out events-induced zero expression read count, inadequate sequencing depth-induced inconsistent low reading counts, noise data, and high dimensionality [8,10]. Therefore, dimensionality reduction and feature selection to eliminate low signals is an essential step in analyzing scRNA-seq data.

Various dimensionality reduction and feature selection methods have been proposed for analyzing scRNA-seq data. ScRNA by non-negative and low rank representation (SinLRR) assumes that scRNA-seq data is inherently low rank and finds the smallest ranked matrix that approximates the original data [11]. Single-cell interpretation via multikernel learning (SIMLR) utilizes multiscale kernel to learn a cell-cell similarity metric that can be used for downstream analysis [12]. Deep learning has also been used to perform dimensionality reduction [13–16].

Traditional dimensionality has also been widely incorporated into scRNA-seq analysis pipeline. Nonlinear dimensionality reduction, such as uniform manifold approximation and projection (UMAP), t-distributed stochastic neighbor embedding (t-SNE), multidimensional scaling (MDS) and isomap have been utilized for visualization [17–20]. However, directly applying such algorithm can be challenging because these methods rely on distance calculation and data sparsity, but high dimensional scRNA-seq data may suffer from poor distance calculations. Recently, correlated clustering and projection (CCP) has been used on scRNA-seq data and its visualization [21, 22]. CCP utilizes gene-gene correlation to partition genes, and uses cell-cell correlation on the partitioned genes to project the original genes into super-genes. Non-negative matrix factorization (NMF) has been widely utilized due to its interpretability. NMF uses matrix factorization, where the basis can be interpreted as meta-gene, which are weighted sums of the original genes. Numerous NMF with various constrains has been proposed for scRNA-seq [23, 24]. One of the oldest dimensionality reduction methods, principal components analysis (PCA) is still one of the most popular method used for scRNA-seq [25].

PCA is a linear dimensionality reduction algorithm, where the goal is to compute a orthonormal basis that maximizes the variance of the projected data. The first component is call the principle component, where the variance of the projected data is maximized. The subsequent components i are orthogonal to the $i-1$ components and maximizes the variance of the residual of the approximation. Though PCA is popular because of its efficiency and is linear due to the having no reliance on utilizing metric computation, it has its drawbacks. First, PCA lacks concrete interpretability because of its eigenmode-representation the data. In scRNA-seq, there are many 0-counts in the data, i.e. the gene is not expressed, which can contribute to the principle component and the coefficient matrix can be dense. Second, PCA assumes that the noise of the data is Gaussian, which may not model scRNA-seq data well because the original data is a count matrix. To tackle the first problem, sparse PCA (sPCA) was introduced [20], where by adding a l_1 penalty term on the basis, it allowed for sparsity in the principle components. An alternative formulation of PCA was introduced by Nie et. al. [26] to improve robustness of PCA by assuming the noise is sampled from the Laplace distribution, which is called rPCA. Graph regularization was further introduced by Jiang et. al. which utilizes a graph Laplacian to incorporate nonlinear manifold structure to the reduction [27]. However,

graph Laplacian can only capture a single scale, and is not able to capture the topological structure of the data.

We introduced persistent Laplacian-Enhanced PCA theory in previous works, which was shown to outperform all other state-of-the-art PCA enhancements for microarray data analysis [28]. Persistent Laplacian, also called persistent combinatorial Laplacian or persistent spectral graph, was introduced as a new generation of topological data analysis (TDA) methods in 2019 [29]. It has stimulated a variety of theoretical developments [30–33] and led to remarkable applications [34–36]. PLPCA has the ability to recognize the stability of topological features in our data at multiple scales, and provide a more thorough spatial view through the filtration of a simplicial complex, which induces a sequence of simplicial complexes. Accounting for the spectra of each corresponding Laplacian matrix for each complex in the sequence enables us extract this topological information, improving our ability to preserve intrinsic geometrical information during dimensionality reduction. We can accomplish this by constructing a weighted sum of each Laplacian matrix, generating an accumulated spectral graph. However, PLPCA requires intensive parametrization.

The objective of the present work is to introduce $L_{2,1}$ norm regularization to improve the sparsity and heterogeneity constraint in PLPCA. The resulting technique, called topological PCA (tPCA), allows us to significantly reduce the distribution of the weightings in the accumulated spectral graph, while still achieving near optimal performance. Additionally, we introduce a k-nearest neighbor (kNN) persistent Laplacian algorithm to improve the robustness of our tPCA. The performance of resulting kNN-tPCA does not depend on parameter search. We extensively validate the performance of tPCA and kNN-tPCA for clustering and classification on a series of 11 scRNA-seq datasets. We demonstrate that our new method is superior to other PCA enhancements as well as NMF.

Our work then proceeds as follows. First, we review the mathematical formulations of each of the enhancements that build up to tPCA. Specifically, graph regularization and sparseness. We then discuss the tools for perssistent Laplacian theory which we will incorporate to arrive at our final procedure: RpLSPCA, or tPCA. We then validate the performance of this new method on a set of 11 scRNA-seq datasets against other notable PCA enhancements, as well as NMF, for clustering and classification. Extensive tests indicate that our methods are the state of the art procedure for dimensionality reduction prior to clustering and classification. Specifically, over the 11 tested datasets, tPCA outperforms UMAP on average by 628% on the F1 metric. Lastly, we visualize the tPCA Eigen-Genes via UMAP and tSNE, as well as Residue Similarity Plots to further assess the performance of our methods.

2 Methods

In this section, we provide an overview of PCA and its derivatives, including sparse PCA (sPCA), and graph Laplacian regularized sparse PCA (gLSPCA). We then introduce our method, topological PCA (tPCA) and a less parameter-intensive alternative kNN-tPCA. We will first define the notation used in the following subsection.

2.1 Notations

We summarize our notations as follows.

1. $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \in \mathbb{R}^{M \times N}$, where $\mathbf{x}_j \in \mathbb{R}^M$ indicates the jth sample or cell, and M is the number of genes.
2. $\|A\|_F = \sqrt{\sum_{j=1}^{N} \sum_{i=1}^{M} A_{ij}}$ is the frobenius norm of matrix A.
3. $\|A\|_{2,1} = \sum_{j=1}^{N} \|\mathbf{a}_j\|_2$ is the $l_{2,1}$ norm of A, where the summation is taken after taking the l_2 norm of each column. Alternatively, we can think of this as taking the l_2-norm of the genes and taking the sum

over the cells.

4. $\mathrm{Tr}(A)$ is the trace of matrix A.
5. Let $m << M$ the dimension of the subspace, and M is the number of original dimension, ie the number of genes.
6. Let N be the number of samples or cells.
7. $U \in \mathbb{R}^{m \times M}$ is the principle components, or the basis of the lower dimensional subspace of X, and m is the number of dimension.
8. $Q \in \mathbb{R}^{m \times N}$ is the projection of X onto the subspace spanned by U.

2.2 PCA

Principal Component Analysis has historically served as the baseline approach for dimensionality reduction during gene expression data analysis [37]. The goal of PCA is to express some high dimensional data $X \in \mathbb{R}^{M \times N}$ in a lower dimensional space. This is accomplished via computing the principal components, which are the eigenvectors of the covariance of X corresponding to the largest eigenvalues. Alternatively, we can express PCA as finding a m-dimensional subspace that approximate the data matrix, i.e.,

$$\min_{U,Q} \|X - UQ^T\|_F^2, \quad \text{s.t. } Q^TQ = I_N \tag{1}$$

where $Q^TQ = I_N$ is the orthonormal constrain, and I_N is the $N \times N$, or cell by cell, identity matrix. When the original matrix X is 0-mean 1-variance scaled, we can see that this formulation is equivalent to take the eigenvectors of the covariance matrix. Alternatively, the orthonormal constrain can be applied to U, which yields traditional PCA.

2.3 Sparse PCA

PCA requires that the principal components be expressed as linear combinations of all the features with non-zero weightings. However, in the context of gene expression data analysis, this introduces unnecessary computational complexity and noise, because many genes are not expressed in the cells, or is only expressed under particular circumstances [3]. Therefore, the interpretability of PCA is significantly aided by the introduction of Sparse PCA (sPCA), which allows for zero weightings [20]. Sparse PCA can take several forms, notably the inclusion of an $L_{2,1}$ norm penalty term in the objective function as in Eq. 2:

$$\min_{U,Q} \|X - UQ^T\|_F + \beta\|Q\|_{2,1}, \quad \text{s.t. } Q^TQ = I_N \tag{2}$$

The $L_{2,1}$ norm is defined as $\|A\|_{2,1} = \sum_{i=1}^{n} \|\mathbf{a}_i\|_2$, or first calculating the L_2 norm of each row, and then computing the L_1 norm of row-based L_2 norms. The β parameter scales the sparse regularization term.

2.4 Graph Laplacian Regularized Sparse PCA

While the inclusion of sparse regularization should address some of the shortcomings of PCA relating to interpretability, it does not address the inability of PCA in recognizing complex geometric structures which are present in the higher dimensional space. Graph Laplacian has been commonly used to incorporate geometric information into the reduction such that similar samples in high dimension will be closer in the lower dimensional embedding. This can be accomplished with neighbor graphs with pairwise edges, specifically, the Laplacian operator [38].

Let $G(V, E, W)$ be a nearest neighbor graph, where V is the set of vertices, E is the edge, and ω are the weights of the edges. E can be defined as $E = \{(\mathbf{x}_j, \mathbf{x}_i) : \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)\}$, where $\mathcal{N}_k(\mathbf{x}_j)$ is

the k-nearest neighbors of sample j under some metric. For pairs of points in the edge set, we can define a weight satisfying the following two properties

$$\Phi(\mathbf{x}_i, \mathbf{x}_j) \to 1 \quad \text{as} \|\mathbf{x}_i - \mathbf{x}_j\| \to 0$$
$$\Phi(\mathbf{x}_i, \mathbf{x}_j) \to 0 \quad \text{as} \|\mathbf{x}_i - \mathbf{x}_j\| \to \infty.$$

Such condition is satisfied by a class of functions called radial basis functions. For this work, we utilize the Gaussian kernel as the edge weights

$$W_{ij} = \begin{cases} e^{-\|\mathbf{x}_i, \mathbf{x}_j\|^2/\eta} & \text{if } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The matrix W is known as the weighted adjacency matrix. Here, $\eta \in \mathbb{R}$ defines the geodesic distance, or the width of the Gaussian kernel. We can then define the Graph Laplacian L, by taking

$$L = D - W, \tag{4}$$
$$D_{ii} = \sum_{j=1}^{N} W_{i,j}, \tag{5}$$

where D is the degree matrix, defined as the row sum of W, which shows the total connectivity of the vertex i. Laplacian graph provides a graphical embedding, which can be used as a regularization for PCA [27].

