

# Exponential-family embedding with application to cell developmental trajectories for single-cell RNA-seq data

Kevin Z. Lin

University of Pennsylvania, Wharton Statistics Department, Philadelphia, PA

Jing Lei

Carnegie Mellon University, Statistics & Data Science Department, Pittsburgh, PA\*

Kathryn Roeder

Carnegie Mellon University, Statistics & Data Science Department, Pittsburgh, PA†

September 25, 2020

## Abstract

Scientists often embed cells into a lower-dimensional space when studying single-cell RNA-seq data for improved downstream analyses such as developmental trajectory analyses, but the statistical properties of such non-linear embedding methods are often not well understood. In this article, we develop the *eSVD* (exponential-family SVD), a non-linear embedding method for both cells and genes jointly with respect to a random dot product model using exponential-family distributions. Our estimator uses alternating minimization, which enables us to have a computationally-efficient method, prove the identifiability conditions and consistency of our method, and provide statistically-principled procedures to tune our method. All these qualities help advance the single-cell embedding literature, and we provide extensive simulations to demonstrate that the eSVD is competitive compared to other embedding methods.

We apply the eSVD via Gaussian distributions where the standard deviations are proportional to the means to analyze a single-cell dataset of oligodendrocytes in mouse brains (Marques et al., 2016). Using the eSVD estimated embedding, we then investigate the cell developmental trajectories of the oligodendrocytes. While previous results are not able to distinguish the trajectories among the mature oligodendrocyte cell types, our diagnostics and results demonstrate there are two major developmental trajectories that diverge at mature oligodendrocytes.

Keywords: *gene expression, oligodendrocytes, latent space models, matrix factorization, random dot product model*

## 1 Introduction

Single-cell RNA-sequencing data give scientists an unprecedented opportunity to analyze the dynamics among individual cells based on their gene expressions, but many analyses require first embedding each cell into a lower-dimensional space as an important preprocessing step in order to make downstream methods more statistically or computationally tractable. For example, these low-dimensional embeddings can be used to visualize high-dimensional data, to control for batch effects, to cluster cells into cell types, to denoise or impute single-cell data, or to estimate trajectories to understand

---

\*This work was supported by National Science Foundation (NSF) grants DMS-2015492 and DMS-1553884.

†This work was supported by National Institute of Mental Health (NIMH) grant R37MH057881 and R01MH123184 and and Simons Foundation Autism Research Initiative (SFARI) grants SF402281 and SF367561.

how cells develop over time; see Sun et al. (2019) for a comprehensive overview. Typically, these embeddings are computed from an  $n \times p$  gene expression matrix, where each of the  $n$  rows and  $p$  columns represent a different cell and a different gene respectively. Two of the most common methods to compute these embeddings, uniform manifold approximation and projection (UMAP, McInnes et al. (2018)) and the singular value decomposition (SVD), have different weaknesses that we strive to remedy in this work. On one hand, UMAP produces flexible, non-linear embeddings that have seen widespread usage for visualization purposes (Becht et al., 2019). However, since UMAP does not yet have proven statistical properties such as consistency, there is a lack of consensus on how to tune this method, and methods that build on these UMAP embeddings inherit this statistical ambiguity; see Cao et al. (2019) and Bergen et al. (2020) for example. On the other hand, the SVD has been extensively studied in the statistical literature, but is often restrictive in practice since it yields only linear embeddings. In this article, we advance the literature by developing the eSVD (exponential-family SVD), a non-linear embedding method that retains desirable statistical properties. As the name suggests, the eSVD is a generalization of the SVD, and embeds each cell in a non-linear fashion into a lower-dimensional space with respect to any one-parameter exponential-family distribution, allowing the researcher to have much broader modeling flexibility. Methodologically, we design the eSVD such that it can be appropriately tuned using matrix-completion ideas. Theoretically, unlike similar work that also bridges this gap between the SVD and UMAP for single-cell applications such as Durif et al. (2017) and Risso et al. (2018), we leverage recent theoretical developments in the nonconvex optimization literature that formalize the statistical properties of the eSVD. With these insights, we use the eSVD to analyze single-cell data<sup>1</sup> in order to demonstrate better downstream analysis results.

To illustrate the importance of embeddings, we focus on analyzing oligodendrocytes – cells that enable rapid transmission of signals by producing myelin and providing metabolic support to neurons in the central nervous system. These cells are intriguing to study due to their constant development throughout a subject’s lifetime, unlike many other cell types that mature at adulthood (Menn et al., 2006). As mentioned in Marques et al. (2016) and Cai and Xiao (2016), understanding how oligodendrocytes develop can lead to new insights into the cause of myelin disorders such as multiple sclerosis and Alzheimer’s disease. We discuss the oligodendrocyte dataset and present a preliminary analysis in Section 2, where we provide various diagnostics demonstrating the shortcomings of the SVD. To better understand this phenomenon, we review the hierarchical model that the SVD implicitly assumes in Section 3. Specifically, suppose a hierarchical model where each cell and each gene has its own low-dimensional latent random vector. In the language of exponential-family distributions, this model assumes that the cell’s expression of a particular gene is a one-parameter exponential-family random variable whose natural parameter is the inner product of the two corresponding latent vectors. By formulating this hierarchical model, we see that the SVD implicitly assumes a Gaussian distribution with constant variance. However, this assumption is often violated since the variance of each cell’s gene expressions is observed to vary dramatically with their mean expression level (Love et al., 2014; Hicks et al., 2017). Hence, as we will review later, there is a rich line of work that extends hierarchical models of this type to analyze single-cell data by replacing the Gaussian distribution with more appropriate exponential-family distributions (Pierson and Yau, 2015; Townes et al., 2017; Durif et al., 2017; Risso et al., 2018), of which this article continues.

The aforementioned work often add additional nuances on top of the hierarchical model in order to model single-cell data better, but this often results in complicated estimators that become too intractable to statistically analyze. Hence, we design the eSVD in such a way that the posited statistical model retains the most important aspects common to the aforementioned work, while we can leverage recent theoretical developments to analyze the estimator’s statistical properties. Specifically, the eSVD uses alternating minimization, a popular and computationally efficient approach used in the matrix factorization literature to solve the nonconvex optimization problem at hand,

---

<sup>1</sup>We use the term “single-cell data” to refer to single-cell RNA-sequencing data specifically in the remainder of this article.

described in Section 4. We present the eSVD’s statistical theory in Section 5, which builds upon the theoretical analyses of Zhao et al. (2015) and Lei (2018). These statistical properties include identifiability conditions and consistency, which ensure that researchers well-understand the estimated embedding and have a solid statistical foundation to build downstream analyses on top of. However, to ensure that the eSVD does not sacrifice too much modeling flexibility for theoretical tractability, we compare the eSVD to competing methods used to analyze single-cell data in Section 6 using synthetic data.

Finally, we return to our preliminary analysis of oligodendrocytes in Section 7, where we show that the eSVD embedding improves our analysis of cell developmental trajectories to match the latest scientific findings. These trajectories explain the heterogeneity among the oligodendrocytes by describing the smooth transition of gene expression among individual cells along a continuum, reflecting the cells’ gradual transcriptional changes during development (Trapnell et al., 2014). Although early research suggest oligodendrocytes develop along a single trajectory (Kessaris et al., 2006), recent work suggest that oligodendrocytes could potentially branch out into various mature types (Marques et al., 2016; van Bruggen et al., 2017; Marques et al., 2018). Our improved analysis match these findings – we show the eSVD embedding estimates two distinct trajectories. We develop visualization tools to show our developmental trajectory findings, and conclude in Section 8 with practical extensions and theoretical questions left open for future work. While we focus on using the eSVD embedding to estimate cell developmental trajectories in this article, we emphasize that this embedding can be used for other applications highlighted earlier in this section, and provide additional analyses on another single-cell dataset in the appendix.

## 2 Preliminary analysis

We analyze a dataset of oligodendrocytes from mice brains collected by Marques et al. (2016) as a prototypical example to demonstrate shortcomings of the SVD embedding when applied to single-cell data. This dataset, henceforth called the Marques dataset, contains the gene expression of 5,069 oligodendrocytes that are clustered into thirteen cell sub-types using a biclustering algorithm (Zeisel et al., 2015) in Marques et al. (2016). These thirteen cell sub-types were later grouped into six major cell types and manually labeled based on cell-type specific marker genes (Zhang et al., 2014), shown in Figure 1.. We preprocess the data by selecting 983 highly informative genes, normalizing each cell by its library size (i.e., total counts across all genes), and  $\log_2$ -transforming each entry. These details are described in Appendix C. As suggested in the literature, it is common to  $\log_2$ -transform the gene expression matrix prior to using the SVD, since the  $\log_2$ -transformation can ideally stabilize the variance (Townes et al., 2017; Butler et al., 2018). However, as we see in the preliminary analysis in this section, this analysis strategy will result in modeling concerns that we wish to remedy in the rest of this article.

We review the SVD embedding, as it provides motivation for the eSVD in the next section. Let  $A \in \mathbb{R}^{n \times p}$  represent the observed single-cell RNA-sequencing data matrix with rank  $m$ , where  $n$  is the number of cells and  $p$  is the number of genes. Here, loosely speaking, each entry  $A_{ij}$  measures how many instances of genetic material for gene  $j$  is observed for cell  $i$  after pre-processing. Let the SVD of  $A$  be denoted as  $\hat{U}\hat{D}\hat{V}^\top$  where  $\hat{U} \in \mathbb{R}^{n \times m}$  and  $\hat{V} \in \mathbb{R}^{p \times m}$  are both orthonormal matrices and  $\hat{D} \in \mathbb{R}^{m \times m}$  is a diagonal matrix. For a given latent dimensionality  $k \leq m$ , the SVD embedding for each cell  $i \in \{1, \dots, n\}$  (denoted as  $\hat{X}_i \in \mathbb{R}^k$ ) and each gene  $j \in \{1, \dots, p\}$  (denoted as  $\hat{Y}_j \in \mathbb{R}^k$ ) is defined as

$$\hat{X}_i = \left(\frac{n}{p}\right)^{1/4} \cdot \left(\sqrt{\hat{D}_{1,1}} \cdot \hat{U}_{i,1}, \dots, \sqrt{\hat{D}_{k,k}} \cdot \hat{U}_{i,k}\right), \quad \text{for } i \in \{1, \dots, n\} \quad (2.1)$$

$$\hat{Y}_j = \left(\frac{p}{n}\right)^{1/4} \cdot \left(\sqrt{\hat{D}_{1,1}} \cdot \hat{V}_{j,1}, \dots, \sqrt{\hat{D}_{k,k}} \cdot \hat{V}_{j,k}\right), \quad \text{for } j \in \{1, \dots, p\}. \quad (2.2)$$

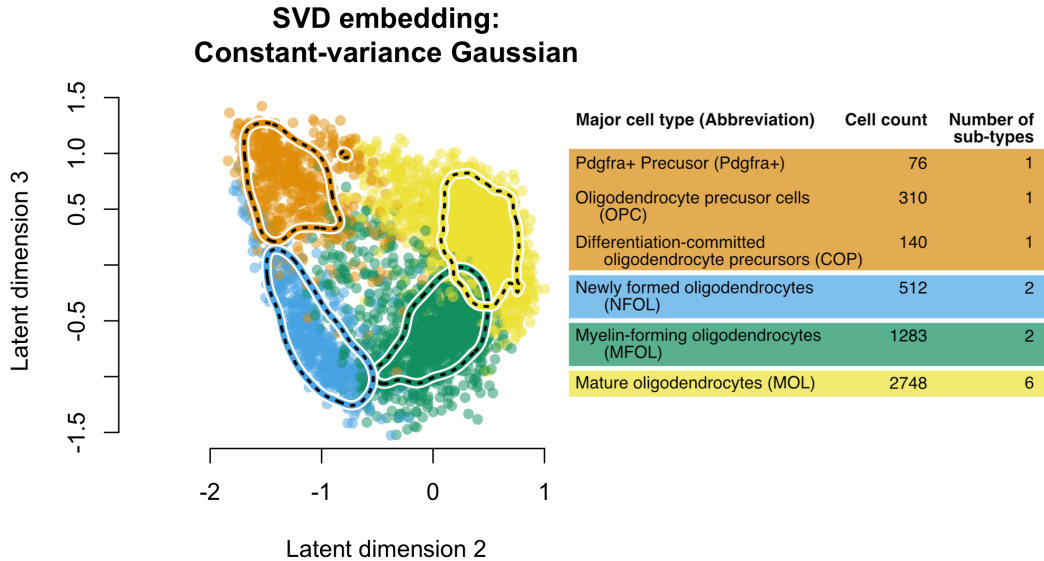


Figure 1: The SVD embedding of the oligodendrocytes from Marques et al. (2016) after preprocessing including a  $\log_2$ -transformation, shown alongside a table summarizing the cell types. The six major cell types are listed in the table with the number of cells in each type, along with how they are differentiated into the thirteen different cell sub-types. The rows are organized from the “youngest” cell types to “most mature” cell types from top to bottom. The youngest three major cell types are colored orange, while the oldest three are colored blue, green and yellow respectively. The second and third latent dimensions are shown on the left, along with contours of the estimated densities to visualize high-density regions (one for each color of cells).

We can see that the SVD embedding is a linear embedding since  $\hat{X}_i$  is the first  $k$  elements of the  $i$ th row in  $(n/p)^{1/4} \cdot A\hat{V}\hat{D}^{-1/2}$ . A scatterplot of the second and third latent dimensions of such an embedding is shown in Figure 1. Later in this article, we will show that this embedding implicitly assumes a constant-variance Gaussian distribution in Section 3, and show that this particular formulation handles identifiability concerns discussed in Section 4.

Now, we show that the SVD embedding (and its equivalent reparameterizations) does not model the data well, which could produce misleading results in downstream analyses. First, we visualize the quality of fit of the SVD embedding by purposefully omitting a small subset of randomly-selected entries in  $A$  and estimating the embedding as a matrix-completion problem. We can then assess the quality of fit of the embedding by comparing the values of these omitted entries in  $A$  to their predicted values. Figure 2 demonstrates this diagnostic, where the left plot shows the observed values in  $A$  that are not omitted (i.e., the “training set”) versus their respective predicted values, while the right plot shows the observed values that are omitted (i.e., the “testing set”) versus their respective predicted values. This missing-value diagnostic is commonly used both for assessing the quality of fit as well as for model selection (Li et al., 2020), and we will return to it in detail in Section 4. We see that while the embedding’s performance on the testing set is more-or-less equivalent to that on the training set, the variability decreases as the gene expression increases. This is in opposition with the working model in the literature that suggests that larger gene expressions should be more variable than smaller ones (Witten, 2011; Risso et al., 2018). In fact, prior to taking the  $\log_2$ -transformation, Figure 3 demonstrates that the variance in gene expression increases with its mean, reinforcing this model. Combined, all the diagnostics demonstrate that applying a  $\log_2$ -transform and using the SVD in conjunction seem to distort the properties of the data. This inspires us to develop a



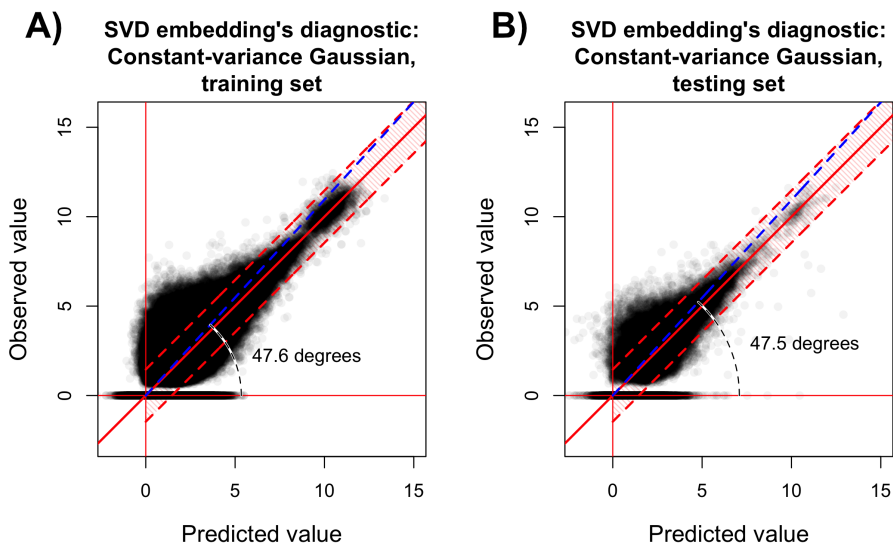


Figure 2: Diagnostic based on matrix completion to assess the fit using the SVD embedding for either the observed values that are not omitted (i.e., the “training set”) (A) or the observed values that are purposefully omitted (i.e., the “testing set”) (B), both verses their respective predicted values. This embedding is estimated using softImpute (Mazumder et al., 2010). The shaded red region is centered around the identity function (the ideal mean function) and marks the 10th to 90th quantiles of the constant-variance Gaussian model (based on the empirical variance) for different values of the predicted mean. The blue dotted line represents the principal angle between the observed values and their predicted value counterparts, where we mark its divergence from the identity function’s  $45^\circ$ . More details of this diagnostic is discussed in Section 4, while details of the fitting process using softImpute can be found in Appendix C.

more appropriate embedding method that still retains desirable statistical properties. In the next section, we review the optimization problem that the SVD solves and see how it can be extended to one-parameter exponential families more generally, which will motivate our method, the eSVD.

### 3 Statistical model and background

In this section, we explain the random dot product model that we investigate in this article, and its relation to other work.

#### 3.1 Statistical model and estimation strategy

We model the entries of the single-cell dataset  $A \in \mathbb{R}^{n \times p}$  as conditionally independent random variables drawn from a random dot product model – a latent hierarchical model commonly used in other work (Pierson and Yau, 2015; Townes et al., 2017; Durif et al., 2017; Risso et al., 2018). Specifically, for an appropriate one-parameter exponential-family distribution  $F$  parameterized by its natural parameter  $\theta_{ij}$ , we impose model,

$$A_{ij} \sim F(\theta_{ij} = X_i^\top Y_j), \quad \text{for } (i, j) \in \{1, \dots, n\} \times \{1, \dots, p\},$$

where  $X_1, \dots, X_n \stackrel{i.i.d.}{\sim} G$ , and  $Y_1, \dots, Y_p \stackrel{i.i.d.}{\sim} H$ . (3.1)

where  $G$  and  $H$  represent two latent  $k$ -dimensional distributions, where  $k$  is much smaller than  $n$  or  $p$ . We assume all the latent random vectors  $X_i$ ’s and  $Y_j$ ’s are jointly independent, and the observed

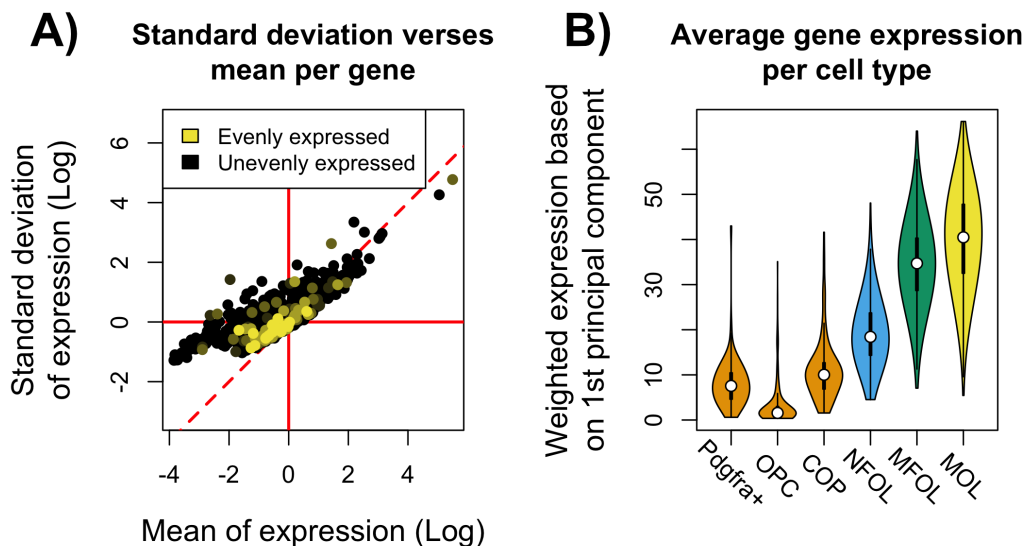


Figure 3: (A) The standard deviation of the expression versus the mean expression across all the cells, where each point represents one of the 983 genes in the preprocessed single-cell dataset. The color of each point depends on how evenly the gene is expressed among each of the six oligodendrocyte cell types show in Figure 1. The solid red horizontal and vertical lines and the dashed red line denoting the line  $y = x$  are for visual reference. (B) Violin plot of the average expression of the genes reweighted according to the first principal component among the six oligodendrocyte cell types, using the color scheme in Figure 1. The statistics in both plots are computed prior to taking the  $\log_2$ -transformation, and is shown on the logarithm scale in Plot A purely for visualization purposes. More details about these plots are in Appendix C.

$A_{ij}$ 's are independent conditioned on the  $X_i$ 's and  $Y_j$ 's. Let the density of the exponential-family distribution  $F$  be denoted as

$$p(A_{ij} | \theta_{ij}) = h(A_{ij}) \exp(T(A_{ij})^\top \eta(\theta_{ij}) - g(\theta_{ij})), \quad (3.2)$$

where  $g(\cdot)$  is a known log-partition function for  $F$  with a domain  $\mathcal{R}$ ,  $\eta(\cdot)$  is a known natural parameter function, and  $T(\cdot)$  is a known sufficient statistic function. For notational convenience, we denote  $X \in \mathbb{R}^{n \times k}$  and  $Y \in \mathbb{R}^{p \times k}$  as the matrices that collect all the latent vectors  $X_1, \dots, X_n$  and  $Y_1, \dots, Y_p$  row-wise, and denote  $\Theta = XY^\top \in \mathbb{R}^{n \times p}$  as the rank- $k$  natural parameter matrix that collects all elements  $\theta_{ij}$ . Given the exponential-family form for  $F$  shown in (3.2), we need to impose the following assumption to ensure the inner products  $X_i^\top Y_j$  for all  $(i, j) \in \{1, \dots, n\} \times \{1, \dots, p\}$  yield valid natural parameters,

**Assumption 3.1** (Bounded inner product). *Let  $\mathcal{R}$  denote the domain of the natural parameters for the distribution  $F$ . Assume that for any  $X_i \sim G$  and  $Y_j \sim H$ ,*

$$\mathbb{P}(X_i^\top Y_j \in \mathcal{R}) = 1, \quad \text{almost surely.}$$

Given the above model, our goal is to estimate the latent random vectors  $X_1, \dots, X_n$  since these latent vectors represent the low-dimensional embedding of all  $n$  cells. For a given exponential-family distribution  $F$ , an intuitive strategy for estimating our desired embedding is to minimize the negative log-likelihood based on the observed data  $A$  over all possible vectors  $X_1, \dots, X_n$  and  $Y_1, \dots, Y_p$ . Specifically, plugging into the exponential-family form (3.2) into the model (3.1), we

derive the loss function

$$\mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{(i,j)} \left[ g(X_i^\top Y_j) - T(A_{ij})^\top \eta(X_i^\top Y_j) \right], \quad (3.3)$$

with the constraints  $X_i^\top Y_j \in \mathcal{R}$  for all pairs  $(i, j)$ . The above loss function is nonconvex, but if  $F$  is the the constant-variance Gaussian distribution, this loss function is proportional to

$$\frac{1}{np} \sum_{(i,j)} (A_{ij} - X_i^\top Y_j)^2,$$

which is specifically what the SVD minimizes (Maezika, 2016). This particular model is convenient to use since the SVD provides a closed-form solution to the corresponding nonconvex optimization problem, and leads to the SVD embedding shown in (2.1).

### 3.2 Relation to other work modeling single-cell data

As we have discussed previously in Section 2, this constant-variance assumption is too restrictive to properly model single-cell data. Hence, many articles cited above replace the constant-variance Gaussian distribution with other exponential-family distributions for  $F$  that allow the variance to increase with the mean. For example, Witten (2011) and Risso et al. (2018) consider the Poisson and negative binomial distribution specifically, after a suitable transformation of the natural parameters. Additionally, these models often add other random effects on top of the existing random dot product model (3.1) that influence the entries in  $A$ . For example, many methods like pCMF (Durif et al., 2017) allow researchers to incorporate *dropout* into the model – a characteristic of single-cell data where a substantial fraction of the gene expression for a cell is recorded as exactly 0 due to low amounts of RNA in the cell (Kharchenko et al., 2014). Other methods such as ZINB-WaVE (Risso et al., 2018) go further and allow covariate information such as gene length and cell size. Most recently, Lopez et al. (2018) use deep autoencoders to estimate the embedding. However, there is often a lack of theoretical analyses for the aforementioned estimators. This is because replacing the exponential-family distribution  $F$  with any distribution aside from the constant-variance Gaussian distribution leads to non-trivial nonconvex estimators that minimize the loss function shown in (3.3), which make traditional statistical techniques for analyzing these estimators unsuitable. Therefore, concerns such as identifiability are typically not addressed theoretically, leading to ambiguity on performance of the estimators of such models.

To resolve this theoretical ambiguity, we design the eSVD to estimate the embedding based on the random dot product model (3.1) for any exponential-family distribution  $F$  with no other random effects. In this way, we can tractably analyze the statistical properties of eSVD while retaining abundant flexibility to effectively model single-cell data.