In order to incorporate manifold regularization into the PCA framework, consider the distance $\|\mathbf{q}_i - \mathbf{q}_j\|^2$, where $\mathbf{q}_i$ and $\mathbf{q}_j$ correspond to the lower dimensional representation of samples $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. Using the graph weights W_{ij}, we see that if $W_{ij} \to 1$, ie $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, then $\|\mathbf{q}_i - \mathbf{q}_j\| \to 0$. Alternatively, if $W_{ij} \to 0$, ie $\mathbf{x}_i$ and $\mathbf{x}_j$ are dissimilar, $\|\mathbf{q}_i - \mathbf{q}_j\|^2 \to \infty$. Using this fact, we want to minimize the following.

$$\begin{aligned} R &= \frac{1}{2} \sum_{ij}^{N} W_{ij} \|\mathbf{q}_i - \mathbf{q}_j\|^2 \\ &= \frac{1}{2} \sum_{ij}^{N} W_{ij} \mathbf{q}_i^T \mathbf{q}_i + \mathbf{q}_j^T \mathbf{q}_j - \sum_{ij}^{N} W_{ij} \mathbf{q}_i^T \mathbf{q}_j^T \\ &= \sum_{i}^{N} D_{ii} \mathbf{q}_i^T \mathbf{q}_i - \sum_{ij}^{N} W_{ij} \mathbf{q}_i^T \mathbf{q}_j^T \\ &= \mathrm{Tr}(Q^T D Q) - \mathrm{Tr}(Q^T W Q) \\ &= \mathrm{Tr}(Q^T L Q). \end{aligned}$$

Utilizing the sparse PCA and the manifold regularization, we obtain the graph Laplacian sparse PCA (gLSPCA)

$$\min_{U,Q} \|X - UQ^T\|_F + \beta \|Q\|_{2,1} + \gamma \mathrm{TR}(Q^T L Q), \quad \text{s.t. } Q^T Q = I_N. \tag{6}$$

We also observe that the loss function of gLSPCA is Frobenius norm regularization, which is sensitive to outliers and data heterogeneity when dealing with multiclass data. We then propose replacing Frobenius norm regularization with $\mathrm{L}_{2,1}$ norm regularization to achieve robustness. This yields the following new objective function, which we call Robust graph Laplacian Sparse PCA (RgLSPCA),

$$\min_{U,Q} \|X - UQ^T\|_{2,1} + \beta \|Q\|_{2,1} + \gamma \mathrm{Tr}(Q^T L Q), \quad \text{s.t } Q^T Q = I_n. \tag{7}$$

2.5 Topological PCA

While the inclusion of sparseness and graph Laplacian regularization seeks to address interpretability and geometric structure capture, it still lacks the ability to recognize the stability of topological features at multiple scales, as well as homotopic shape information [29]. To this end, we turn to persistent Laplacian regularization. Like persistent homology, persistent spectral graph theory tracks the birth and death of topological features of data as they change over scales [30, 39]. We perform this analysis via a filtration procedure on our data to construct a family of geometric structures [29]. We then can study the topological properties of each configuration by its corresponding Laplacian matrix.

First, we must briefly review the notion of a simplex, simplicial complex, q-chain, and boundary. A 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle, and so on. Generally, we consider a q-simplex, σ_q. A simplicial complex is then a means of approximating a topological space by gluing together the faces of simplices. More formally, a simplicial complex K is a collection of simplices such that:

1. If $\sigma_q \in K$ and σ_p is a face of σ_q then $\sigma_p \in K$.
2. The nonempty intersection of any two simplices is a face of both simplices.

A q-chain is then defined as a formal sum of q-simplices in a simplicial complex K with coefficients in $\mathbb{Z}_2$. The set of q-chains has a basis in the set of q-simplices in K, and this set forms a finitely generated free Abelian group $C_q(K)$. We then define the boundary operator as a homomorphism relating the Chain groups, $\partial_q : C_q(K) \to C_{q-1}(K)$. The boundary operator is defined as:

$$\partial_q \sigma_q = \sum_{i=0}^{q} (-1)^i \sigma_{q-1}. \tag{8}$$

where σ_{q-1} is a $q-1$ simplex. The sequence of chain groups connected by this homomorphism is then a Chain Complex:

$$... \xrightarrow{\partial_{q+1}} C_q(K) \xrightarrow{\partial_q} C_{q-1}(K) \xrightarrow{\partial_{q-1}} ....$$

It is well known that the boundary operator and the Chain Complex associated with a simplicial complex gives the number of q-dimensional holes in that topological space. Specifically, the qth Homology Group is defined as $H_q = \ker\partial_q / \mathrm{Im}\partial_q$. This is also known as the qth Betti Number, β_q. The matrix representation of the qth boundary operator with respect to the standard basis in $C_q(K)$ and $C_{q-1}(K)$ is given as $\mathcal{B}_q$. Besides considering the Homology of our topological space, we can also consider its cohomology. To that end, we define the adjoint operator of ∂_q as:

$$\partial_q^* : C_{q-1}(K) \to C_q(K), \tag{9}$$

and the transpose of $\mathcal{B}_q$, denoted $\mathcal{B}_q^T$, is the matrix representation of ∂_q^* with respect to the same basis. We can now define the q-combinatorial Laplacian matrix as:

$$\mathcal{L}_q := \mathcal{B}_{q+1}\mathcal{B}_{q+1}^T + \mathcal{B}_q^T \mathcal{B}_q. \tag{10}$$

The harmonic spectrum of the q-combinatorial Laplacian matrix reveals the dimension of the qth Homology group, or the number of q-dimensional holes in our simplicial complex. The non-harmonic spectrum then reveals further homotopic shape information [40]. Intuitively, β_0 reveals the number of connected components in K, β_1 reveals the number of loops in K, and β_2 reveals the number of 2D voids in K.

However, this framework is confined to the analysis of only a single simplicial complex, or the connectivities at only a single scale. To enrich our spectral information, Persistent spectral graph theory proposes creating a sequence of simplicial complexes by varying a filtration parameter [40]:

$$\{\emptyset\} = K_0 \subseteq K_1 \subseteq ... \subseteq K_p = K.$$

For each subcomplex K_t we can denote its chain group to be $C_q(K_t)$, and the q-boundary operator $\partial_q^t : C_q(K_t) \to C_{q-1}(K_t)$. By convention, we define $C_q(K_t) = \{0\}$ for $q < 0$ and the q-boundary operator to then be the zero map. The boundary operator and adjoint boundary operator are otherwise defined similarly as before for each K_t in the sequence, which allows us to define a sequence of Chain Complexes.

Next, we introduce persistence to the Laplacian spectra. Define the subset of C_q^{t+p} whose boundary is in C_{q-1}^t as $\mathbb{C}_q^{t,p}$, assuming the natural inclusion map from $C_{q-1}^t \to C_{q-1}^{t+p}$.

$$\mathbb{C}_q^{t,p} := \{\beta \in C_q^{t+p} | \partial_q^{t,p}(\beta) \in C_{q-1}^t\}. \tag{11}$$

On this subset, one may define the p-persistent q-boundary operator denoted by $\hat{\partial}_q^{t,p} : \mathbb{C}_q^{t,p} \to C_{q-1}^t$ and corresponding adjoint operator $(\hat{\partial}_q^{t,p})^* : C_{q-1}^t \to \mathbb{C}_q^{t,p}$, as before. The matrix representation of the p-persistent q-boundary operator in simplicial basis is then $\mathcal{B}_{q+1}^{t,p}$, and the matrix representation of the adjoint operator is again the transpose. This allows us to define the q-order p-persistent Laplacian matrix as:

$$\mathcal{L}_q^{t,p} := \mathcal{B}_{q+1}^{t,p}(\mathcal{B}_{q+1}^{t,p})^T + (\mathcal{B}_q^t)^T \mathcal{B}_q^t. \tag{12}$$

We may again recognize the multiplicity of zero in the spectrum of $\mathcal{L}_q^{t,p}$ as the q'th order p-persistent Betti number $\beta_q^{t,p}$ which counts the number of (independent) q-dimensional voids in K_t that still exists in K_{t+p} [29]. We can then see how the q'th-order Laplacian is actually just a special case of the q'th-order 0-persistent Laplacian at a simplicial complex K_t, or rather, at a snapshot of the filtration.

We can capture a more thorough view of the spatial features of our data by focusing on the 0-persistent Laplacian as we have done in previous works [28]. Specifically, we calculate Vietoris-Rips complexes by varying a filtration parameter on the weighted entries of our Laplacian matrix, which correspond to the weighted edges in our graph structure. By gradually increasing a distance threshold, we induce a sequence of simplicial complexes and subgraphs to analyze. In previous works on Persistent Laplacian-enhanced PCA, we provided a convenient computational method for this. For a graph Laplcian matrix L, observe:

$$L = (l_{ij}), l_{ij} = \begin{cases} l_{ij}, i \neq j, i, j = 1, ..., n \\ l_{ii} = -\sum_{j=1}^n l_{ij}. \end{cases} \tag{13}$$

For $i \neq j$, let $l_{\max} = \max(l_{ij})$, $l_{\min} = \min(l_{ij}), d = l_{\max} - l_{\min}$. Set the t^{th} Persistent Laplacian $L^t, t = 1, ..., p$:

$$L^t = (l_{ij}^t), l_{ij}^t = \begin{cases} 0, \text{ if } l_{ij} \leq (t/p)d + l_{\min} \\ -1, \text{otherwise.} \end{cases} \tag{14}$$

Here,

$$l_{ii}^t = -\sum_{j=1}^n l_{ij}^t. \tag{15}$$

We then weight each L^t in the sequence and sum to consolidate each subgraph into a single term, denoted PL. This new term should encode the persistence of topological features as the filtration progresses over multiple scales

$$PL := \sum_{t=1}^p \zeta_t L^t. \tag{16}$$

The optimal ζ weightings are hyper parameters which are obtained via Grid Search. Ideally, we should recognize which scales of connectivity contribute the most important information to our analysis, and place greater emphasis on that corresponding Laplacian matrix in the sum. More details regarding this procedure can be found in our previous works [28]. Substituting this PL term into Eq. 7 gives rise to Robust Persistent Laplacian Sparse PCA (RpLSPCA) which is obtained via the following optimization formula:

$$\min_{U,Q} \|X - UQ^T\|_{2,1} + \beta\|Q\|_{2,1} + \gamma \text{Tr}(Q^T(PL)Q), \quad \text{s.t } Q^TQ = I_n. \tag{17}$$

For convenience, we can also refer to this method simply as Topological PCA (tPCA). This method should better retain geometrical structure information by emphasizing topological features that are persistent at multiple scales through the harmonic spectra, while the non harmonic spectra contributes other geometric shape information.

2.6 KNN Induced Persistent Laplacians

Rather than varying a distance threshold on the entries of our weighted Laplacian matrix to induce filtration, we can instead vary the number of neighbors we use to construct each graph structure. This construction has been used in the past to establish kNN-based persistent homology techniques, and extends naturally to the construction of a sequence of Persistent Laplacians [41]. Observe Figure 1. We then can replace PLs in Eq. 16 by kNN-PLs

$$L^k = (l^k_{ij}), l^k_{ij} = \begin{cases} -1, \text{ if } i \neq j \text{ and } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0, \text{ otherwise} \end{cases} \tag{18}$$

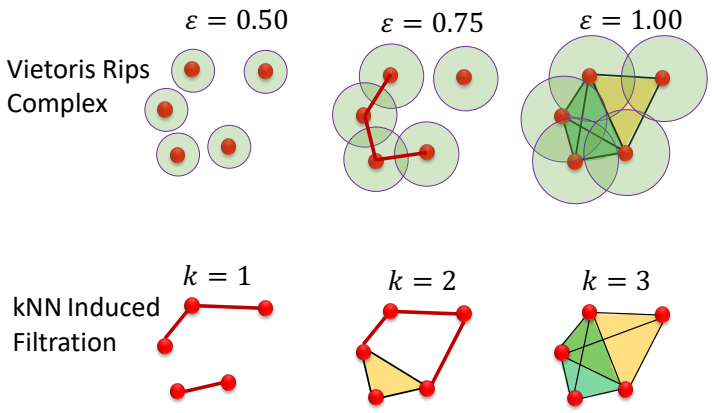

Figure 1: Comparison of inducing filtration via Vietoris Rips Complex, which is reliant on a chosen distance threshold ϵ, and k-nearest-neighbors induced filtration, which is induced by varying the number of nearest neighbors at each node in our graph.