### 3.3 Matrix factorization

To the best of our knowledge, the first statistical results for estimators that extended the SVD to generic exponential-family distributions by minimizing a loss function similar to (3.3) come from Gunasekar et al. (2014) and Lafond (2015). There, the authors minimize the loss function over the natural parameter matrix  $\Theta$  and add a trace penalization term to encourage the estimate to be low-rank. While this formulation yields a convex optimization problem, it requires solving a semidefinite program which can be computationally prohibitive for large datasets. This consideration has motivated researchers to investigate the statistical properties of estimators that minimize the loss function (3.3) directly as a nonconvex optimization problem. Specifically, *alternating minimization* is a suitable candidate for this task, where each iteration alternates between optimizing either one of two low-rank matrices  $X$  and  $Y$  while treating the other fixed. This algorithmic strategy pre-dates

the convex relaxation approach; see Collins et al. (2002), Jain et al. (2013), Udell et al. (2016), Landgraf and Lee (2019) and the references within for discussions and additional variants. From an algorithmic standpoint, our method is a direct continuation of such work. However, the statistical properties of such estimators have only recently been characterized rigorously. For example, to accommodate the constraints in Assumption 3.1, Wang et al. (2016), Yu et al. (2020) and Chi et al. (2019) adapt the theoretical framework to study slightly different estimators based on alternating projected gradient descent. In contrast, our work will build on techniques used in Zhao et al. (2015) and Balakrishnan et al. (2017) to retain our focus on alternating minimization.

However, all the aforementioned theoretical results do not directly apply the random dot product model (3.1), which contains an additional source of randomness induced by the hierarchical structure. In contrast, our theory is able to account for this additional source of randomness by drawing upon connections to the network literature. Specifically, the random dot product model (3.1) is similar to those used in latent position random graphs studied in the network literature (Hoff et al. (2002) and Athreya et al. (2017)). Hence, we draw inspiration from Lei (2018) on how to address these identifiability concerns and to develop proof techniques in this article.

There are many other embedding methods in the literature more broadly, such as non-negative matrix factorization, kernel PCA, and manifold-based embedding methods such as UMAP and Isomap. We defer a thorough discussion contrasting eSVD with such estimators to Appendix B.

## 4 Method: eSVD (Exponential-family SVD)

We describe the eSVD in this section, which is designed to be a general framework to minimize the loss function (3.3) for any choice of a one-parameter exponential-family distribution  $F$ . To keep the presentation clear, we describe some of the more nuanced implementation details in Appendix B. We also describe an important diagnostic to assess the quality of fit, as demonstrated in Section 2. This diagnostic can be also be used as a tuning procedure to select the most appropriate choice for  $F$  or nuisance parameters.

### 4.1 eSVD and its application to the curved Gaussian distribution

Similar to other nonconvex matrix factorization methods (Wang et al., 2016; Yu et al., 2020), our method requires an initial estimate of the rank- $k$  matrix of natural parameters,  $\hat{\Theta}'$ , where  $k$  is pre-determined. To achieve this, we use an initialization method based on Wang et al. (2016). To simplify the presentation here, we provide details in Appendix B.1. This initialization scheme performs a rank- $k$  SVD based on transforming each entry of  $A$  via the inverse of the log-partition function  $g(\cdot)$ . Given this initial estimate, consider its SVD  $\hat{\Theta}' = UDV^\top$ . To start the alternating minimization stage of our method, we set  $\bar{Y}^{(0)} = V$ .

After initialization, the eSVD then refines the estimate by performing alternating minimizations. Denoting a generic matrix and its SVD by  $\hat{\Theta} = UDV^\top$ , let  $\text{LeftSVD}(\hat{\Theta}) = U$ , the function that maps a matrix to its left singular vectors. Then, for iterations  $t \in \{0, \dots, T-1\}$ ,

$$X^{(t+1)} = \underset{X \in \mathbb{R}^{n \times k}}{\operatorname{argmin}} \mathcal{L}_n(X, \bar{Y}^{(t)}) : X_i^\top \bar{Y}_j^{(t)} \in \mathcal{R}, \quad \forall (i, j), \quad (4.1)$$

$$\bar{X}^{(t+1)} = \sqrt{n} \cdot \text{LeftSVD}(X^{(t+1)}), \quad (4.2)$$

$$Y^{(t+1)} = \underset{Y \in \mathbb{R}^{p \times k}}{\operatorname{argmin}} \mathcal{L}_n(\bar{X}^{(t+1)}, Y) : (\bar{X}_i^{(t+1)})^\top Y_j \in \mathcal{R}, \quad \forall (i, j), \quad (4.3)$$

$$\bar{Y}^{(t+1)} = \sqrt{p} \cdot \text{LeftSVD}(Y^{(t+1)}). \quad (4.4)$$

After all  $T$  iterations, the eSVD outputs the final estimate after a reparameterization. That is,

letting  $\widehat{\Theta}^{(T)} = \overline{X}^{(T)}(Y^{(T)})^\top$  have a rank- $k$  SVD of  $\widehat{U}\widehat{D}\widehat{V}^\top$ , the final estimates are

$$\widehat{X}_i = \left(\frac{n}{p}\right)^{1/4} \cdot \left(\sqrt{\widehat{D}_{1,1}} \cdot \widehat{U}_{i,1}, \dots, \sqrt{\widehat{D}_{k,k}} \cdot \widehat{U}_{i,k}\right), \quad i = 1, \dots, n, \quad (4.5)$$

$$\widehat{Y}_j = \left(\frac{p}{n}\right)^{1/4} \cdot \left(\sqrt{\widehat{D}_{1,1}} \cdot \widehat{V}_{j,1}, \dots, \sqrt{\widehat{D}_{k,k}} \cdot \widehat{V}_{j,k}\right), \quad j = 1, \dots, p. \quad (4.6)$$

This is the same reparameterization used in (2.1) and (2.2).

**Remarks about algorithmic design.** We make a few remarks about the design of our algorithm. Optimizing over  $X$  and  $Y$  directly raises identifiability issues, since for any orthogonal matrix  $Q$ ,  $\mathcal{L}_n(XQ, YQ^\top) = \mathcal{L}_n(X, Y)$ . To address this, Ge et al. (2017) append a penalty term

$$\frac{1}{8} \|X^\top X - Y^\top Y\|_F^2,$$

while Zhao et al. (2015) use the QR-decomposition between iterations, and we use the LeftSVD( $\cdot$ ) operator. In practice, we found all three choices behave similarly. The factors  $\sqrt{n}$  and  $\sqrt{p}$  in (4.2) and (4.4) are included for theoretical reasons to ensure the spectrum of the Hessian is well-controlled and to ensure the values do not underflow if  $n$  or  $p$  are too large empirically. Also, the final reparameterizations in (4.5) and (4.6) are designed such that the sample second-moment matrices of  $\widehat{X}$  and  $\widehat{Y}$  are both equal and diagonal, i.e.,

$$\frac{1}{n} \widehat{X}^\top \widehat{X} = \frac{1}{p} \widehat{Y}^\top \widehat{Y},$$

which is important for our statistical analysis later on.

Lastly, to perform the constrained optimization (4.1) and (4.3), we use a first-order method called Frank-Wolfe (Jaggi, 2013), which we found more stable compared to using projected gradient descent. While there are theoretical guidelines for choosing step-sizes related to the convexity and smoothness for this method, we found these choices often led to poor empirical performance.

**Example with the curved Gaussian distribution.** To make the eSVD's workings more concrete, we demonstrate what the minimization in (4.1) would entail when we set  $F$  to be the *curved Gaussian* distribution. This will be useful later in this article when we use this distribution to analyze the oligodendrocytes. Specifically, we say the random variable  $A_{ij}$  follows a curved Gaussian distribution with a known parameter  $\tau > 0$  if  $A_{ij} \sim N(\mu_{ij}, \mu_{ij}^2/\tau^2)$ , for an unknown mean parameter  $\mu_{ij} > 0$  (Efron et al., 1978; Liu and Martin, 2020)<sup>2</sup>. This sets the standard deviation to be linearly proportional to the mean. Writing this distribution in exponential-family form (3.2) yields,

$$p(A_{ij} \mid \theta_{ij}) = \frac{\tau \exp(-\tau^2/2)}{\sqrt{2\pi}} \exp\left(\left[\begin{array}{c} \tau^2 A_{ij} \\ -\tau^2 A_{ij}^2/2 \end{array}\right]^\top \left[\begin{array}{c} \theta_{ij} \\ \theta_{ij}^2 \end{array}\right] + \log(\theta_{ij})\right), \quad (4.7)$$

where the relation between the natural parameter  $\theta_{ij}$  and the mean parameter  $\mu_{ij}$  can be derived to be  $\mu_{ij} = 1/\theta_{ij}$ . Here, the domain of the natural parameters would be  $\mathcal{R} = \mathbb{R}_+$ , the positive half-line. After simple calculations, one can derive the negative log-likelihood and conclude that the minimization in (4.1) becomes

$$X^{(t+1)} = \operatorname{argmin}_{X \in \mathbb{R}^{n \times k}} \frac{1}{np} \sum_{(i,j)} \left[ -\log(X_i^\top \overline{Y}_j^{(t)}) - \left[\begin{array}{c} \tau^2 A_{ij} \\ -\tau^2 A_{ij}^2/2 \end{array}\right]^\top \left[\begin{array}{c} X_i^\top \overline{Y}_j^{(t)} \\ (X_i^\top \overline{Y}_j^{(t)})^2 \end{array}\right] \right].$$

<sup>2</sup>We call it a curved Gaussian distribution since this distribution is a curved exponential-family distribution.

Analogous calculations for other common distributions are shown in Appendix B.

The curved Gaussian distribution is relevant in practice since if  $\tau \geq 2$ , this distribution reflects the phenomenon that genes with larger expression also exhibit larger variance, while most of the distribution’s mass is still positive. Additionally, in many instances, this distribution can capture more variability than the Poisson and negative binomial distributions, which can be beneficial when the single-cell data is intrinsically noisier. In general however, if the researcher wants to use the eSVD for an arbitrary one-parameter exponential-family distribution  $F$ , all she needs to pass into our implementation is the computation of the loss function (3.3), its gradients, and information about the domain  $\mathcal{R}$ .

## 4.2 Matrix-completion diagnostic and tuning procedure

We provide the following diagnostic to assess the embedding’s quality of fit or to determine which choice of  $F$  is most appropriate for our data, which was used in Figure 2. Inspired by network cross-validation work such as Li et al. (2020), we use matrix completion to determine the quality of our model fit. As alluded to in Section 2, to do this, we omit a small percentage of the entries of  $A$  when estimating the embedding and compare these values to their predicted expected value counterparts. To compute this expected value, recall that for exponential-family distributions (3.2),  $g'(\cdot)$  (the derivative of the log-partition function  $g(\cdot)$ ) maps the natural parameter to the expected value. We note that we are able to adopt this matrix completion strategy to tune our method since our alternating minimization procedure can be adapted to handle missing values. In contrast, embedding methods like ZINB-WaVE (Risso et al., 2018) and PCMF (Durif et al., 2017) do not offer an analogous tuning procedure. This procedure is formalized below.

1. For bootstrap trials  $b \in \{1, \dots, B\}$ :
  - (a) Randomly sample  $m$  of the entries of  $A$ , denoted as  $\mathcal{O} = \{(i_1, j_1), \dots, (i_m, j_m)\}$ , which will be omitted in the following estimation step. Here,  $m$  can be any small number, such as  $\lceil 0.01 \cdot (np) \rceil$ .
  - (b) Estimate the latent vectors by  $\hat{X}$  and  $\hat{Y}$  according to Subsection 4.1 where the loss function (3.3) omits the entries in  $\mathcal{O}$  and is parameterized based on the desired distribution of  $F$ .
  - (c) Compute  $v_1$ , defined as the leading eigenvector of the matrix formed by the omitted observed values  $A_{\mathcal{O}} = \{A_{i_1, j_1}, \dots, A_{i_m, j_m}\}$  and their predicted expected value counterparts  $g'(\hat{X}\hat{Y}^\top)_{\mathcal{O}} = \{g'(\hat{X}_{i_1}^\top \hat{Y}_{j_1}), \dots, g'(\hat{X}_{i_m}^\top \hat{Y}_{j_m})\}$ .
  - (d) Compute model fit quality,  $q^{(b)}$  defined as the angle between  $v_1$  and the vector  $(1, 1)$ , representing the identity function.
2. Average the model fit qualities across all trials,  $q^{(1)}, \dots, q^{(B)}$ .

Observe that we define the quality of fit  $q^{(b)}$  above by how much the leading eigenvector  $v_1$  deviates from  $45^\circ$ . This angle of  $v_1$  is what we called the *principal angle* in Figure 2. Having an eigenvector’s angle close to  $45^\circ$  means that on average, the predicted values correspond closely with the observed value. The advantage of this quality-of-fit’s definition is that it is easily comparable even across different distributions  $F$ , unlike the negative log-likelihood or MSE.

While we advocate constructing plots such as Figure 2 to obtain a more holistic sense of how well the embedding fits the data in general, we can also use the above procedure to obtain an automated model selection method in the following way – if we try the above diagnostic for multiple distributions for  $F$ , the distribution that yields the smallest average of  $q^{(1)}, \dots, q^{(B)}$  is deemed the most appropriate model for  $A$ . In this way, we can also use this diagnostic as a tuning procedure to select the dimensionality of the latent space  $k$  or nuisance parameters for exponential-family distributions



such as  $\tau$  in the curved Gaussian distribution (4.7) in a grid-search fashion. An additional variant of this tuning procedure is detailed in Appendix B.