We vary k from 1 to p to establish a sequence of p Persistent Laplacians with different connectivities at each scale of the filtration. Again, diagonal entries are set equal to negative row sums, giving the total connectivity of each vertex. While the Vietoris-Rips based construction requires a choice of scale parameter for the Gaussian Kernel weighting of our graph Laplacian for the sake of normalization, the kNN construction eliminates the need for this since the filtrations do not depend on varying a distance threshold [41]. Furthermore, the sparseness of scRNA-seq data can affect distance measurements, leading to difficulties in establishing meaningful connections between cells when reliant on a distance metric. By inducing filtration through varying the number of neighbors rather than a distance threshold, the result is then a more standardized sequence of connectivity. We also tend to observe the formation of more connected cycles via this method, which lends itself to richer, more interesting topological features in our data. We now validate the performance of these new methods against other enhanced PCA procedures, as well as NMF and tSNE, on several benchmark scRNA-seq datasets.

3 Results

3.1 Data and Preprocessing

Table 1 shows the summary of the data, including the GEO accession ID, reference, source organism, number of samples, number of genes, and number of cell types. For each dataset, log-transform was applied, and genes with values less than 10^{-6} were set to zero. Afterward, $20\% - 25\%$ of the lowest variance genes were dropped. In certain instances where a class had fewer than 15 samples, we dropped that class from the analysis. For the PCA methods, values were then demeaned and scaled by the standard deviation, while for NMF, values were standardized by sklearn's NMF function. Note the differing dimensionality and number of cell types of each dataset, underscoring the comprehensiveness of our proposed methods.

Table 1: Accession ID, source organism, and the counts for samples, genes, cell types and normalization for 11 datasets

Accession ID	Reference	Organism	Samples	Genes	Cell types
GSE67835	Darmanis [42]	Human	420	22084	8
GSE75748 cell	Chu [43]	Human	1018	19097	7
GSE75748 time	Chu [43]	Human	758	19189	6
GSE82187	Gokce [44]	Mouse	705	18840	10
GSE94820	Villani [45]	Human	1140	26593	5
GSE84133human1	Veres [46]	Human	1937	20125	14
GSE84133human2	Veres [46]	Human	1724	20125	14
GSE84133human3	Veres [46]	Human	3605	20125	14
GSE84133mouse1	Veres [46]	Mouse	822	14878	13
GSE84133mouse2	Veres [46]	Mouse	1064	14878	13
GSE45719	Deng [47]	Mouse	300	22431	8

3.2 Evaluation Metrics

3.2.1 Adjusted Rand Index (ARI)

ARI describes how well two clusterings agree with each other by comparing pairs of data points and their respective class assignments. It also can account for possible random agreement and adjusts the similarity score accordingly, assigning a value in the range of -1 to 1. A value of 1 indicates a perfect agreement between clusterings, a value of 0 indicates a random chance agreement, and a value of -1 suggests that the clusterings are less similar than they would be by chance. For two clusterings $X = \{X_1, ..., X_r\}$ and $Y = \{Y_1, ..., Y_s\}$, we construct a contingency table $A \in \mathbb{R}^{r \times s}$ with elements a_{ij} which describe the overlap between X_i and Y_j. We then take row sums and column sums to obtain another set of values: $\{q_1, ..., q_r\}$ is the set of row sums and $\{p_1, ..., p_s\}$ is the set of column sums. We can then define the Adjusted Rand Index as:

$$\text{ARI} = \frac{\sum_{ij} \binom{a_{ij}}{2} - (\sum_i \binom{q_i}{2} \sum_j \binom{p_j}{2})}{\frac{1}{2}(\sum_i \binom{q_i}{2} + \sum_j \binom{p_j}{2}) - (\sum_i \binom{q_i}{2} \sum_j \binom{p_j}{2})} \tag{19}$$

3.2.2 Normalized Mutual Information (NMI)

Mutual Information considers a split of the data according to clusters and a split according to true class labels, and measures how these splittings agree with each other. NMI then corrects for any bias and normalizes the scores between 0 and 1. A value of 1 indicates a perfect agreement between the splittings while a value of 0 indicates random chance agreement. The mathematical definition of NMI is given as:

$$\text{NMI}(T, P) = \frac{\text{MI}(T, P)}{(\text{E}(T) + \text{E}(P))/2} \tag{20}$$

Where $\mathrm{MI}(\cdot,\cdot)$ and $\mathrm{E}(\cdot)$ represent mutual information and entropy and T, P represent the true and predicted cluster labels, respectively.

3.2.3 Classification Metrics

Regarding the evaluation metrics used to measure performance for classification tasks, beyond simple accuracy, there are several metrics that are commonly considered. Notably, Precision, Recall, and F1-Score. Below we list the mathematical definition of each.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{21}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{22}$$

$$\text{F1-Score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{23}$$

We note that the F1-score is particularly relevant as it accounts for both precision and recall, making it more robust to class imbalances within the data. In our case, there are noticeable imbalances between cell types in each of the tested dataset, so we emphasize this metric as the most informative in measuring performance.

Given that our datasets generally contain multiple classes, the evaluation criterion we employ is the mean for each cell type. This evaluation approach is commonly known as a macro metric, where performance measures are calculated for each cell type individually and then averaged to obtain an overall score.

$$\text{Macro-Recall} = \frac{1}{c}\sum_{i=1}^{c} \text{Recall}_i \tag{24}$$

$$\text{Macro-Precision} = \frac{1}{c}\sum_{i=1}^{c} \text{Precision}_i \tag{25}$$

$$\text{Macro-F1} = 2\frac{\text{Macro-Precision} \times \text{Macro-Recall}}{\text{Macro-Precision} + \text{Macro-Recall}} \tag{26}$$

3.2.4 Residue Similarity Scores

To enhance visualization, Residue Similarity (RS) scores can be computed [21]. Traditional visualization techniques often involve reducing the data to two or three dimensions, which may result in the loss of structure and integrity in multiclass data. R-S plots were introduced as a method to visualize results while better preserving the underlying structure of the data.

An R-S plot consists of two main components: the residue score and the similarity score. The residue score is calculated as the sum of distances between classes, capturing the dissimilarity between them. On the other hand, the similarity score represents the average similarity within each class, indicating the degree of similarity between instances belonging to the same class. By considering both scores, R-S plots provide a comprehensive representation of the data's structure in a visualization.

Given data of the form $\{(\vec{x}_i, y_i) | \vec{x}_i \in \mathbb{R}^N, y_i \in \mathbb{Z}_l\}_{i=1}^{M}$, we have y_i representing the class label of our ith data point $\vec{x}_i \in \mathbf{X}$. Say that our data has N samples, M features, and L classes. We can then partition our dataset $\mathbf{X}$ into subsets containing each of the classes by taking $\mathcal{C}_l = \{\vec{x}_i \in \mathbf{X} | y_i = l\}$. For each class l we then define the residue score as follows:

$$R_i := R(\vec{x}_i) = \frac{1}{R_{\max}} \sum_{\vec{x}_j \notin \mathcal{C}_l} \|\vec{x}_i - \vec{x}_j\|, \tag{27}$$

where $\|\cdot\|$ denotes the Euclidean distance between vectors and $R_{\max}$ is the maximal residue score for that subset. The similarity score, meanwhile, is given as:

$$S_i := S(\vec{x}_i) = \frac{1}{|\mathcal{C}_l|} \sum_{\vec{x}_j \in \mathcal{C}_l} \left(1 - \frac{\|\vec{x}_i - \vec{x}_j\|}{d_{\max}}\right), \tag{28}$$

where $d_{\max}$ is the maximal pairwise distance of the dataset. For constructing R-S plots, we then take $R(\vec{x})$ to be the x-axis and $S(\vec{x})$ to be the y-axis.

3.3 Comparison of tPCA and Other Methods for KMeans Clustering

We tested tPCA's performance on the datasets described in Table 1, and compared it to PCA, sPCA, RgLSPCA, NMF, UMAP, and tSNE. For this analysis, we performed K-Means clustering after reducing the dimensionality of each dataset such that the number of dimensions equals the number of clusters. To ensure the greatest accuracy in our analysis, we used sklearn's KMeans function, and increased the number of times the k-means algorithm is run with different centroid seeds from 10 to 150. We then performed the clustering with 30 random instances and considered the average performance. To further facilitate a fair and accurate comparison, we utilized sklearn's NMF function, initialized as non-negative random matrices with values scaled by the square root of the mean of X and divided by the number of components. We also increased the number of maximum iterations from 200 to 300, and took the average performance over 20 random initializations. Likewise, we also compare our method to sklearn's tSNE over 20 random intializations with maximum iterations increased to 300. For UMAP, we used the default minimum distance allowed for packing points together in the embedding space, the low value generally should improve clustering performance by providing a cleaner separation between clusters. We used the default number of nearest neighbors in UMAP's kNN structure, which was 15. This value is the same as the initial number of neighbors used to describe the manifold structure in tPCA.

We do, however, note certain weaknesses with NMF, UMAP, and tSNE in this analysis. First, methods such as NMF tend to perform poorly when reducing to such low dimensionality, while methods such as tSNE and UMAP have notable issues with structure preservation. Specifically, neither method is capable of strictly preserving the density or distance in our data. Therefore, we expect that our procedure should prove substantially more effective as a preprocessing step than any of these methods for clustering via KMeans or classifying via K-Nearest-Neighbors, as both models are reliant on the intrinsic distance and density in the data. For the clustering analysis, a summary of our results for Adjusted Rand Index and Normalized Mutual Information can be found in Tables 2,3. For kNN-Induced tPCA, when results are shown as $(\cdot)*$, this implies that a generic connectivity weighting was used, and there was no parameter optimization needed on that dataset to obtain nearly optimal performance.

Table 2: Comparison of tPCA and other methods for Adjusted Rand Index

Dataset and Method	kNN-tPCA	tPCA	RgLSPCA	sPCA	PCA	NMF	tSNE	UMAP
GSE67835	(0.9496)*	**0.9552**	0.9496	0.7625	0.7604	0.6926	0.4200	0.5579
GSE75748cell	0.7475	**0.7789**	0.7454	0.7440	0.7440	0.7530	0.5301	0.6511
GSE75748time	(0.7123)*	**0.7129**	0.7036	0.6128	0.6128	0.6852	0.3746	0.3677
GSE82187	**(0.9932)***	**0.9932**	0.7622	0.7530	0.7530	0.5113	0.4727	0.5193
GSE94820	0.5731	**0.5749**	0.5159	0.5396	0.5396	0.5400	0.3820	0.3917
GSE84133human1	(0.7942)*	**0.8065**	0.7914	0.5675	0.5675	0.6436	0.4446	0.5396
GSE84133human2	**(0.9277)***	**0.9277**	0.9162	0.8090	0.8090	0.6168	0.6237	0.5680
GSE84133human3	**(0.8727)***	0.8722	0.7799	0.6976	0.6976	0.6061	0.5964	0.7016
GSE84133mouse1	**0.7746**	0.7699	0.7699	0.7688	0.6320	0.5417	0.5391	0.4293
GSE84133mouse2	**0.6172**	0.6139	0.6165	0.5041	0.5041	0.4029	0.3978	0.3851
GSE45719	0.4262	**0.4386**	0.4212	0.4133	0.4133	0.4073	0.4068	0.4088

We see from the results in Table 2 that persistent Laplacian-enhanced PCA is able to outperform not only the other enhanced PCA procedures, but also NMF, UMAP, and tSNE, on all of the tested datasets for Adjusted Rand Index. We specifically note the superior performance on GSE82187. our tPCA outperformed RgLSPCA's ARI score by a considerable margin of 0.231. These results clearly indicate the superior ability to preserve local non-linear geometric structure achieved by persistent Laplacian regularization compared to graph Laplacian regularization, and the result is superior performance in clustering analysis. Furthermore, in several instances KNN-Induced Laplacian regularization was able to match the performance of the standard construction without any optimization. Overall, it was shown to at least outperform the other procedures on all but one tested dataset, and with a fraction of the effort required for parameter search.

Table 3: Comparison of tPCA and other methods for Normalized Mutual Information

Dataset and Method	kNN-tPCA	tPCA	RgLSPCA	sPCA	PCA	NMF	tSNE	UMAP
GSE67835	(0.9224)*	**0.9275**	0.9224	0.8192	0.8174	0.78630	0.5817	0.7272
GSE75748cell	0.8850	**0.9230**	0.8825	0.8810	0.8810	0.9036	0.6940	0.7956
GSE75748time	(0.8276)*	**0.8338**	0.8248	0.7220	0.7220	0.8074	0.4847	0.5047
GSE82187	**(0.9853)***	**0.9853**	0.8967	0.8995	0.8995	0.8040	0.6849	0.7841
GSE94820	**0.6757**	0.6738	0.6275	0.6367	0.6367	0.6444	0.4949	0.4831
GSE84133human1	(0.8412)*	**0.8604**	0.8391	0.7905	0.7905	0.8142	0.6550	0.7748
GSE84133human2	**(0.9158)***	**0.9158**	0.9036	0.7816	0.7816	0.7603	0.7044	0.7571
GSE84133human3	**(0.8742)***	0.8736	0.8131	0.8159	0.8160	0.7644	0.6921	0.8226
GSE84133mouse1	**0.8587**	0.8514	0.8581	0.8473	0.7480	0.6847	0.6159	0.6671
GSE84133mouse2	**0.7924**	0.7918	0.7923	0.7094	0.7095	0.6424	0.5860	0.6403
GSE45719	0.6549	**0.6747**	0.6116	0.6022	0.6022	0.5897	0.5960	0.6034

Once again, the results in Table 3 showcase the superiority of tPCA in all tested cases for NMI. We again note the remarkably superior performance of our method on the GSE82187 dataset specifically, where we outperform RgLSPCA in NMI by 0.089. Again, we also note the ability of the kNN-tPCA to provide optimal or near optimal performance compared to the distance based construction while not requiring an extensive parameter tuning procedure. Now, we can average the performance in each metric over all tested datasets to reveal the extent to which tPCA and kNN-tPCA outperform the other methods overall, across our 11 tested datasets.

Table 4: Comparison of tPCA and other methods for each performance metric averaged over all tested datasets

Method	ARI	NMI
kNN-tPCA	**0.7625**	**0.8393**
tPCA	**0.7676**	**0.8464**
RgLSPCA	0.7247	0.8156
sPCA	0.6520	0.7732
PCA	0.6401	0.7640
NMF	0.5818	0.7455
tSNE	0.4716	0.6172
UMAP	0.5018	0.6872

We see from the final results in Table 4 that, on average, tPCA outperforms NMF by a significant measure of 31.92% for ARI and 13.53% for NMI, and RgLSPCA by 3.78% for NMI and 5.92% for ARI. kNN-tPCA, meanwhile, outperforms NMF by 31.05% for ARI and 12.58% for NMI, and RgLSPCA by 2.91% for NMI and 5.22% for ARI. To intuitively illustrate this point, in Figure 2 we provide a barplot comparing the performance metrics of the mentioned procedures averaged over each of the 11 tested datasets.

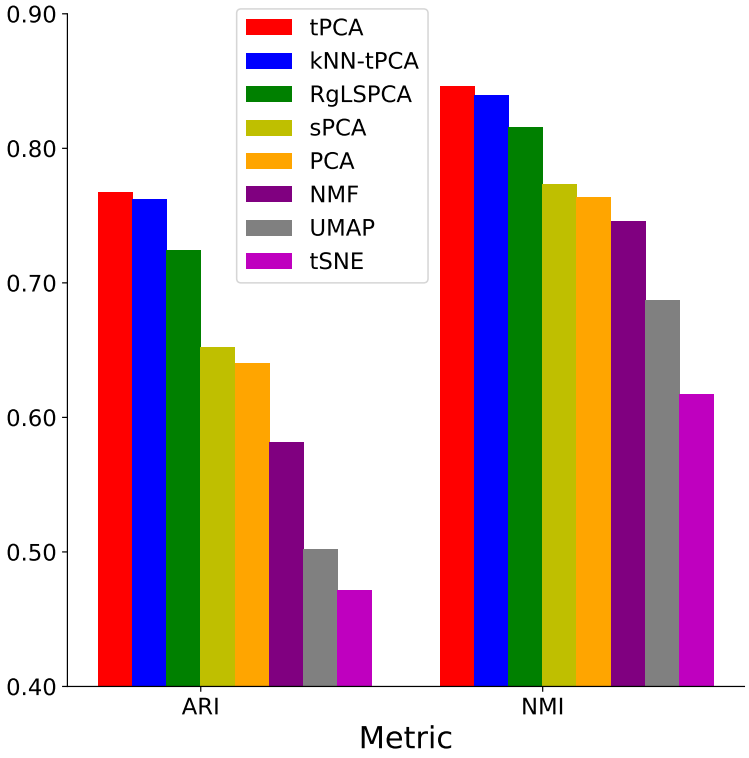

Figure 2: NMF and ARI comparisons for each method averaged over all datasets.

From the depicted image it is clearly evident that both methods for tPCA are superior to all other tested dimensionality reduction techniques, particularly other PCA enhancements. We especially emphasize the superiority of kNN-tPCA given the significantly reduced need for parameter optimization with this method. These results strongly reaffirms the importance of incorporating the topological information and multi-scale

analysis that is possible with topological PCA into a dimensionality reduction technique. While previous techniques can capture some geometrical structure information through graph Laplacian regularization, we see that incorporating the additional filtrations greatly improves performance. Having confirmed the efficacy of our methods, we can move to examining the impact that kNN-induced filtration has on the scale of parameter tuning in more detail, as well as comparing different visualization techniques of the Eigen-Genes each method produces.

3.4 Comparison of kNN-tPCA and Other Methods for Classification

To further validate the efficacy of our proposed method, we can supplement these clustering results with a classification study using kNN. The classification of various cell types begins by randomly splitting our gene expression data into training and testing sets. The kNN model is trained on 60% of the data, and then tested on the remaining 40%. To mitigate the impact of data distribution, we employed a 5-fold cross-validation approach. The classification accuracy was calculated as the average performance over five repetitions. The mean accuracy of the classification was then recorded for subspace dimensions ranging from {100, 90, ..., 10, 1}. The results of this analysis can be seen in Tables 5 and 6.

Table 5: Comparison of average results for kNN-tPCA and Other Methods for Classification after dimensionality reduction to $m = 1, 10, ..., 100$

Dataset	Method	Mean-ACC	Mean Macro-REC	Mean Macro-PRE	Mean Macro-F1
GSE67835	**kNN-tPCA**	**0.8909**	**0.8345**	**0.8672**	**0.8455**
	RgLSPCA	0.8808	0.8032	0.8358	0.8084
	sPCA	0.8319	0.6872	0.7918	0.7052
	PCA	0.7732	0.6172	0.7631	0.63805
	NMF	0.4136	0.2749	0.2740	0.2620
	tSNE	0.6470	0.4986	0.5037	0.4835
	UMAP	0.1793	0.1380	0.0509	0.0588
GSE75748cell	**kNN-tPCA**	**0.9568**	**0.9267**	**0.9332**	**0.9291**
	RgLSPCA	0.9499	0.9204	0.9260	0.9218
	sPCA	0.9305	0.9006	0.9175	0.9046
	PCA	0.7222	0.5731	0.6603	0.5736
	NMF	0.4210	0.3784	0.3784	0.3784
	tSNE	0.5292	0.5257	0.5402	0.5224
	UMAP	0.3505	0.3554	0.2485	0.2678
GSE75748time	**kNN-tPCA**	**0.8222**	**0.8006**	**0.8874**	**0.8068**
	RgLSPCA	0.7928	0.7660	0.8692	0.7667
	sPCA	0.7590	0.7307	0.8353	0.7222
	PCA	0.7587	0.7303	0.8352	0.7219
	NMF	0.3792	0.3501	0.3625	0.3312
	tSNE	0.3858	0.3394	0.3309	0.3178
	UMAP	0.2305	0.1975	0.1006	0.1114
GSE82187	**kNN-tPCA**	**0.9028**	**0.8520**	**0.9115**	**0.8710**
	RgLSPCA	0.8422	0.7280	0.8273	0.7489
	sPCA	0.7357	0.5917	0.6890	0.5958
	PCA	0.7222	0.5731	0.6603	0.5736
	NMF	0.6164	0.3831	0.4070	0.3773
	tSNE	0.5887	0.5648	0.5710	0.5621
	UMAP	0.4791	0.1896	0.1272	0.1374
GSE94820	**kNN-tPCA**	**0.8914**	**0.8346**	**0.8677**	**0.8455**
	RgLSPCA	0.8803	0.8029	0.8349	0.8072
	sPCA	0.8319	0.6872	0.7918	0.7052
	PCA	0.7732	0.6172	0.7631	0.6380
	NMF	0.6618	0.4001	0.4592	0.4263
	tSNE	0.3330	0.3288	0.3358	0.3110
	UMAP	0.2485	0.1983	0.0748	0.0906

We see from this first round of results that kNN-tPCA provides a stellar improvement to performance metrics for classifications using kNN, carried out after dimensionality reduction. Notably, we observe a 2.95% improvement in F1-Score when compared to the standard graph regularization in tPCA and a remarkable 17.5% improvement when compared to traditional PCA. Compared to other dimensionality reduction techniques such as UMAP, tSNE, and NMF, the results are even more significant. This demonstrates the comprehensiveness of tPCA in being able to reduce data to a variety of embeddding dimensions while also preserving important structural information in the data. The results in Table 5 as well as those in Table

6 demonstrate that kNN induced Persistent Laplacian regularization for PCA is a comprehensive method capable of enhancing the performance of classification tasks for Single Cell RNA-Sequence data analysis. Combined with the results in Table 4, we can conclude that tPCA is a superior dimensionality reduction technique for a variety of Machine Learning methods.