## 5 Statistical theory

In this section, we prove the consistency for the eSVD when applied to the random dot product model (3.1), which is important to ensure that the eSVD is estimating a well-defined quantity. This result provides the needed statistical foundation for downstream tasks such as clustering and RNA velocity, as mentioned in Section 1. Additionally, our theory gives us better insights into the eSVD since our analysis also reveals the identifiability conditions that formalize to what degree the embedding can be estimated. While our theorems currently assume the correctly-specified setting (i.e., the eSVD's choice of  $F$  matches the true generating distribution and  $k$  is correctly-specified), we hope these theorems provide a roadmap for future work to prove analogous statements for broader settings or for more complex methods like ZINB-WaVE which currently do not exist.

We discuss the additional notation here. For a generic matrix  $A$ , let  $\|A\|_F$  denote its Frobenius norm. For two sequences  $a_n$  and  $b_n$  and two random sequences  $A_n$  and  $B_n$ , let  $a_n = O(b_n)$  and  $A_n = O_P(B_n)$  denote that  $a_n/b_n$  or  $A_n/B_n$  is bounded for large enough  $n$  deterministically or in probability respectively. For all  $i \in \{1, \dots, n\}$  and  $j \in \{1, \dots, p\}$ , let the the population second moment matrices of  $X_i$  and  $Y_j$  and the corresponding eigen-decompositions be defined respectively as

$$\mathbb{E}[X_i X_i^\top] = C_X^* = \Phi^* \Lambda^* \Phi^{*\top}, \quad \text{and} \quad \mathbb{E}[Y_j Y_j^\top] = C_Y^* = \Psi^* \Gamma^* \Psi^{*\top}.$$

Our proposition below requires the following assumptions.

**Assumption 5.1** (Sub-Gaussian distribution of latent vectors). *Assume that  $X_i$  for all  $i \in \{1, \dots, n\}$  are i.i.d. sub-Gaussian random vectors. That is, there exists a fixed constant  $D$  such that for any vector  $v \in \mathbb{R}^k$  where  $\|v\|_2 = 1$  and any integer  $c \geq 1$ ,  $(\mathbb{E}[|X_i^\top v|^c])^{1/c} \leq Dc^{1/2}$ , and a similar assumption holds for  $Y_j$  for all  $j \in \{1, \dots, p\}$  with also the same constant  $D$ .*

**Assumption 5.2** (Second moment properties). *First, assume the population second moment matrices  $C_X^*$  and  $C_Y^*$  are equal and are both diagonal matrices, where  $(C_X^*)_{i,i} \geq (C_X^*)_{j,j}$  for any  $1 \leq i < j \leq k$ . Second, assume there exists positive numbers  $c_1 \leq c_2$  and  $1 < \alpha \leq \beta$  such that for all  $\ell \in \{1, \dots, k\}$ , the eigenvalues satisfy*

$$c_1 \ell^{-\alpha} \leq \lambda_\ell^* \leq c_2 \ell^{-\alpha}, \quad \text{and} \quad \lambda_\ell^* - \lambda_{\ell+1}^* \geq c_1 \ell^{-\beta},$$

with the convention that  $\lambda_{k+1}^* = 0$ .

Both assumptions are common in work that study the spectrum associated with random dot product models (Lei, 2018). Assumption 5.1 assumes  $G$  and  $H$  are sub-Gaussian distributions, which enables sharp rates for estimating their second-moment matrices  $C_X^*$  and  $C_Y^*$  respectively (Vershynin, 2012). On the other hand, the second part of Assumption 5.2 enables our estimator to accurately estimate its eigenvalues and eigenvectors. Importantly however, the first part of Assumption 5.2 can be interpreted instead as an identifiability condition, which we formalize below.

**Proposition 5.1.** *Given two  $k$ -dimensional distributions  $G$  and  $H$ , each with at least two moments where the population second moment matrices are full rank, consider two independent random variables  $X' \sim G$  and  $Y' \sim H$ . Then there exists a linear and invertible transformation  $R$  such that the population second moment matrices of  $X = RX'$  and  $Y = R^{-\top} Y'$  are the same, i.e.,*

$$\mathbb{E}[X X^\top] = \mathbb{E}[Y Y^\top].$$

Furthermore, both population second moment matrices of  $X$  and  $Y$  are diagonal matrices.

The proof is in Appendix I, which provides an explicit construction of the matrix  $R$ . Note that since  $R$  is invertible, we guarantee that the distribution of the inner product is preserved, i.e.,

$$\mathbb{P}((X')^\top Y' \leq t) = \mathbb{P}(X^\top Y \leq t), \quad \forall t \in \mathcal{R}.$$

Hence, the first part of Assumption 5.2 can be interpreted as an identifiability condition, since we can only estimate  $G$  and  $H$  only up to a linear transformation.

**Proposition 5.2.** *Assume the model in (3.1) where Assumptions 5.1 and 5.2 hold. If the estimator  $\hat{\Theta}$  satisfies  $\|\Theta - \hat{\Theta}\|_F \leq \epsilon$  conditioned on  $X$  and  $Y$ , and  $k = o(\min\{n, p\})$ , then up to sign,<sup>3</sup> eSVD achieves the rate after reparameterizations (4.5) and (4.6),*

$$\frac{1}{n} \|X - \hat{X}\|_F^2 = O_P\left(\max\left\{\frac{k^{4\beta-\alpha+4}}{\min\{n, p\}}, \frac{k^{2\beta-\alpha+2} \max(\epsilon^2, \epsilon)}{np}\right\}\right). \quad (5.1)$$

**Discussion of consistency.** Assuming  $k$  is fixed, the above proposition states that the eSVD embedding is consistent as long as  $\epsilon$  (the rate of convergence for the matrix of natural parameters,  $\|\Theta - \hat{\Theta}\|_F$ ) is faster than  $O_P(\sqrt{np})$ . We formalize this statement in Appendix D, where we add assumptions common to the literature such as strong convexity and smoothness of the negative log-likelihood function associated with  $F$ . The details are deferred because these assumptions are technical to describe and detract from the main text. More generally, however, the above proposition addresses the additional source of randomness induced by the random dot product model mentioned in Section 3 that other theoretical investigations typically do not address. Therefore, *any* such estimator equipped with a rate for  $\|\Theta - \hat{\Theta}\|_F$  can be plugged into Proposition 5.2 directly. Additionally, note that a similar rate (5.1) holds for estimating  $Y$ , the embedding of the genes.

**Application to curved Gaussian model.** While Proposition 5.2 and the results in Appendix D apply for any generic one-parameter exponential-family distribution  $F$  satisfying certain conditions, we specifically apply these results to the curved Gaussian distribution (4.7) in Appendix E to demonstrate what the rates are for a particular exponential-family distribution. At a high level, we show that when  $n$  and  $p$  grow asymptotically at the same rate and  $k$  is fixed,

$$\frac{1}{n} \|X - \hat{X}\|_F^2 = O_P\left(\frac{\log^{1/2}(n)}{n^{1/2}}\right).$$

## 6 Numerical study

In this section, we study the performance of the eSVD and other competitive methods based on synthetic data. Our setup for all the simulations in this section are as follows: based on model (3.1), we set the dimensionality to  $k = 2$  and sample  $X_1, \dots, X_n$  i.i.d. uniformly from four connected linear segments (which we call the “trajectories”) with additive Gaussian noise, as illustrated in Figure 4. These four segments loosely represent four cell types. We also sample  $Y_1, \dots, Y_p$  i.i.d. from a mixture of two Gaussians. These sampling procedures represent  $G$  and  $H$  respectively, up to identifiability conditions. We enforce  $X_i^T Y_j \in \mathcal{R}$  for all pairs  $(i, j)$ . The distribution family  $F$  varies among different simulations. We do not use R packages such as Splatter (Zappia et al., 2017) to generate our synthetic data because we want to have precise control over the true embedding. The full details of the simulation setups and usage of various estimators in this section are in Appendix F.

<sup>3</sup>We use “up to sign” similar to Fan et al. (2018), where each column of  $\hat{X}$  can be multiplied by  $\pm 1$  since the SVD is not unique.

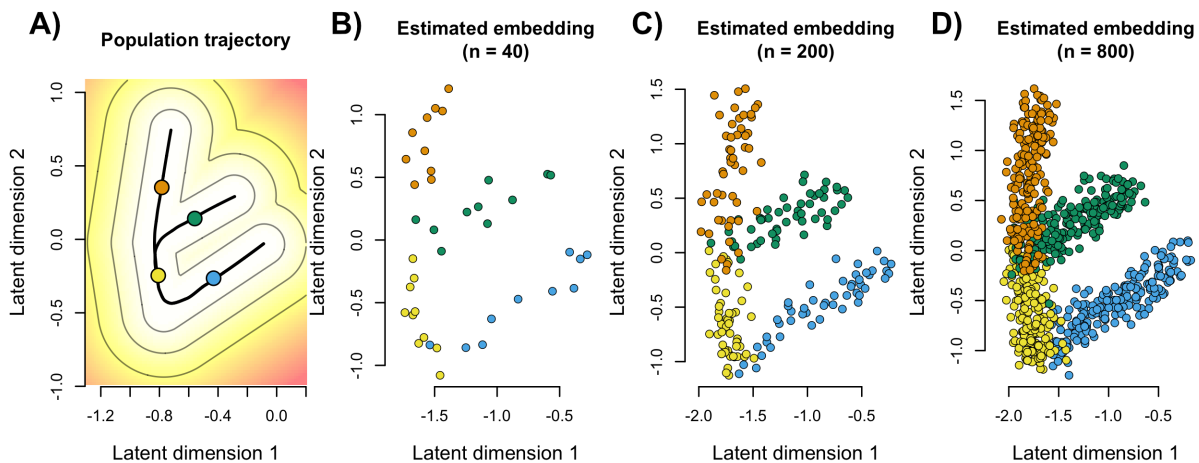


Figure 4: (A) The two-dimensional population density of  $G$ , visualized as a heat map with contour lines of the density along with the true “trajectories” (black lines). The mean vector for each of four cell types are labeled in a different color (blue, yellow, green, orange). (B to D) The estimated embedding  $\hat{X}_1, \dots, \hat{X}_n$  of the synthetically-generated  $A$  for varying levels of  $n$ , (i.e., number of cells or number of rows), colored by the true cell type, which are labels used only for visual reference and not used during estimation.

**Consistency of the estimated embedding.** In this first simulation suite, we demonstrate that the estimated embedding converges towards the true embedding. Specifically, we generate  $A \in \mathbb{R}^{n \times p}$  where each entry  $A_{ij}$  is sampled independently from the negative binomial distribution with a natural parameter  $\theta_{ij} = \bar{X}_i^\top Y_j$  and  $r = 50$ , and fit the eSVD using the correctly specified model. Figure 4 is an illustration that demonstrates the asymptotic properties of the eSVD. Specifically, we see that the distribution of the embedding  $\hat{X}_1, \dots, \hat{X}_n$  approximates  $G$  as  $n$  increases. We provide details and additional results that verify the consistency of the eSVD’s embedding in Appendix F.

**Comparison of different embedding methods.** In our second simulation suite, we demonstrate that the cells’ latent positions estimated by the eSVD are more accurate in relation to one another than those estimated by other methods. Here, we fix  $n = 200$  and  $p = 400$ . We compare the eSVD via the negative binomial distribution to nine other methods commonly used to embed single-cell data: ZINB-WaVE (Risso et al., 2018), pCMF (Durif et al., 2017), SVD, non-negative matrix factorization (NMF), independent component analysis (ICA), UMAP (Becht et al., 2019), t-SNE (Maaten and Hinton, 2008), Isomap (Tenenbaum et al., 2000) and diffusion map (Haghverdi et al., 2015). The first two methods and our tuning procedures are explained in Appendix F. Importantly, SVD implicitly assumes  $F$  is a constant-variance Gaussian distribution as mentioned in Section 3, while ZINB-WaVE and pCMF assume  $F$  is a negative binomial and Poisson distribution respectively.

We simulate data from a negative binomial model in this simulation, which is the distribution that is most commonly used to model sequencing data (Love et al., 2014). Specifically, we sample the observed count matrix  $A$  conditionally independent on  $X_1, \dots, X_n$  and  $Y_1, \dots, Y_p$  where  $A_{ij}$ ’s are sampled from a negative binomial distribution with natural parameter  $X_i^\top Y_j$  and dispersion parameter  $r = 50$ . Then, when we estimate the embedding using eSVD, we use the tuning procedure mentioned in Section 4 to select the most appropriate value of the dispersion parameter  $r$  from the set  $\{5, 50, 100\}$ .

We find that on average across 100 trials, our method estimates the relative latent positions of each cell to be more accurate than other methods (Figure 5A). To define our notion of accuracy, consider each cell  $i$  and its Euclidean distance to all other  $n - 1$  cells in the latent space in both the true and estimated embedding. We then compute the Kendall’s tau correlation between these

two vectors, which only relies on the ranks of the distances, and then average this value over all  $n$  cells. Hence, a high averaged Kendall’s tau value suggests the latent positions of the  $n$  cells are well-estimated with respect to one another. We call this notion of accuracy the *relative embedding correlation*. We define our notion of accuracy in this way to ensure it is insensitive to arbitrary rotations or constant rescalings of the embedding. Figure 5B compares the different estimated embeddings to the true embedding as an illustration. Both the eSVD and ZINB-WaVE estimate embeddings where the four cell types are relatively in the correct configuration, and their accuracy are quite high. For the other methods, the high variability due to the overdispersion of the negative binomial seems to dramatically skew the embedding for certain cells. We defer additional simulations using other distributions  $F$  to Appendix F.