To intuitively illustrate these results, in Figure 3 we have provided an illustration depicting the distribution of Accuracy and F1 performance between PCA, RgLSPCA, and kNN-tPCA as we vary the dimensionality of our reduced space on GSE82187. This clearly indicates the superior performance of our proposed method across a wide range of reduced dimensions, especially as the number of dimensions grows larger, where PCA typically suffers from stability issues. This clearly further validates our findings.

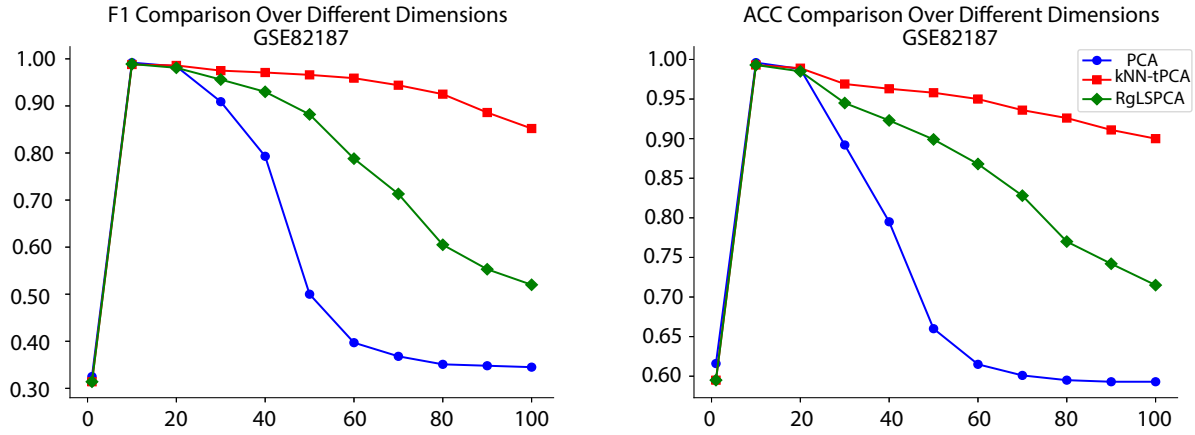

Figure 3: Distributions of ACC and F1 performance for PCA, RgLSPCA, and kNN-tPCA as we vary the number of reduced subspace dimensions.

Table 6: Comparison of average results for kNN-tPCA and Other Methods for kNN Classification after dimensionality reduction to $m = 1, 10, ..., 100$

Dataset	Method	Mean-ACC	Mean Macro-REC	Mean Macro-PRE	Mean Macro-F1
GSE45719	**kNN-tPCA**	**0.9282**	**0.9022**	**0.9119**	**0.9055**
	RgLSPCA	0.9282	0.9003	0.9112	0.9038
	sPCA	0.8371	0.8209	0.8735	0.8296
	PCA	0.8426	0.8260	0.8757	0.8356
	NMF	0.2470	0.2413	0.2460	0.2196
	tSNE	0.4376	0.4452	0.4426	0.4302
	UMAP	0.1581	0.1454	0.071	0.0856
GSE84133human1	**kNN-tPCA**	**0.8911**	**0.8209**	**0.8745**	**0.8417**
	RgLSPCA	0.8837	0.8075	0.8718	0.8308
	sPCA	0.8279	0.7679	0.8633	0.7951
	PCA	0.8279	0.7680	0.8633	0.7952
	NMF	0.5109	0.3973	0.3800	0.3672
	tSNE	0.7697	0.6175	0.6180	0.5960
	UMAP	0.1858	0.1818	0.0827	0.0979
GSE84133human2	**kNN-tPCA**	**0.9216**	**0.8761**	**0.8997**	**0.8829**
	RgLSPCA	0.9157	0.8687	0.8993	0.8765
	sPCA	0.9178	0.8657	0.8861	0.8727
	PCA	0.8973	0.8271	0.8903	0.8395
	NMF	0.5383	0.3652	0.3718	0.3521
	tSNE	0.5786	0.5302	0.5651	0.5258
	UMAP	0.1322	0.1759	0.0758	0.0903
GSE84133human3	**kNN-tPCA**	**0.9062**	**0.8487**	**0.8758**	**0.8600**
	RgLSPCA	0.9034	0.8461	0.8734	0.8573
	sPCA	0.8838	0.8178	0.8615	0.8358
	PCA	0.8279	0.7680	0.8633	0.7952
	NMF	0.5712	0.4298	0.4568	0.3990
	tSNE	0.7226	0.5889	0.6358	0.5867
	UMAP	0.2047	0.1773	0.0948	0.1054
GSE84133mouse1	**kNN-tPCA**	**0.9172**	**0.8825**	**0.8995**	**0.8898**
	RgLSPCA	0.9149	0.8766	0.8986	0.8857
	sPCA	0.9011	0.8541	0.8887	0.8665
	PCA	0.9011	0.8544	0.8889	0.8666
	NMF	0.5620	0.4004	0.4215	0.3886
	tSNE	0.8928	0.6995	0.7092	0.6924
	UMAP	0.3541	0.2593	0.1458	0.1691
GSE84133mouse2	**kNN-tPCA**	**0.9213**	**0.8963**	**0.9002**	**0.8976**
	RgLSPCA	0.9209	0.8953	0.8995	0.8968
	sPCA	0.9018	0.8542	0.8894	0.8667
	PCA	0.9011	0.8544	0.8889	0.8666
	NMF	0.5403	0.3428	0.3567	0.3309
	tSNE	0.6695	0.3759	0.3752	0.3627
	UMAP	0.2501	0.1563	0.0897	0.1008

When we average each of the performance metrics over the 11 tested datasets, we can assess the total improvement that our method provides compared to other techniques as shown in Table 7. We can then intuitively visualize these results by examining Figure 4, which clearly showcases the superiority of kNN-tPCA for classification tasks on a variety of datasets with different dimensionalities and data imbalances.

Table 7: Comparison of kNN-tPCA and other methods for each performance metric averaged over all tested datasets

Method	ACC	REC	PRE	F1
kNN-tPCA	**0.9045**	**0.8613**	**0.8935**	**0.8704**
RgLSPCA	0.8920	0.8377	0.8770	0.8458
sPCA	0.8507	0.7798	0.8443	0.7908
PCA	0.8134	0.7280	0.8138	0.7403
NMF	0.4965	0.3603	0.3739	0.3484
tSNE	0.5958	0.5013	0.5115	0.4900
UMAP	0.2520	0.1977	0.1056	0.1195

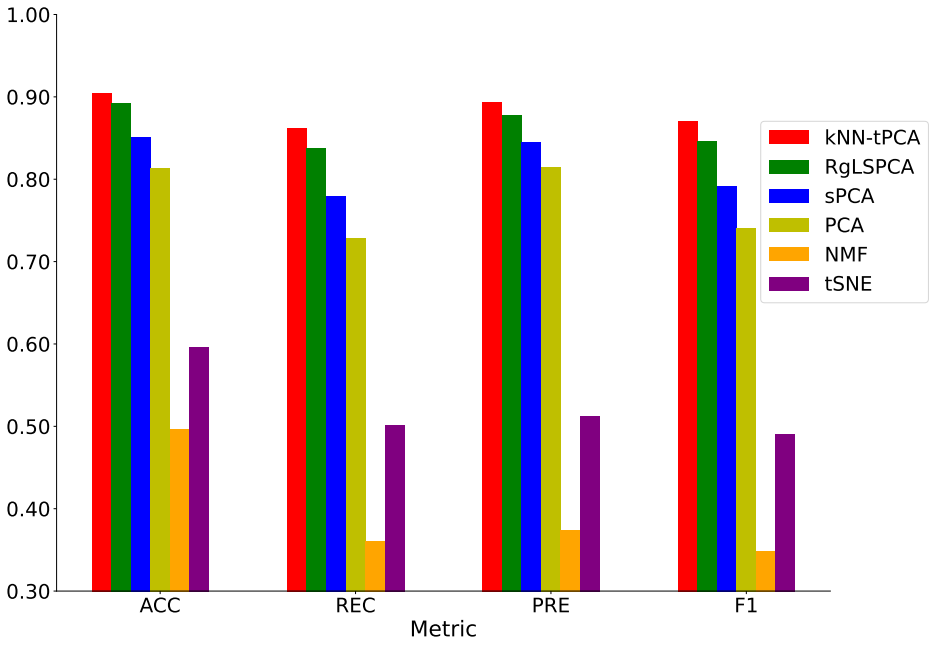

Figure 4: ACC, PRE, REC, and F1 comparisons for each methods averaged over all datasets.

We specifically note that, on average, kNN-tPCA outperforms RgLSPCA by a margin of 1.39% for Macro-ACC, 1.88% for Macro-PRE, 2.82% for Macro-REC, and 2.91% for Macro-F1. This clearly demonstrates the benefits of incorporating multi-scale analysis through the inclusion of persistent Laplacians. Furthermore, we note that our method outperforms traditional PCA by up to a considerable 18.3% for these metrics over the 11 datasets.

4 Discussion

4.1 Parameter Analysis

Regarding the optimization of hyper-parameters for tPCA, we especially note the presence of the ζ weights in the PL term:

$$PL := \sum_{t=1}^{p} \zeta_t L^t \tag{29}$$

Which must be manually chosen for each dataset depending on the connectivity information that is most important. Specifically, for p filtrations we generally consider a distribution of $\{1, 1/2, ..., 1/p, 0\}$ and perform a parameter search over this distribution, while also simultaneously searching for an optimal γ value. However, for larger p this clearly becomes an extremely computationally intensive task, with the number of parameter combinations equaling $((p+1)^p)$(Size of γ distribution). Therefore, rather than considering all parameter combinations over this distribution at once, we can instead consider different combinations of scales of connectivity, say, long, middle, and close range. In other words, for 7 filtrations, testing combinations of $p = 7, 5, 3$, and from there recognizing which scales contribute the most valuable information to narrow our search. Doing so reduces the number of combinations from $((p+1)^p)$(Size of γ distribution) to $(m)((p+1)^3)$(Size of γ distribution), where m is the number of connectivity combinations we need to test to achieve the best results. In practice, we found that generally $m = 3$ obtained allowed us to obtain optimal performance, which is a considerable improvement from the traditional approach to grid search, though clearly still not preferable for practical purposes.

Ideally, the more standardized filtrations present with kNN Laplacians will reduce the need for parameter optimization entirely, significantly reducing computation and time requirements. As opposed to performing grid search for each dataset, we can universally choose a given set of weights that decrease as connectivity information decreases, such as $\{\zeta_t = 1/t\}, t = 1, ..., p$, and observe whether there is still a meaningful improvement in performance without the need for any parameter search. In case there is still need for some optimization, we can at least significantly restrict the parameter distribution, decreasing the amount of time needed for tuning. Specifically, we weight connectivities as being either unimportant ($\zeta = 0$), or important ($\zeta = 1$). This results in the number of tested parameter combinations equaling (2^p)(Size of γ distribution). Ultimately, in practice with 8 filtrations we found that this meant fewer than 1/5 the amount of tested parameter combinations were needed to obtain optimal or near optimal results compared to the original construction. In most cases, however, no parameter search was even necessary at all. In Figure 5, we include a chart illustrating the scale of the respective parameter searches as we vary the number of filtrations.