**Investigation using misspecified models and the effect of  $k$ .** In our third simulation suite, we empirically compare the eSVD to other methods when  $F$  is *misspecified*. We also demonstrate how different values of  $k$  can affect the quality of the embedding. While the theorems we developed in Section 5 do not currently handle such settings, these results help us understand how the eSVD performs in more realistic and more challenging scenarios. Due to space constraints, we defer these results to Appendix F, which show that the eSVD can roughly estimate the relative positions of the cells’ embedding well compared to other methods despite the model misspecification. All-in-all, our takeaway message is that the eSVD’s flexibility in choosing which exponential-family distribution  $F$  to use and the diagnostics provided in Section 4 allow our method to remain competitive among the ten methods.

## 7 Single-cell analysis

We return to modeling the Marques dataset (Marques et al., 2016), as described in Section 2, to determine if the embedding based on the curved Gaussian distribution (4.7) is more appropriate than that based on the constant-variance Gaussian distribution, and if so, investigate how the embedding affects the downstream trajectory analysis. As alluded to in Section 2, the six major cell types in Figure 1 have a determined ordering, starting from Pdgfra+ precursors and ending with the mature oligodendrocytes. Our goal in this analysis is to estimate the trajectories among the cell sub-types constrained to this ordering. For example, in Marques et al. (2016), after embedding the cells into a latent space, the authors estimate one developmental trajectory connecting the first five major cell types starting from the Pdgfra+ precursors, but do not definitively conclude how the six mature oligodendrocyte cell sub-types differentiate. Instead, they relied on analyzing the percentage of different cell types across different brain regions to hypothesize that these six sub-types differentiate into multiple different trajectories.

### 7.1 Details of estimating cell developmental trajectories

We provide more details on how we estimate the developmental trajectories based on the low-dimensional embedding  $\hat{X}_1, \dots, \hat{X}_n$ . As alluded in Section 1, these trajectories show how these different cell sub-types develop from one to another, assuming the latent vectors  $\hat{X}_i$  gradually change along the trajectories. Trajectory analyses are an important step in studying the cellular dynamics from single-cell data, as most single-cell technologies provide only a snapshot of all the cells. This is because most technologies destroy the cells during the sequencing step, which prevent longitudinal studies. In this article, we use Slingshot (Street et al., 2018) (with minor modifications) to estimate these cell developmental trajectories. Roughly speaking, Slingshot is a two-step algorithm that requires the latent vectors to already be clustered, where we treat each cell sub-type as a cluster. In the first stage, Slingshot estimates the number of trajectories and ordering of the cell sub-types based on minimizing the distances between cell sub-type centers via a minimum spanning tree. In

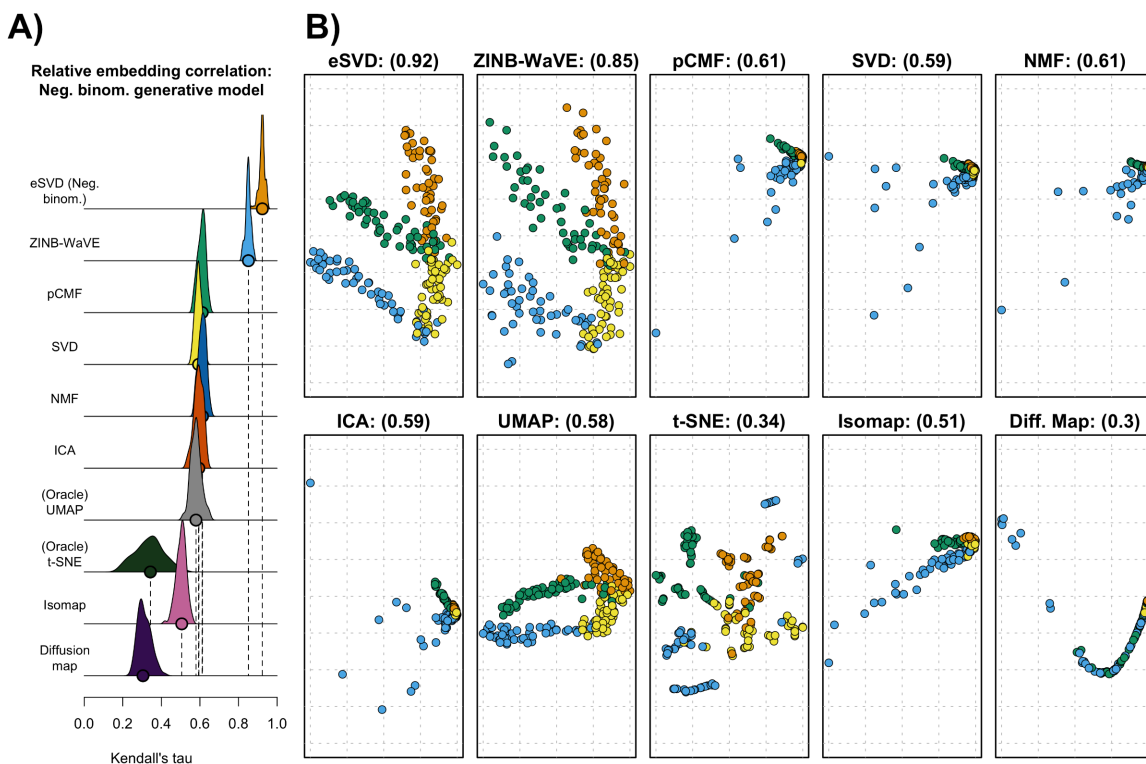


Figure 5: (A) The density plot of each embedding methods' accuracy (eSVD, ZINB-WaVE, pCMF, SVD, NMF, ICA, UMAP, t-SNE, Isomap and diffusion map), based on the relative embedding correlation. The circles along each method's x-axis denotes the median accuracy across the 100 trials. Here, both the data-generating distribution  $F$  and the eSVD use the negative binomial distribution. See Appendix F to see how we tuned the methods. (B) The ten estimated embeddings, chosen among the trial with the median accuracy, noted in each plot's title in parenthesis. The coloring of the samples persists from Figure 4. The x- and y-axes represent the coordinate system estimated by the respective embedding methods, reflected in the dashed grids.

the second stage, Slingshot fits variants of principal curves (Hastie and Stuetzle, 1989) that pass through the cell sub-type centers in the estimated ordering. These principal curves can be thought of as smooth curves that pass through high-density regions in the latent space. Throughout our analysis in this section, we apply Slingshot to the embedding using all latent dimensions, but only visualize the estimated trajectories with respect to the first three latent dimensions. More details about Slingshot and our modifications of it are given in Appendix G.

We briefly mention that the original study (Marques et al., 2016) uses the Monocle algorithm (Trapnell et al., 2014), to estimate the cell developmental trajectories. We use Slingshot instead as it is the current state-of-the-art method based on extensive benchmarking comparisons in Saelens et al. (2019).

## 7.2 Analysis using the constant-variance Gaussian distribution

Building on the analysis in Section 2, we perform a trajectory analysis using the SVD embedding shown in Figure 1 on the  $\log_2$ -transformed data, which assumes the constant-variance Gaussian model. Applying Slingshot directly to this embedding results in two trajectories, both heavily overlapping one another when visualized (Figure 6A). These results are similar to Marques et al.



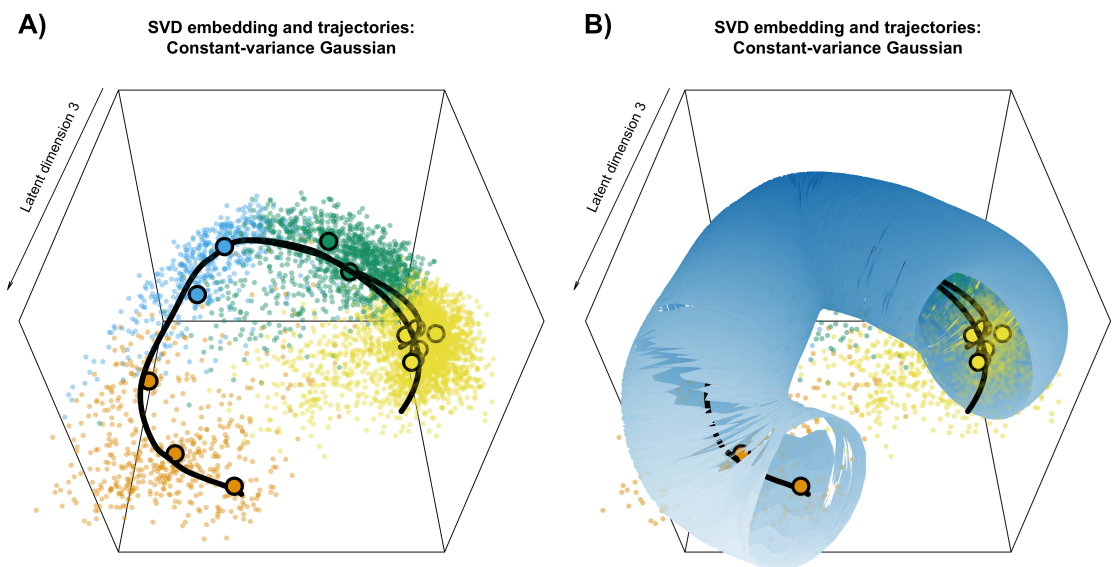


Figure 6: (A) Three-dimensional plot of the estimated latent positions via the SVD embedding with the two estimated cell developmental trajectories laid on top, corresponding to the data shown in the Figure 1. The thirteen bolded points correspond to the cluster centers of the thirteen cell sub-types, where the color scheme persists from Figure 1. (B) The uncertainty tube overlaid on top of Figure A.

(2016) in two ways. First, the authors show that all cells develop from *Pdgfra*<sup>+</sup> precursors to myelin-forming oligodendrocytes in the same way, which we estimate as well. Second, the authors do not definitively conclude if the mature oligodendrocytes diverge in their development. Our trajectories themselves also leave this ambiguity unresolved due to the heavy overlap between the two trajectories. However, we perform the following additional visual diagnostic to quantify if these two trajectories are well approximated by a single trajectory.

To formalize to what degree the different trajectories are the same, we use a bootstrap resampling procedure to construct a uniform uncertainty tube around each trajectory. These tubes capture the variance of each estimated trajectory, and plotting these tubes is a useful descriptive tool. This is an important tool for our analysis because Slingshot is sensitive to small perturbations in the data due to its graph-based strategy to estimate the ordering of the cell sub-types. Specifically, small variations can dramatically change the number of estimated trajectories or ordering of cell sub-types within those trajectories. Hence, our procedure to construct these uniform uncertainty tubes first samples with replacement from all embedded cells within each of the thirteen cell sub-types. For each bootstrap sample, we apply Slingshot to estimate a new set of trajectories. We then compute the  $\ell_2$  distance between the new trajectories and the original trajectories. After applying this procedure multiple times, the 95% quantile of the  $\ell_2$  distances determines the uniform radius of the uncertainty tube, centered around the original trajectory. More details of this procedure are in Appendix G. Based on this construction, both trajectories lie in a single uncertainty tube (Figure 6B); hence, we conclude there is effectively one trajectory that connects all thirteen cell sub-types. This result explains why previous work such as Marques et al. (2016) had difficulty explaining how mature oligodendrocytes differentiate in their trajectory analysis.

### 7.3 Analysis using the curved Gaussian model

The above conclusions, however, rest on the questionable constant-variance Gaussian distributional assumption (see Figure 3). As we have seen in Figure 2, our matrix-completion diagnostic suggests



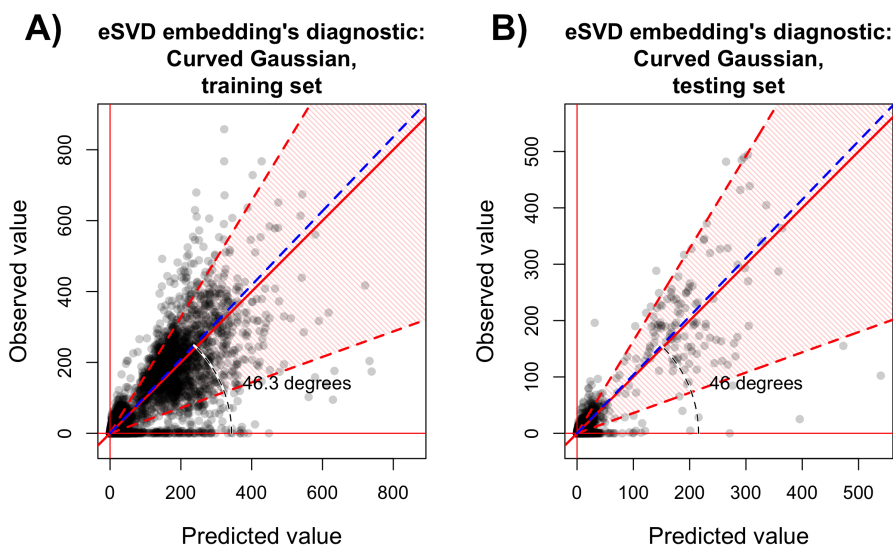


Figure 7: Diagnostic based on matrix completion to assess the fit using the eSVD embedding via the curved Gaussian model with  $k = 5$  and  $\tau = 2$  for either the observed values that are not omitted (i.e., the “training set”) (A) or the observed values that are purposefully omitted (i.e., the “testing set”) (B), both verses their respective predicted values. Both plots are comparable to those in Figure 2. Specifically, the 10th to 90th quantiles of the curved Gaussian model is marked by the shaded red region.

that this assumption is not suitable for modeling the oligodendrocyte dataset at hand.