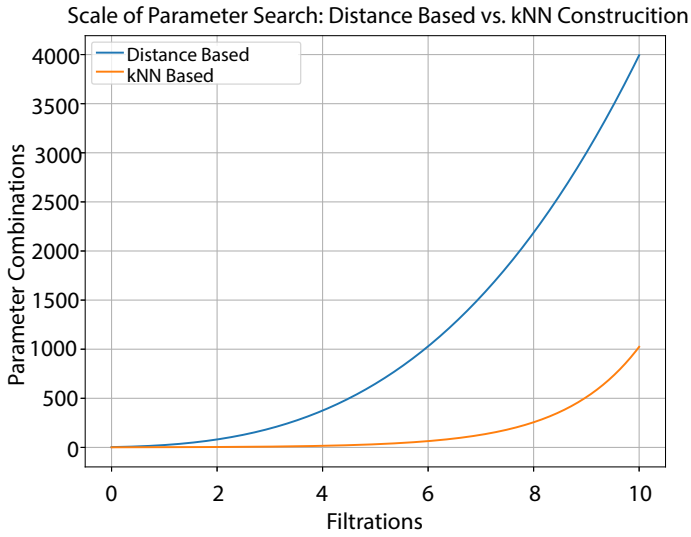

Figure 5: Comparison of the scales of parameter searches necessary between the limited approach to distance based filtrations and the limited approach to kNN-Induced filtrations.

We see from this that, for a reasonable number of filtrations, the parameter search necessary for the kNN construction is a fraction of that needed for the standard construction, while the results listed in Table 4 showcase that the performance is still optimal or at least near optimal compared to other dimensionality reduction techniques.

The β and γ parameters are similarly found via grid search. In Figure 6, we depict how different parameter combinations impact the accuracy of our K-Means clustering. Ultimately, parameter values ranging from 10^{-10} to 10^{10} were found to produce stable results.

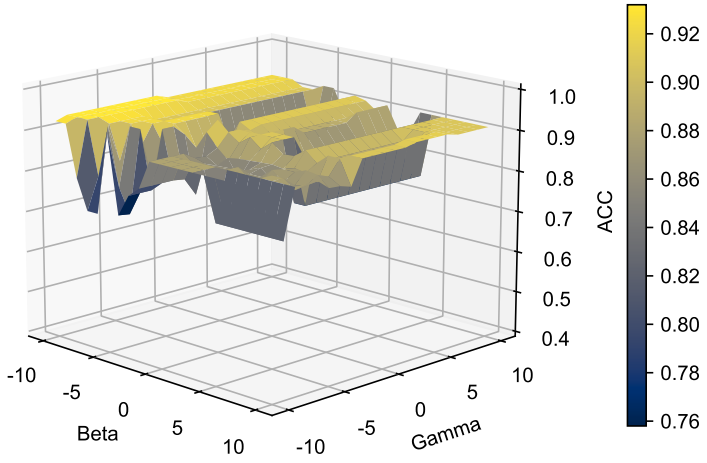

Figure 6: Variations in KMeans accuracy for different combinations of γ and β parameter values

4.2 Visualization of tPCA Eigen-Gene

In the context of scRNA-seq clustering, an eigen-gene refers to the principal components produced by our dimensionality reduction. Eigen-genes summarize the gene expression patterns within a cluster given that they are linear sums of the features which explain the most variation in that cluster. By reducing our data to two dimensions via UMAP or tSNE, we can visualize our eigen-genes in a 2D plot. This visualization can identify the clusters with similar or distinct gene expression profiles.

For a dataset containing, say, 20,000 genes, an aggressive reduction to $k = 2$ dimensions typically results in poorly maintaining the integrity of the data, leading to ineffective visualizations of the eigen-genes. Thus, an important step in the data visualization process is pre-processing. If we first reduce our data to, say, $k = 50$ dimensions via PCA before then reducing again to $k = 2$ dimensions via tSNE or UMAP, we should see an improvement in the representation of our clusters. However, given the associated weaknesses with traditional PCA that we have discussed previously, there may be additional benefit to be gained from pre-processing the data with Topological PCA instead. In Figures 7, 8, 9 we compare the 2D visualizations for several of the tested datasets when pre-processing via PCA and tPCA to assess this improvement in terms of visualization and potential biological insights.

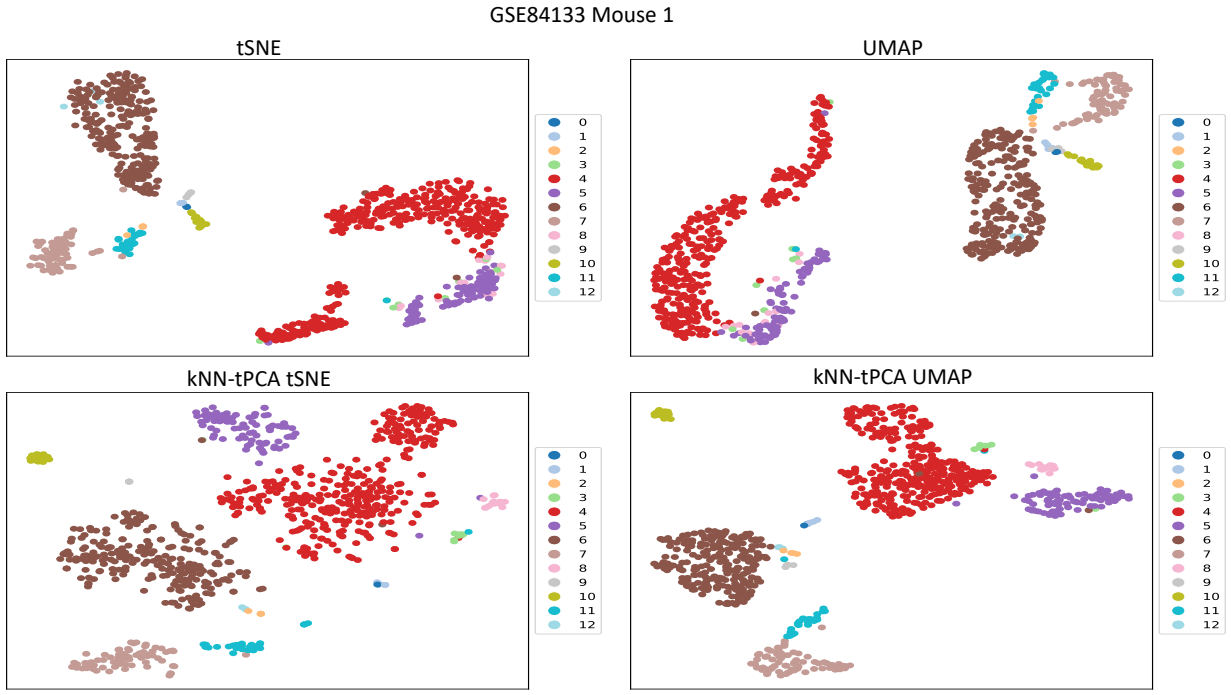

Figure 7: Comparison of visualization techniques between PCA-enhanced tSNE and UMAP, and kNN-tPCA-Enhanced tSNE and UMAP for GSE84133mouse1. Data was log-transformed, with low variance genes removed. For kNN-tPCA-Enhanced tSNE and UMAP, ζ weights were chosen universally as $\{\zeta_t\} = \{1/t\}$ for the tth filtration, and data was reduced to $k = 50$ dimensions. Cells are color coded according to true cell types provided by original authors. Labels 0 through 12 correspond to B cells, T cells, Activated Stellate, α cells, β cells, δ cells, Ductal cells, Endothelial cells, γ cells, Immune cells (other), Macrophage cells, Quiescent Stellate, and Schwann cells respectively.

In Figure 7, we compare visualization techniques for GSE84133mouse1. In Veres et al, β cells were found to have heterogeneity between two distinct subpopulations [46]. However, traditional PCA-enhanced tSNE separates the subpopulations into two clusters that are far away and considerably mixed with δ and other cell types. Our method manages to visualize the cells more similarly, while still displaying the genetic heterogeneity in the population. Furthermore, there is considerably improved separation between the β cells and other cell types, particularly δ cells. For UMAP, we observe that PCA-enhanced UMAP clusters all cell types into two relatively homogeneous clusters. Pre-processing with kNN-tPCA, meanwhile, manages

to separate all cell types fairly well, with the exception of Endothelial and Quiescent Stellate cells. This is likely explained by quiescent stellate cells being located primarily around vascular cells in the pancreas, including Endothelial cells, leading to similar gene expression profiles between the two cell types. Gaining a further understanding of this spatial organization is crucial for understanding the mechanisms underlying pancreatic diseases.

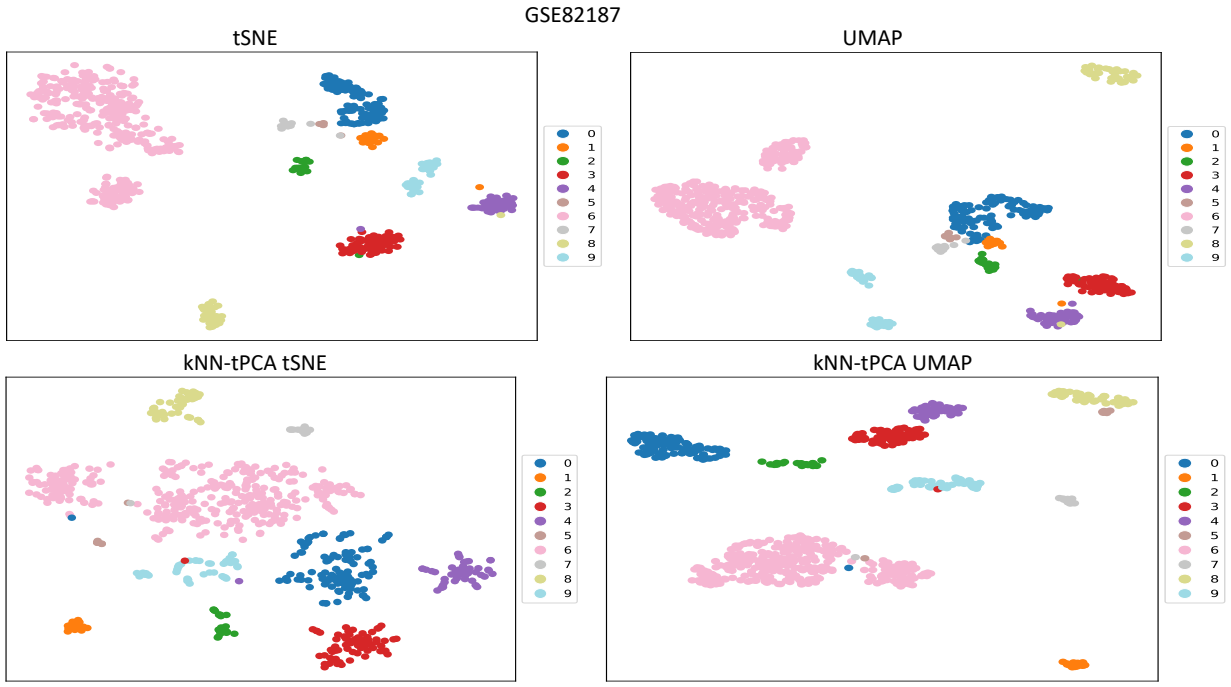

Figure 8: Comparison of visualization techniques between PCA-enhanced tSNE and UMAP, and kNN-tPCA-Enhanced tSNE and UMAP for GSE82187. Data was log-transformed, with low variance genes removed. For kNN-tPCA-Enhanced tSNE and UMAP, ζ weights were chosen universally as $\{\zeta_t\} = \{1/t\}$ for the tth filtration, and data was reduced to $k = 50$ dimensions. Cells are color coded according to true cell types provided by original authors. Labels 0 through 9 correspond to Astro cells, Ependy-C cells, Ependy-Sec cells, Macrophage cells, Microglia cells, NSC cells, Neuron cells, OPC cells, Oligo cells, and Vascular cells respectively.

In Figure 8, we compare visualization techniques for GSE82187. For tSNE as well as UMAP, we note improved separation between Astro, Ependy-C, and OPC cells when pre-processing with tPCA rather than traditional PCA. Furthermore, like with traditional PCA-enhanced UMAP and tSNE, kNN-tPCA pre-processing still enables us to identify the distinct D1 and D2 medium spiny neuron subtypes even when inducing sparsenss in our principal components. In both of our improved visualizations, there seems to be more of a continuous gradient between the subtypes rather than a discrete separation. Continuous gradients indicate that neurons within each subtype lie on a spectrum of gene expression values, with many cells having a range of intermediate expression values. These results are supported by the findings in Gokce et al [44].

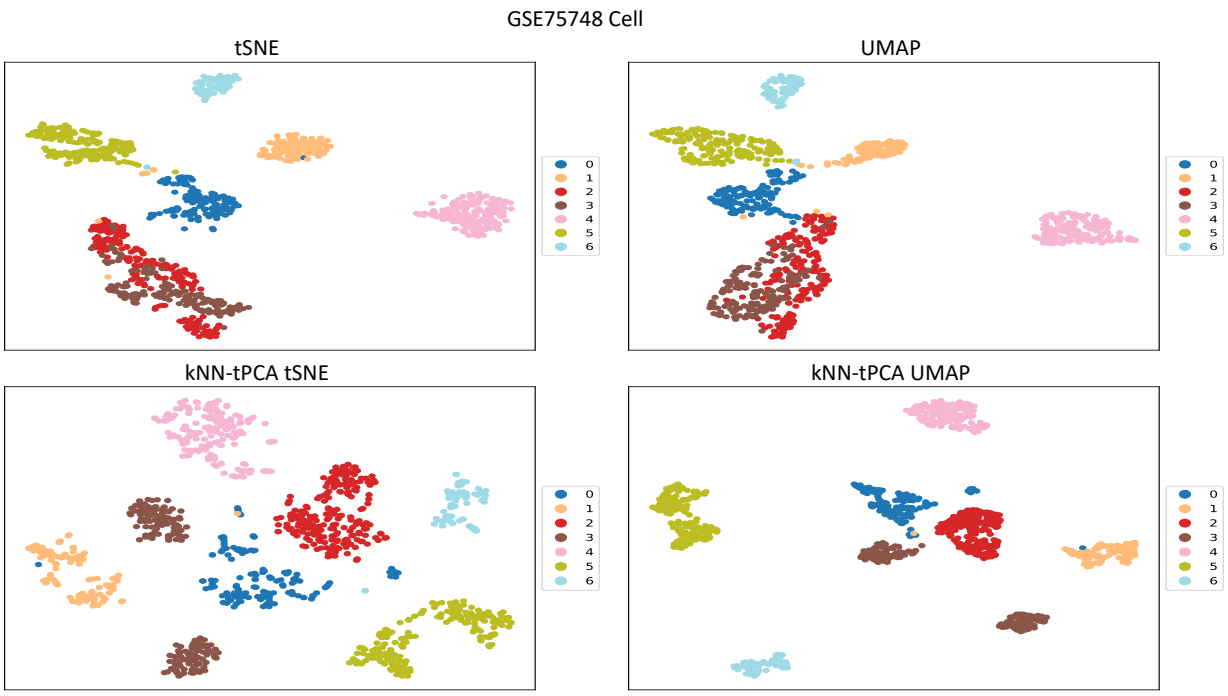

Figure 9: Comparison of visualization techniques between PCA-enhanced tSNE and UMAP, and kNN-tPCA-Enhanced tSNE and UMAP for GSE75748cell. Data was log-transformed, with low variance genes removed. For kNN-tPCA-Enhanced tSNE and UMAP, ζ weights were chosen universally as $\{\zeta_t\} = \{1/t\}$ for the tth filtration, and data was reduced to $k = 50$ dimensions. Cells are color coded according to true cell types provided by original authors. Labels 0 through 6 correspond to DEC cells, EC cells, H1 cells, H9 cells, HFF cells, NPC cells, and TB cells respectively.

In Figure 9, we compare visualization techniques for GSE75748cell. We note for both tSNE and UMAP, the PCA pre-processed version clusters H1 and H9 cells into one homogeneous cluster given the similar gene expression profile of these cells [43]. However, the kNN-tPCA enhanced versions were still able to differentiate these cell types. The same can be said of DEC and EC cells. DEC cells were also found to have lower similarity in their clustering with kNN-tPCA enhanced tSNE, indicating a heterogeneous pool of DEC cells. These results are supported by the findings in Chu et al [43]. In both instances when pre-processing with tPCA, the H9 cells formed two distinct clusters, indicating some kind of possible heterogeneity in the genetic profiles of these cells.

4.3 RS Plot Analysis

To more effectively visualize our gene expression data after dimensionality reduction, we can generate Residue-Similarity plots for some of the tested datasets [21]. We can then compare results for classifying cell types after reducing the data via RgLSPCA and kNN-tPCA. In Figure 10 we produce RS plots for each method on GSE82187 to compare classification accuracy. We observe a significant improvement, particularly in identifying the cell types in panels two and eight, or Ependy-C and Vascular cells respectively. Note specifically that for Ependy-C cells the samples are situated in the top-right corner, indicating a significantly improved cluster boundary separation and inter-cluster similarity in that clustering when utilizing persistent Laplacian regularization.

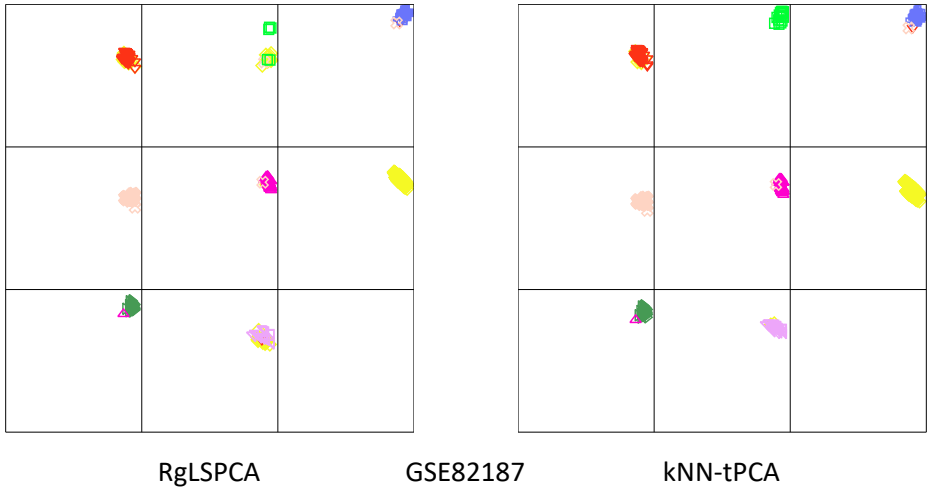

Figure 10: RS plots of clusters generated from RgLSPCA and kNN-tPCA based dimensionality reduction. The x-axis is the residual score, and the y-axis is the similarity score. Each section corresponds to one cluster and the data were colored according to the predicted labels from kNN on the GSE82187 dataset at $k = 100$.

Similarly, for GSE67835 we note a considerable improvement in our ability to correctly identify replicating fetal neurons and Microglia in panels five and six respectively. For Microglia cells in particular, we again observe a significant improvement in the residual score for that clustering, indicating that tPCA yields a greater dissimilarity between these cells and other cell types than RgLSPCA. Specifically, tPCA improves the separation between Microglia and quiescent fetal neurons/OPC cells. In panel one, we note that classification after dimensionality reduction via kNN-tPCA has a slightly greater tendency to misidentify OPC cells with lower similarity scores as Microglia cells, indicating that these cells exhibited a similar gene expression profile, which is supported by the findings in Darmanis et. al. for a subset of the OPC population [42].

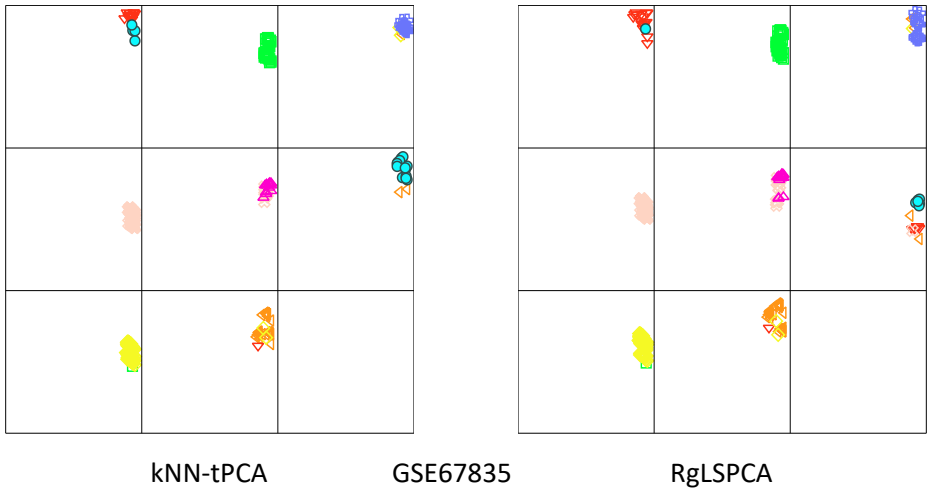

Figure 11: RS plots of clusters generated from RgLSPCA and kNN-tPCA based dimensionality reduction. The x-axis is the residual score, and the y-axis is the similarity score. Each section corresponds to one cluster and the data were colored according to the predicted labels from KNN on the GSE67835 dataset at $k = 100$.

5 Conclusion

Single Cell RNA sequencing technologies have grown considerably in popularity in recent years, and with the ability to reveal vast amounts of information regarding the drivers behind various diseases as well as potential bio-therapeutic targets, effective analysis of this data is of paramount importance in the field of biomedical research. As we have seen, intrinsic high dimensionality of the data introduces computational complexity as well as considerable noise, hindering any meaningful analysis. Thus, dimensionality reduction is a crucial step of the process, and we seek, as always, to maximize the accurate representation of our data in the new, reduced space. To this end, we propose topological PCA for scRNA-seq clustering. This method combines a new robustness via $L_{2,1}$ norm regularization, sparsity constraints, and improved geometrical structure capture via persistent Laplacian regularization.