This finding motivates us to analyze the data using the eSVD to embed each cell with respect to the curved Gaussian distribution (4.7), and to re-examine the resulting diagnostics. Following suggestions from articles like Risso et al. (2018) and Durif et al. (2017), we no longer  $\log_2$ -transform the entries of  $A$  for our eSVD analysis, but rather model the counts in  $A$  directly after accounting for the library size. Based on our tuning procedure, the curved Gaussian distribution with  $k = 5$  and  $\tau = 2$  best fits the data, determined among a grid of candidate values. When we plot the resulting diagnostic for the eSVD in Figure 7, we obtain results that suggest a much better fit compared to that of the SVD. Specifically, the variance is appropriately increasing with the mean, unlike the trend shown in Figure 2. We conclude that the curved Gaussian model (without a  $\log_2$ -transformation) is more appropriate than the constant-variance Gaussian model (with a  $\log_2$ -transformation) for modeling our oligodendrocyte dataset.

We visualize the eSVD embedding in Figure 8 alongside its estimated trajectories and uncertainty tubes, and we see two distinct trajectories that differentiate among the mature oligodendrocytes. Specifically, when we apply Slingshot to the eSVD embedding, we find that we still retain the conclusion that all cells from  $\text{Pdgfra}^+$  precursors to myelin-forming oligodendrocytes develop in the same way, similar to Marques et al. (2016). However, in contrast to that work, we are now able to observe substantial differentiation among the mature oligodendrocytes, with two distinct trajectories supported by the uncertainty tubes. Specifically, within this major cell type, only one of the six mature oligodendrocytes sub-types is shared between the two trajectories. Among the five remaining mature oligodendrocytes sub-types, three sub-types branch off in one trajectories while two sub-types branch into the other trajectory. This is in contrast with the analysis using the SVD embedding where all the estimated trajectories lay within one uncertainty tube (Figure 6B). We show additional plots corresponding to these results, as well as follow-up analyses and diagnostics of the oligodendrocytes using UMAP or ZINB-WaVE as well as plots based on the highly informative genes in Appendix H.

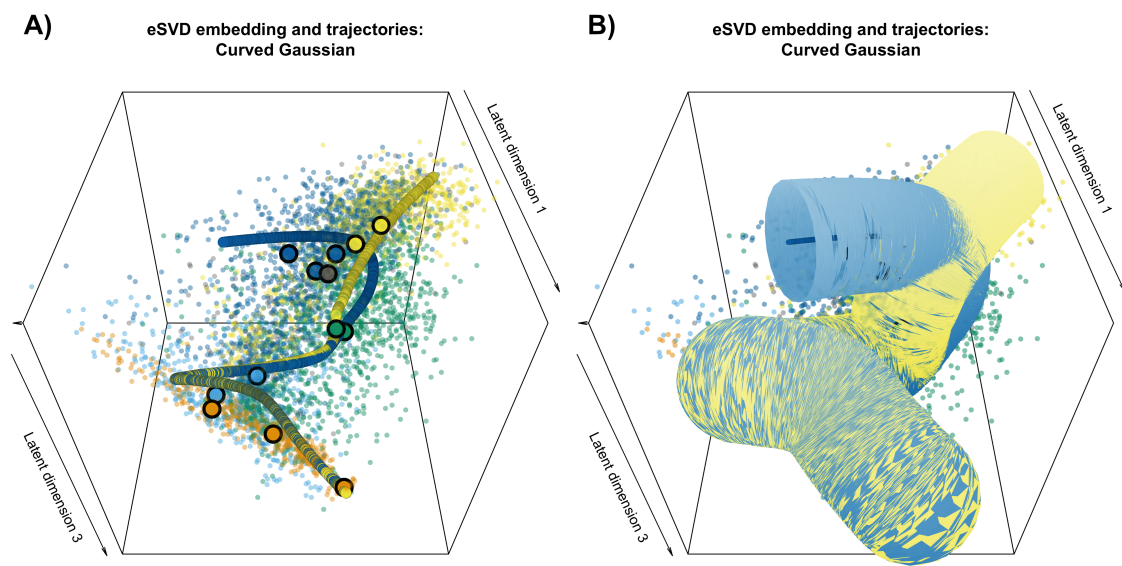


Figure 8: (A) Three-dimensional plot of the estimated latent positions via the eSVD embedding with the curved Gaussian distribution for  $k = 5$  and  $\tau = 2$  with the estimated cell developmental trajectory laid on top. The thirteen bolded points correspond to the cluster centers of the thirteen cell sub-types. The two estimated cell developmental trajectories are colored in yellow and blue. These correspond with the two of six mature oligodendrocytes cell sub-types unique to one trajectory (colored in yellow) and three mature oligodendrocytes cell sub-types unique to the other trajectory (colored in blue). The remaining mature oligodendrocyte cell sub-type is common to both trajectories, prior to the branching (colored in gray). The coloring of cells of other cell types persists from Figure 1. (B) The uncertainty tubes overlaid on top of Figure A. Both plots are comparable to Figure 6.

In summary, from the diagnostic (Figure 7), we conclude that the curved Gaussian distribution is more appropriate for the Marques data, and using this model we identify two distinct developmental trajectories (Figure 8). This is an improvement from the analysis in Marques et al. (2016) which suggested multiple trajectories, but was not able to directly verify this conjecture. Our comparison of the results obtained using the SVD versus the eSVD embeddings can help explain why previous scientific findings suggest that oligodendrocytes effectively follow a single developmental trajectory, while newer analyses based on more flexible statistical models (van Bruggen et al., 2017; Marques et al., 2018) suggest multiple trajectories.

We include an analysis of the single-cell dataset released in Zeisel et al. (2015) in Appendix H to demonstrate eSVD's performance in a different setting. There, the downstream task is to cluster cells rather than to infer developmental trajectories.

## 8 Discussion

In this article, we develop an estimator to non-linearly embed the cells in a single-cell RNA-sequencing dataset into a lower dimensional space with respect to a random dot product model where the inner product of two latent vectors is the natural parameter of a one-parameter exponential-family distribution  $F$ . This embedding method can greatly improve the estimation of cell developmental trajectories overall because it can handle distributions beyond the constant-variance Gaussian distribution, both in theory and practice. While the spirit of such embedding is not new, our contribution is two-fold. First, we develop the eSVD, an alternating minimizing estimator which is computationally efficient and also enables both a tuning procedure based on matrix completion and

a theoretical investigation of its statistical properties such as identifiability and consistency. Second, we apply our estimator to analyze the oligodendrocytes in mouse brains, and our results coincide with recent scientific hypotheses (van Bruggen et al., 2017; Marques et al., 2018).

For future work, we plan to further the eSVD both in its modeling flexibility in practice as well as its theoretical properties, as we believe embeddings based on the random dot product model are appealing for single-cell analyses. Specifically, we plan to extend the eSVD to model the dropout effect, allow different nuisance parameters for each gene, or incorporate the library size directly into the statistical model directly. These trends occur in work such as Witten (2011), Pierson and Yau (2015), Townes et al. (2017) and Risso et al. (2018). While the models within these investigations are more flexible, we reiterate that their corresponding estimators were previously often believed to be too complicated to analyze from a theoretic perspective. Therefore, we are interested in studying how flexible our methods can be while still retaining tractability for theoretical analyses or how they perform in misspecified settings. Additionally, our current theory also does not address the trajectory estimation itself, which we think is another promising direction for theoretical investigation. On the methodological side, we plan to provide tuning procedures that are less computationally demanding compared to our current grid-search approach and to work on investigating other downstream applications of the eSVD embedding, such as cell clustering, batch correction, imputation, and RNA velocity (La Manno et al., 2018). Also, while this article focuses trajectory analysis based solely on the cells' latent vectors, other downstream applications such as gene clustering and finding marker genes could be developed based on the genes' latent vectors.

## References

- Arora, S., Ge, R., Kannan, R., and Moitra, A. (2016). Computing a nonnegative matrix factorization – Provably. *SIAM Journal on Computing*, 45(4):1582–1611.
- Arora, S., Ge, R., and Moitra, A. (2012). Learning topic models—going beyond SVD. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 1–10. IEEE.
- Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., Levin, K., Lyzinski, V., and Qin, Y. (2017). Statistical inference on random dot product graphs: A survey. *The Journal of Machine Learning Research*, 18(1):8393–8484.
- Balakrishnan, S., Wainwright, M. J., Yu, B., et al. (2017). Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120.
- Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W., Ng, L. G., Ginhoux, F., and Newell, E. W. (2019). Dimensionality reduction for visualizing single-cell data using UMAP. *Nature biotechnology*, 37(1):38.
- Bergen, V., Lange, M., Peidli, S., Wolf, F. A., and Theis, F. J. (2020). Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology*, pages 1–7.
- Blanchard, G., Bousquet, O., and Zwald, L. (2007). Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2-3):259–294.
- Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411–420.
- Cai, T. T., Zhang, A., et al. (2018). Rate-optimal perturbation bounds for singular subspaces with applications to high-dimensional statistics. *The Annals of Statistics*, 46(1):60–89.

- Cai, Z. and Xiao, M. (2016). Oligodendrocytes and Alzheimer’s disease. *International Journal of Neuroscience*, 126(2):97–104.
- Candès, E. J., Sur, P., et al. (2020). The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression. *The Annals of Statistics*, 48(1):27–42.
- Cao, J., Spielmann, M., Qiu, X., Huang, X., Ibrahim, D. M., Hill, A. J., Zhang, F., Mundlos, S., Christiansen, L., Steemers, F. J., et al. (2019). The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502.
- Chi, Y., Lu, Y. M., and Chen, Y. (2019). Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269.
- Collins, M., Dasgupta, S., and Schapire, R. E. (2002). A generalization of principal components analysis to the exponential family. In *Advances in neural information processing systems*, pages 617–624.
- Donoho, D. and Stodden, V. (2004). When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, pages 1141–1148.
- Durif, G., Modolo, L., Mold, J., Lambert-Lacroix, S., and Picard, F. (2017). Probabilistic count matrix factorization for single cell expression data analysis. In *Research in Computational Molecular Biology*, page 254. Springer.
- Efron, B. et al. (1978). The geometry of exponential families. *The Annals of Statistics*, 6(2):362–376.
- Eldridge, J., Belkin, M., and Wang, Y. (2018). Unperturbed: Spectral analysis beyond Davis-Kahan. pages 321–358.
- Fan, J., Wang, W., and Zhong, Y. (2018). An  $\ell_\infty$  eigenvector perturbation bound and its application to robust covariance estimation. *Journal of Machine Learning Research*, 18(207):1–42.
- Feng, C., Wang, H., Han, Y., Xia, Y., and Tu, X. M. (2013). The mean value theorem and Taylor’s expansion in statistics. *The American Statistician*, 67(4):245–248.
- Ge, R., Jin, C., and Zheng, Y. (2017). No spurious local minima in nonconvex low rank problems: A unified geometric analysis. pages 1233–1242.
- Gillis, N. (2017). Introduction to nonnegative matrix factorization. *arXiv preprint arXiv:1703.00663*.
- Gunasekar, S., Ravikumar, P., and Ghosh, J. (2014). Exponential family matrix completion under structural constraints. In *International Conference on Machine Learning*, pages 1917–1925.
- Haghverdi, L., Buettner, F., and Theis, F. J. (2015). Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics*, 31(18):2989–2998.
- Haghverdi, L., Buettner, M., Wolf, F. A., Buettner, F., and Theis, F. J. (2016). Diffusion pseudotime robustly reconstructs lineage branching. *Nature methods*, 13(10):845.
- Hastie, T. and Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, 84(406):502–516.
- Herman, J. S. and Grün, D. (2018). FateID infers cell fate bias in multipotent progenitors from single-cell RNA-seq data. *Nature methods*, 15(5):379.
- Hicks, S. C., Townes, F. W., Teng, M., and Irizarry, R. A. (2017). Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics*.

- Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.
- Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435.
- Jain, P., Netrapalli, P., and Sanghavi, S. (2013). Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM.
- Kessaris, N., Fogarty, M., Iannarelli, P., Grist, M., Wegner, M., and Richardson, W. D. (2006). Competing waves of oligodendrocytes in the forebrain and postnatal elimination of an embryonic lineage. *Nature neuroscience*, 9(2):173.
- Kharchenko, P. V., Silberstein, L., and Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11(7):740.
- Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., Green, A. R., et al. (2017). SC3: Consensus clustering of single-cell RNA-seq data. *Nature methods*, 14(5):483–486.
- Kraemer, G., Reichstein, M., and Mahecha, M. D. (2018). dimRed and coRanking – Unifying dimensionality reduction in R. *R Journal*, 10(1):342–358.
- Kundu, A., Bach, F., and Bhattacharyya, C. (2017). Convex optimization over intersection of simple sets: Improved convergence rate guarantees via an exact penalty approach. *arXiv preprint arXiv:1710.06465*.
- La Manno, G., Soldatov, R., Zeisel, A., Braun, E., Hochgerner, H., Petukhov, V., Lidschreiber, K., Kastrioti, M. E., Lönnerberg, P., Furlan, A., et al. (2018). RNA velocity of single cells. *Nature*, 560(7719):494–498.
- Lafond, J. (2015). Low rank matrix completion with exponential family noise. In *Conference on Learning Theory*, pages 1224–1243.
- Landgraf, A. J. and Lee, Y. (2019). Generalized principal component analysis: Projection of saturated model parameters. *Technometrics*, pages 1–14.
- Ledoux, M. and Talagrand, M. (2013). *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media.
- Lee, A. B. and Wasserman, L. (2010). Spectral connectivity analysis. *Journal of the American Statistical Association*, 105(491):1241–1255.
- Lei, J. (2018). Network representation using graph root distributions. *arXiv preprint arXiv:1802.09684*.
- Li, T., Levina, E., and Zhu, J. (2020). Network cross-validation by edge sampling. *Biometrika*, 107(2):257–276.
- Li, W. V. and Li, J. J. (2018). An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature communications*, 9(1):997.
- Liu, C. and Martin, R. (2020). Inferential models and possibility measures. *arXiv preprint arXiv:2008.06874*.



- Liu, L. T., Dobriban, E., Singer, A., et al. (2018). ePCA: High dimensional exponential family PCA. *The Annals of Applied Statistics*, 12(4):2121–2150.
- Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053.
- Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology*, 15(12):550.
- Lun, A. T., Bach, K., and Marioni, J. C. (2016). Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome biology*, 17(1):75.
- Ma, C., Wang, K., Chi, Y., and Chen, Y. (2018). Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion, and blind deconvolution. *Foundations of Computational Mathematics*, pages 1–182.
- Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.
- Maezika, M. (2016). The singular value decomposition and low rank approximation.
- Marques, S., van Bruggen, D., Vanichkina, D. P., Floriddia, E. M., Munguba, H., Våremo, L., Giacomello, S., Falcão, A. M., Meijer, M., Björklund, Å. K., et al. (2018). Transcriptional convergence of oligodendrocyte lineage progenitors during development. *Developmental cell*, 46(4):504–517.
- Marques, S., Zeisel, A., Codeluppi, S., van Bruggen, D., Falcão, A. M., Xiao, L., Li, H., Häring, M., Hochgerner, H., Romanov, R. A., et al. (2016). Oligodendrocyte heterogeneity in the mouse juvenile and adult central nervous system. *Science*, 352(6291):1326–1329.
- Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322.
- McCarthy, D. J., Campbell, K. R., Lun, A. T., and Wills, Q. F. (2017). Scater: Pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics*, 33(8):1179–1186.
- McInnes, L., Healy, J., and Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.
- Menn, B., Garcia-Verdugo, J. M., Yaschine, C., Gonzalez-Perez, O., Rowitch, D., and Alvarez-Buylla, A. (2006). Origin of oligodendrocytes in the subventricular zone of the adult brain. *Journal of Neuroscience*, 26(30):7907–7918.
- Olshen, A., Seshan, V. E., Lucito, R., and Wigler, M. (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572.
- Pierson, E. and Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome biology*, 16(1):241.
- Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature communications*, 9(1):284.
- Saelens, W., Cannoodt, R., Todorov, H., and Saeys, Y. (2019). A comparison of single-cell trajectory inference methods. *Nature biotechnology*, 37(5):547.
- Street, K., Risso, D., Fletcher, R. B., Das, D., Ngai, J., Yosef, N., Purdom, E., and Dudoit, S. (2018). Slingshot: Cell lineage and pseudotime inference for single-cell transcriptomics. *BMC genomics*, 19(1):477.



- Sun, S., Zhu, J., Ma, Y., and Zhou, X. (2019). Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. *Genome biology*, 20(1):269.
- Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.
- Tibshirani, R. J. (2017). Dykstra’s algorithm, ADMM, and coordinate descent: Connections, insights, and extensions. In *Advances in Neural Information Processing Systems*, pages 517–528.
- Townes, F. W., Hicks, S. C., Aryee, M. J., and Irizarry, R. A. (2017). Varying-censoring aware matrix factorization for single cell RNA-sequencing. *bioRxiv*, page 166736.
- Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381.
- Tropp, J. A. (2012). User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434.
- Udell, M., Horn, C., Zadeh, R., Boyd, S., et al. (2016). Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118.
- van Bruggen, D., Agirre, E., and Castelo-Branco, G. (2017). Single-cell transcriptomic analysis of oligodendrocyte lineage cells. *Current opinion in neurobiology*, 47:168–175.
- Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.
- Vershynin, R. (2012). How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686.
- Wang, J., Huang, M., Torre, E., Dueck, H., Shaffer, S., Murray, J., Raj, A., Li, M., and Zhang, N. R. (2018). Gene expression distribution deconvolution in single-cell RNA sequencing. *Proceedings of the National Academy of Sciences*, 115(28):E6437–E6446.
- Wang, L., Zhang, X., and Gu, Q. (2016). A unified computational and statistical framework for nonconvex low-rank matrix estimation. *arXiv preprint arXiv:1610.05275*.
- Wang, Z., Gu, Q., Ning, Y., and Liu, H. (2015). High dimensional EM algorithm: Statistical optimization and asymptotic normality. pages 2521–2529.
- Witten, D. M. (2011). Classification and clustering of sequencing data using a Poisson model. *The Annals of Applied Statistics*, 5(4):2493–2518.
- Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534.
- Yu, M., Gupta, V., Kolar, M., et al. (2020). Recovery of simultaneous low rank and two-way sparse coefficient matrices, a nonconvex approach. *Electronic Journal of Statistics*, 14(1):413–457.
- Yu, Y., Wang, T., and Samworth, R. J. (2014). A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323.
- Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. *Genome biology*, 18(1):174.

- Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142.
- Zhang, A., Cai, T. T., and Wu, Y. (2018). Heteroskedastic PCA: Algorithm, optimality, and applications. *arXiv preprint arXiv:1810.08316*.
- Zhang, Y., Chen, K., Sloan, S. A., Bennett, M. L., Scholze, A. R., O’Keeffe, S., Phatnani, H. P., Guarnieri, P., Caneda, C., Ruderisch, N., et al. (2014). An RNA-sequencing transcriptome and splicing database of glia, neurons, and vascular cells of the cerebral cortex. *Journal of Neuroscience*, 34(36):11929–11947.
- Zhao, T., Wang, Z., and Liu, H. (2015). Nonconvex low rank matrix factorization via inexact first order oracle. *Advances in Neural Information Processing Systems*.
- Zhou, X. (2018). On the Fenchel duality between strong convexity and Lipschitz continuous gradient. *arXiv preprint arXiv:1803.06573*.
- Zhu, L., Lei, J., Klei, L., Devlin, B., and Roeder, K. (2019). Semisoft clustering of single-cell data. *Proceedings of the National Academy of Sciences*, 116(2):466–471.
- Zwald, L. and Blanchard, G. (2006). On the convergence of eigenspaces in kernel principal component analysis. In *Advances in neural information processing systems*, pages 1649–1656.

## A Code and reproducibility

The code for the method, simulation, and data analysis, as well as the original data used, can be found at <https://github.com/linnykos/esvd>, in the `eSVD`, `simulation`, `main` and `data` folders respectively. The dataset we analyzed was originally collected by Marques et al. (2016), found at the Gene Expression Omnibus with accession number GSE75330 (<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE75330>).

## B Discussion on estimator

### B.1 Initialization method

We first define notation needed to describe the initialization method, inspired by Wang et al. (2016). For a given one-parameter exponential-family distribution, let  $(g')^{-1}(\cdot)$  be the inverse function of the  $g'(\cdot)$ , the derivative of the log-partition function for  $F$ , which is guaranteed to exist by the convexity of  $g(\cdot)$ . Furthermore, for a generic matrix  $\Theta \in \mathbb{R}^{n \times p}$  that is rank  $k$ , let  $\mathcal{L}_n(\Theta)$  be equivalent to the loss function  $\mathcal{L}_n(X, Y)$  defined in (3.3) for any  $X \in \mathbb{R}^{n \times k}$  and  $Y \in \mathbb{R}^{p \times k}$  such that  $\Theta = XY^\top$ . Lastly, define  $\Pi_k(\cdot)$  to be the projection operator (based on alternating between truncating the singular values and thresholding) to project a given matrix onto the set

$$\left\{ \Theta \in \mathbb{R}^{n \times p} : \text{rank}(\Theta) \leq k, \quad \text{and} \quad \theta_{ij} \in \mathbb{R} \quad \forall (i, j) \right\}. \quad (\text{B.1})$$

We initialize our estimate,  $\hat{\Theta}^{(0)}$ , to be  $(g')^{-1}(A)$ , where the function  $(g')^{-1}(\cdot)$  is applied entry-wise. Afterwards, we perform projected gradient descent. That is, for  $t \in \{0, \dots, T' - 1\}$  iterations, for a stepsize  $\gamma > 0$ , we iterate,

$$\hat{\Theta}^{(t+1)} = \Pi_k \left( \hat{\Theta}^{(t)} - \gamma \cdot \nabla \mathcal{L}_n(\hat{\Theta}^{(t)}) \right).$$

Let  $\hat{\Theta}^{(T')} = \hat{\Theta}'$  defined in Subsection 4.1 be the initial estimate of  $\Theta$ .

**Determining  $\gamma$ .** In our implementation, we use project gradient steps where within each iteration,  $\gamma$  is selected within each iteration via binary search to be the largest value such that the objective function  $\mathcal{L}_n(\cdot)$  decreases (akin to backtracking line search)

**Lack of convergence.** Unfortunately, in our experiments, we have found that there are a handful of instances where the initial projection onto the set (B.1) does not converge empirically. While there is a rich body of literature studying the projection onto the intersection of many convex sets (Kundu et al. (2017) and Tibshirani (2017)), the set of all rank- $k$  matrices is nonconvex. In instances of lack of convergence, we terminate the above alternating projection and instead fit a  $k$ -block model to  $g^{-1}(\Theta)$  based applying  $k$ -means to both the first  $k$  left and right singular vectors separately. This procedure is reminiscent of those used to fit stochastic block models described in various papers such as Li et al. (2020). We hope to further investigate this projection issue in future work.

We do not use the non-negative matrix factorization (NMF) of  $g^{-1}(\Theta)$  since the computational complexity of most non-negative matrix factorization methods is of a similar order to the entire eSVD method. See Subsection B.4 for additional discussion of how NMF compares to the eSVD.

### B.2 Additional variant of tuning procedure

The metric described in Subsection 4.2 uses the angle of the leading eigenvector to assess the quality of the model fit. While this is a useful metric when comparing among different exponential-family

distributions  $F$  (and generally applicable across a broad range of tuning procedures), when the specific distribution  $F$  is already decided, using the negative log-likelihood can sometimes be another useful metric to use. Specifically, one can compute the negative log-likelihood on all the entries  $\mathcal{O}$  that were omitted during estimation (i.e., the “testing set”), and then choose the model that yields the smallest negative log-likelihood on average across the  $B$  trials. The advantage of this negative log-likelihood metric over the angle metric described in Subsection 4.2 is that it accounts for the predicted variance among each entry in  $\mathcal{O}$ . However, as mentioned in Subsection 4.2, the disadvantage is that it is hard to compare the negative log-likelihood across different exponential-family distributions  $F$ .

### B.3 Usage for common one-parameter exponential-family distributions

The eSVD can be applied to any one-parameter exponential-family distribution, so here, we derive all the necessary ingredients (calculation of the objective function and gradient with respect to  $X$  and  $Y$ ) for common one-parameter exponential-family distributions. We explain the pipeline to derive how to fit a given one-parameter exponential-family distribution  $F$  into the eSVD framework.

1. **Writing distribution in exponential-family form:** For a given one-parameter exponential distribution  $F$ , write the probability density function (or probability mass function) in the form described in (3.2). That is, determine the functions  $g(\cdot)$  (the log-partition function for  $F$ ),  $\eta(\cdot)$  (the natural parameter function) and  $T(\cdot)$  (the sufficient statistic function) such that

$$p(A_{ij} | \theta_{ij}) = h(A_{ij}) \exp(T(A_{ij})^\top \eta(\theta_{ij}) - g(\theta_{ij})).$$

2. **Determine the domain:** Next, based on the log-partition function  $g(\cdot)$ , determine its domain  $\mathcal{R}$ .
3. **Determining the objective function:** Next, with the functions  $g(\cdot)$ ,  $\eta(\cdot)$  and  $T(\cdot)$  explicitly derived, plug them into the objective function,

$$\mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{(i,j)} \left[ g(X_i^\top Y_j) - T(A_{ij})^\top \eta(X_i^\top Y_j) \right].$$

4. **Calculating the gradients:** Lastly, derive the gradient of the above objective function with respect to  $X$  and  $Y$ .