Extensive benchmark testing on 11 scRNA-seq datasets showcases that our proposed method significantly outperforms other similar PCA enhancements, as well as non-negative matrix factorization, for KMeans clustering after dimensionality reduction. While previous methods such as graph Laplacian Sparse PCA account for sparsity and local geometry preservation, the method is limited by analysis of a simplicial complex at only a single scale. Furthermore, Frobenius norm regularization is sensitive to outliers. The incorporation of a persistent Laplacian term contributes to multi-scale analysis through a sequence of filtrations, as well as persistent homology information derived from the harmonic spectra of our Laplacian matrices. Compared to NMF, we observe an average improvement of 13.53% for NMI and 31.92% for ARI.

While our method achieves superb results for clustering analysis after dimensionality reduction, there is still considerable room for improvement. First, our method considers only $\mathcal{L}_0$ Laplacian, and therefore lacks higher order connectivity information. Furthermore, there remains the work of continuing our parameter analysis, to arrive at a means of optimizing our $\{\zeta\}$ weights which is more efficient and produces more optimal results than simple grid search. While kNN-induced persistent Laplacians seem to be less dependent on parameter tuning, there is still added benefit to examining means of optimizing the performance, and so we should hope to achieve this in a more efficient manner.

6 Data and Model Availability

The data and model used to produce these results can be obtained at the Single Cell Data Processing and RpLSPCA scRNA-seq GitHub Repositories:

Topological PCA GitHub repository: https://github.com/seanfcottrell/Topological-PCA

Single Cell Data Processing GitHub repository: https://github.com/hozumiyu/SingleCellDataProcess

7 Acknowledgments

This work was supported in part by NIH grants R01GM126189, R01AI164266, and R35GM148196, National Science Foundation grants DMS2052983, DMS-1761320, and IIS-1900473, NASA grant 80NSSC21M0023, Michigan State University Research Foundation, and Bristol-Myers Squibb 65109.

References

[1] Aaron TL Lun, Davis J McCarthy, and John C Marioni. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with bioconductor. 2016.

[2] Peter V Kharchenko. The triumphs and limitations of computational methods for scrna-seq. *Nature Methods*, 18(7):723–732, 2021.

[3] Malte D Luecken and Fabian J Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular systems biology*, 15(6):e8746, 2019.

[4] Geng Chen, Baitang Ning, and Tieliu Shi. Single-cell rna-seq technologies and related computational data analysis. *Frontiers in genetics*, page 317, 2019.

[5] Raphael Petegrosso, Zhuliu Li, and Rui Kuang. Machine learning and statistical methods for clustering single-cell rna-sequencing data. *Briefings in bioinformatics*, 21(4):1209–1223, 2020.

[6] Wei Vivian Li and Jingyi Jessica Li. A statistical simulator scDesign for rational scRNA-seq experimental design. *Bioinformatics*, 35(14):i41–i50, 2019.

[7] Tallulah S Andrews, Vladimir Yu Kiselev, Davis McCarthy, and Martin Hemberg. Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data. *Nature protocols*, 16(1):1–9, 2021.

[8] David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J McCarthy, Stephanie C Hicks, Mark D Robinson, Catalina A Vallejos, Kieran R Campbell, Niko Beerenwinkel, Ahmed Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21(1):1–35, 2020.

[9] Mario Flores, Zhentao Liu, Tinghe Zhang, Md Musaddaqui Hasib, Yu-Chiao Chiu, Zhenqing Ye, Karla Paniagua, Sumin Jo, Jianqiu Zhang, Shou-Jiang Gao, et al. Deep learning tackles single-cell analysis—a survey of deep learning for scRNA-seq analysis. *Briefings in bioinformatics*, 23(1):bbab531, 2022.

[10] Ruochen Jiang, Tianyi Sun, Dongyuan Song, and Jingyi Jessica Li. Statistics or biology: the zero-inflation controversy about scRNA-seq data. *Genome biology*, 23(1):1–24, 2022.

[11] Ruiqing Zheng, Min Li, Zhenlan Liang, Fang-Xiang Wu, Yi Pan, and Jianxin Wang. SinNLRR: a robust subspace clustering method for cell type detection by non-negative and low-rank representation. *Bioinformatics*, 35(19):3642–3650, 2019.

[12] Bo Wang, Junjie Zhu, Emma Pierson, Daniele Ramazzotti, and Serafim Batzoglou. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nature methods*, 14(4):414–416, 2017.

[13] Mario Flores, Zhentao Liu, Ting-He Zhang, Md Musaddaqui Hasib, Yu-Chiao Chiu, Zhenqing Ye, Karla Paniagua, Sumin Jo, Jianqiu Zhang, Shou-Jiang Gao, Yu-Fang Jin, Yidong Chen, and Yufei Huang. Deep learning tackles single-cell analysis a survey of deep learning for scrna-seq analysis, 2021.

[14] Jianping Zhao, Na Wang, Haiyun Wang, Chunhou Zheng, and Yansen Su. SCDRHA: A scRNA-seq data dimensionality reduction algorithm based on hierarchical autoencoder. *Frontiers in Genetics*, 12, 2021.

[15] Jiarui Ding, Anne Condon, and Sohrab Shah. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature Communications*, 9, 05 2018.

[16] Zixiang Luo, Chenyu Xu, Zhen Zhang, and Wenfei Jin. A topology-preserving dimensionality reduction method for single-cell rna-seq data using graph autoencoder. *Scientific Reports*, 11:20028, 10 2021.

[17] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[18] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction, 2020.

[19] Raghd Rostom, Valentine Svensson, Sarah Teichmann, and Gozde Kar. Computational approaches for interpreting scRNA-seq data. *FEBS letters*, 591, 05 2017.

[20] Xiaoshuang Shi, Lai Zhihui, Zhenhua Guo, Wan Minghua, Cairong Zhao, and Kong Heng. Sparse principal component analysis via joint l 2,1-norm penalty. volume 8272, pages 148–159, 12 2013.

[21] Yuta. Hozumi, Rui. Wang, and Guo-Wei Wei. CCP: Correlated clustering and projection for dimensionality reduction. *arXiv preprint arXiv:2206.04189*, 2022.

[22] Yuta Hozumi, Kiyoto Aramis Tanemura, and Guo-Wei Wei. Preprocessing of single cell RNA sequencing data using correlated clustering and projection. *Journal of Chemical Information and Modeling*, 0(0):null, 0. PMID: 37402705.

[23] Zhenqiu Shu, Qinghan Long, Luping Zhang, Zhengtao Yu, and Xiao-Jun Wu. Robust graph regularized NMF with dissimilarity and similarity constraints for scrna-seq data clustering. *Journal of Chemical Information and Modeling*, 62(23):6271–6286, 2022. PMID: 36459053.

[24] Thomas Höfer and Chunxuan Shao. Robust classification of single-cell transcriptome data by nonnegative matrix factorization. *Bioinformatics*, 33, 09 2016.

[25] I. Jolliffe. Principal component analysis. *Encyclopedia of statistics in behavioral science*, 2005.

[26] Feiping Nie and Heng Huang. Non-greedy l21-norm maximization for principal component analysis, 2016.

[27] B. Jiang, C. Ding, B. Luo, and J. Tang. Graph-laplacian pca: Closed-form solution and robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3492–3498, 2013.

[28] Sean Cottrell, Rui Wang, and Guowei Wei. PLPCA: Persistent Laplacian enhanced-PCA for microarray data analysis, 2023.

[29] R. Wang, D. Nguyen, and G. Wei. Persistent spectral graph. *International journal for numerical methods in biomedical engineering*, 36(9):e3376, 2020.

[30] Facundo Mémoli, Zhengchao Wan, and Yusu Wang. Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science*, 4(2):858–884, 2022.

[31] Xiaoqi Wei and Guo-Wei Wei. Persistent sheaf laplacians. *arXiv preprint arXiv:2112.10906*, 2021.

[32] Jian Liu, Jingyan Li, and Jie Wu. The algebraic stability for persistent laplacians. *arXiv preprint arXiv:2302.03902*, 2023.

[33] Dong Chen, Jian Liu, Jie Wu, and Guo-Wei Wei. Persistent hyperdigraph homology and persistent hyperdigraph laplacians. *Foundations of Data Science*, doi: 10.3934/fods.2023010, 2023.

[34] Jiahui Chen, Yuchi Qiu, Rui Wang, and Guo-Wei Wei. Persistent laplacian projected Omicron BA.4 and BA.5 to become new dominating variants. *Computers in Biology and Medicine*, 151:106262, 2022.

[35] Yuchi Qiu and Guo-Wei Wei. Persistent spectral theory-guided protein engineering. *Nature Computational Science*, 3(2):149–163, 2023.

[36] Zhenyu Meng and Kelin Xia. Persistent spectral–based machine learning (PerSpect ML) for protein-ligand binding affinity prediction. *Science advances*, 7(19):eabc5329, 2021.

[37] I. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[38] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14, 2001.

[39] J. Chen, R. Zhao, Y. Tong, and G. Wei. Evolutionary de Rham-Hodge method. *Discrete and continuous dynamical systems. Series B*, 26(7):3785, 2021.

[40] R. Wang, R. Zhao, E. Ribando-Gros, J. Chen, Y. Tong, and G. Wei. HERMES: Persistent spectral graph software. *Foundations of data science (Springfield, Mo.)*, 3(1):67, 2021.

[41] Minh Quang Le and Dane Taylor. Persistent homology with k-nearest-neighbor filtrations reveals topological convergence of pagerank, 2022.

[42] Spyros Darmanis, Steven A. Sloan, Ye Zhang, Martin Enge, Christine Caneda, Lawrence M. Shuer, Melanie Hayden Gephart, Ben A. Barres, and Stephen R. Quake. A survey of human brain transcriptome diversity at the single cell level. *Proceedings of the National Academy of Sciences of the United States of America*, 112:7285 – 7290, 2015.

[43] Li-Fang Chu, Ning Leng, Jue Zhang, Zhonggang Hou, Daniel Mamott, David Vereide, Jeea Choi, Christina Kendziorski, Ron Stewart, and James Thomson. Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biology*, 17, 08 2016.

[44] Ozgun Gokce, Ozgun Gokce, Geoffrey M. Stanley, Barbara Treutlein, Norma F. Neff, J. Gray Camp, Robert C. Malenka, Patrick E. Rothwell, Marc Vincent Fuccillo, Thomas C. Südhof, Thomas C. Südhof, Stephen R. Quake, and Stephen R. Quake. Cellular taxonomy of the mouse striatum as revealed by single-cell rna-seq. *Cell reports*, 16 4:1126–1137, 2016.

[45] Alexandra-Chloé Villani, Rahul Satija, Gary Reynolds, Siranush Sarkizova, Karthik Shekhar, James Fletcher, Morgane Griesbeck, Andrew Butler, Shiwei Zheng, Suzan Lazo, Laura Jardine, David Dixon, Emily Stephenson, Emil Nilsson, Ida Grundberg, David McDonald, Andrew Filby, Weibo Li, Philip L. De Jager, Orit Rozenblatt-Rosen, Andrew A. Lane, Muzlifah Haniffa, Aviv Regev, and Nir Hacohen. Single-cell rna-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science*, 356(6335):eaah4573, 2017.

[46] Maayan Baron, Adrian Veres, Sam Wolock, Aubrey Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Ryu, Bridget Wagner, Shai Shen-Orr, Allon Klein, Douglas Melton, and Itai Yanai. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell systems*, 3:346–360, 10 2016.

[47] Qiaolin Deng, Daniel Ramsköld, Björn Reinius, and Rickard Sandberg. Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, 343(6167):193–196, 2014.