Using the above pipeline, we provide the calculations for the following five distributions as an example for readers. We only write down the gradients with respect to  $X$ , as the gradients with respect to  $Y$  are similar.

- **Gaussian:** For a Gaussian with known variance  $\sigma^2$ , the density is

$$p(A_{ij} | \theta_{ij}) = \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-A_{ij}^2}{2\sigma^2}\right) \right] \cdot \exp\left([A_{ij}/\sigma^2]^\top [\theta_{ij}] - \frac{\theta_{ij}^2}{2\sigma^2}\right),$$

where  $\theta_{ij} = \mu_{ij}$ , and the domain of  $\theta_{ij}$  is  $\mathcal{R} = \mathbb{R}$ . Hence, we can complete the square to derive the objective function, yielding

$$\mathcal{L}_n(X, Y) \propto \frac{1}{np} \sum_{(i,j)} \left[ \left( \frac{(X_i^\top Y_j)}{\sqrt{2\sigma}} - \frac{A_{ij}}{\sqrt{2\sigma}} \right)^2 \right] \propto \frac{1}{np} \sum_{(i,j)} \left[ \left( (X_i^\top Y_j) - A_{ij} \right)^2 \right].$$

Then the gradient with respect to  $X$  is a  $n \times k$  matrix where the  $i$ th row is

$$\nabla_{X_i} \mathcal{L}_n(X, Y) \propto \frac{1}{np} \sum_{j=1}^p \left[ 2 \left( (X_i^\top Y_j) - A_{ij} \right) \cdot Y_j \right].$$

- **Curved Gaussian:** For a curved Gaussian with parameter  $\tau$  (i.e.,  $N(\mu_{ij}, \mu_{ij}^2/\tau^2)$ ), we derive after some algebraic manipulations that the density is,

$$p(A_{ij} | \mu_{ij}) = \left[ \frac{\tau \exp(-\tau^2/2)}{\sqrt{2\pi}} \right] \cdot \exp \left( \frac{-\tau^2 A_{ij}^2}{2} \cdot \frac{1}{\mu_{ij}} + \tau^2 A_{ij} \cdot \frac{1}{\mu_{ij}} + \log(1/\mu_{ij}) \right)$$

Hence, replacing  $\theta_{ij} = 1/\mu_{ij}$  (meaning  $\mathbb{E}[A_{ij}] = 1/\theta_{ij}$ ), and the domain of  $\theta_{ij}$  is  $\mathcal{R} = (0, \infty)$ . Hence, the objective function becomes

$$\mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{(i,j)} \left[ -\log(X_i^\top Y_j) - \begin{bmatrix} \tau^2 A_{ij} \\ -\tau^2 A_{ij}^2/2 \end{bmatrix}^\top \begin{bmatrix} X_i^\top Y_j \\ (X_i^\top Y_j)^2 \end{bmatrix} \right].$$

Then the gradient with respect to  $X$  is a  $n \times k$  matrix where the  $i$ th row is

$$\nabla_{X_i} \mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{j=1}^p \left[ \left( -\frac{1}{X_i^\top Y_j} - \tau^2 A_{ij} + \tau^2 A_{ij}^2 (X_i^\top Y_j) \right) \cdot Y_j \right].$$

- **Exponential:** The density is

$$p(A_{ij} | \theta_{ij}) = [1] \cdot \exp \left( [A_{ij}]^\top [\theta_{ij}] - (-\log(-\theta_{ij})) \right),$$

where  $\theta_{ij} = -\lambda_{ij}$  (meaning  $\mathbb{E}[A_{ij}] = -1/\theta_{ij}$ ), and the domain of  $\theta_{ij}$  is  $\mathcal{R} = (-\infty, 0)$ . Hence, the objective function becomes

$$\mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{(i,j)} \left[ -\log(-X_i^\top Y_j) - [A_{ij}]^\top [X_i^\top Y_j] \right].$$

Then the gradient with respect to  $X$  is a  $n \times k$  matrix where the  $i$ th row is

$$\nabla_{X_i} \mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{j=1}^p \left[ \left( \frac{-1}{X_i^\top Y_j} - A_{ij} \right) \cdot Y_j \right]$$

- **Poisson:** The density is

$$p(A_{ij} | \theta_{ij}) = \left[ \frac{1}{A_{ij}!} \right] \cdot \exp \left( [A_{ij}]^\top [\theta_{ij}] - \exp(\theta_{ij}) \right),$$

where  $\theta_{ij} = \log(\lambda_{ij})$  (meaning  $\mathbb{E}[A_{ij}] = \exp(\theta_{ij})$ ), and the domain of  $\theta_{ij}$  is  $\mathcal{R} = \mathbb{R}$ . Hence, the objective function becomes

$$\mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{(i,j)} \left[ \exp(X_i^\top Y_j) - [A_{ij}]^\top [X_i^\top Y_j] \right].$$

Then the gradient with respect to  $X$  is a  $n \times k$  matrix where the  $i$ th row is

$$\nabla_{X_i} \mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{j=1}^p \left[ \left( \exp(X_i^\top Y_j) - A_{ij} \right) \cdot Y_j \right]$$

- **Negative Binomial:** The negative binomial represents the number of successes before a specified number of failures  $r$  occurs. (This is different from the binomial distribution which represents the number of success among a fixed number of trials.) This parameter  $r$  is often also called the dispersion parameter. For a fixed number of failures  $r$ , the density of Negative Binomial( $r, p_{ij}$ ) is

$$p(A_{ij} | \theta_{ij}) = \left[ \binom{A_{ij} + r - 1}{A_{ij}} \right] \cdot \exp \left( [A_{ij}]^\top [\theta_{ij}] - (-r \log(1 - \exp(\theta_{ij}))) \right),$$

where  $\theta_{ij} = \log(p_{ij})$  (meaning  $\mathbb{E}[A_{ij}] = r \exp(\theta_{ij}) / (1 - \exp(\theta_{ij}))$ ), and the domain of  $\theta_{ij}$  is  $\mathcal{R} = (-\infty, 0)$ . Hence, the objective function becomes

$$\mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{(i,j)} \left[ (-r \log(1 - \exp(X_i^\top Y_j))) - [A_{ij}]^\top [X_i^\top Y_j] \right].$$

Then the gradient with respect to  $X$  is a  $n \times k$  matrix where the  $i$ th row is

$$\nabla_{X_i} \mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{j=1}^p \left[ \left( \frac{r \exp(X_i^\top Y_j)}{1 - \exp(X_i^\top Y_j)} - A_{ij} \right) \cdot Y_j \right]$$

## B.4 Additional comparison of eSVD to estimators in the literature

Continuing Section 3, we discuss additional nuances for the eSVD that makes it different from other methods.

**Comparison to alternating gradient descent.** The majority of theoretical work that investigate nonconvex estimators to perform matrix factorization use alternating projected gradient descent to refine the initial estimate where each iteration updates the current estimates with a gradient step (Wang et al. (2016), Yu et al. (2020), and Chi et al. (2019)). This is in contrast with our choice of using alternating constrained minimization in the eSVD. While we have found alternating projected gradient descent is more amendable for theoretical analysis, in practice, we have found it more numerically unstable due to its sensitivity to the chosen step-sizes.

**Comparison to non-negative matrix factorization.** There is an extremely rich literature on non-negative matrix factorization (NMF, Donoho and Stodden (2004), Arora et al. (2016), Gillis (2017)), and upon first glance, the eSVD seems similar to NMF. However, there are two importance distinctions. First, the key difference between the eSVD and NMF can be seen in Assumption 3.1. Specifically, the eSVD only assumes that the inner products between  $X$  and  $Y$  lie in  $\mathcal{R}$ , such as the positive half-line. This is in contrast to NMF, where either each entry in  $X$  or  $Y$  (or both) lies on the positive half-line. Intuitively, this means the eSVD's task is easier than NMF's task (loosely speaking), since ensuring that each entry in  $X$  and  $Y$  lying on the positive half-line implies that the inner product between each pair of rows in  $X$  and  $Y$  lies on the positive half-line. Second, the eSVD's identifiability assumptions are more tractable than NMF's identifiability assumptions. The eSVD's identifiability assumptions are outlined in Proposition 5.1, which shows that there exists a linear transformation such that the population second moment matrix for  $X$  and  $Y$  are equal. NMF's identifiability conditions are much more nuanced, and leads to concepts such as simplicial cones, separability and anchor words as discussed in Donoho and Stodden (2004) and Arora et al. (2012).



**Comparison to embeddings based on symmetric matrices.** There are many embeddings in the literature that are based on symmetric matrices such as covariance matrices, distance matrices or adjacency matrices of neighborhood graphs. These are in contrast with the eSVD as well as other embeddings based on the random dot product model such as ZINB-WaVE (Risso et al., 2018), pCMF (Durif et al., 2017) and NMF. We discuss the broad merits of these two categories of embeddings, and contrast the specific embedding methods with eSVD below. Embeddings based on symmetric matrices can typically be mathematically analyzed in more detail when compared to those for embeddings based on the random dot product models since the former methods typically have more tractable identifiability conditions and often rely simply on the eigen-decompositions, which are mathematically well understood. However, we feel embeddings based on the random dot product models are generally more appealing for single-cell applications since these methods estimate latent vectors for each cell and each gene jointly and can often incorporate more modeling aspects that reflect our understanding of single-cell data. For example, these latter methods can inherently accommodate downstream tasks like bi-clustering and marker gene detection.

- **Comparison to low-rank covariance matrix estimation for exponential families.** Liu et al. (2018) and Zhang et al. (2018) estimate a low-rank covariance matrix where each entry in the observed matrix  $A$  is drawn from an exponential-family distribution. While this task is non-trivial due to the possible dependency between the mean and covariance (which prevents naively centering the variables around 0), this is a different task than our goals posited in this paper since eSVD estimates a low-rank matrix of natural parameters.
- **Comparison to other embedding methods.** Methods such as kernel PCA, Isomap (Tenenbaum et al., 2000), t-SNE (Maaten and Hinton, 2008), diffusion maps (Haghverdi et al., 2015), and UMAP (McInnes et al., 2018) also perform non-linear embeddings. We lump these methods together as they generally involve constructing an  $n \times n$  symmetric matrix that captures which cells are “neighbors” to which cell or how similar a cell is to another cell given some appropriate notion of similarity. See papers like Lee and Wasserman (2010), Kraemer et al. (2018) and Sun et al. (2019) and the references within for more details. While some of these methods also have rigorous statistical theory (Zwald and Blanchard (2006) and Blanchard et al. (2007)), we have found that in practice, it is hard to assess the quality of fit for these embeddings, and hence also hard to tune the possible parameters of these embeddings effectively. Nonetheless, since these embeddings are often quite successful in practice for complex data, we are interested finding ways to incorporating ideas of the eSVD to these method for future work.

## C Formal description of analysis pipeline

### C.1 Main analysis pipeline

The following procedure describes how we analyze the data to obtain the eSVD-related results in Section 7.

1. **Screening genes:** The dataset starts with 23,556 genes. However, since not all the genes are informative for our analysis, we use the following two methods to select genes, based loosely on Zhu et al. (2019).
  - **Sparse principal component analysis** (Witten et al., 2009): This method uses a tuning parameter that controls the  $\ell_1$ -norm of the eigenvectors, of which we try 10 values spaced exponentially between 0 and  $\log(23,556)$ . We choose the model such that the first  $K = 5$  sparse eigenvectors involve more than 500 genes and would capture more than 90% of the variance.

- **DESCEND** (Wang et al., 2018): Based on the statistical model developed in Wang et al. (2018) that fits a cubic spline regressing the Gini index onto the log of the estimated mean expression of each gene, we find highly variable genes that deviate from the mean by 50 times the normalized difference.

Together, this results in 983 unique genes being selected. In the following parts of the analysis, we will denote the preprocessed data as  $A \in \mathbb{R}_+^{n \times p}$  where there are  $n = 5069$  cells and  $p = 983$  genes.

2. **Rescaling:** We normalize each cell by its library size (i.e., divide each row by its sum) and then multiply the entire matrix by a scalar such that the maximum value is 1000. This is to prevent small numbers from underflowing our method later on. Note that similar to works like Durif et al. (2017) and Risso et al. (2018), we choose to not  $\log_2$ -transform our library-normalized matrix  $A$ .
3. **Tuning the dimensionality of the embedding  $k$  and the nuisance parameter of the curved Gaussian distribution  $\tau$ :** Omitting four entries per row and per column of  $A$  for each trial of  $b$  over a total of  $B = 3$  trials, we use the matrix-completion procedure outlined in Subsection 4.2 to select the number of latent dimensions  $k \in \{3, 5, 10, 20\}$  and nuisance parameter controlling the relation between the mean and variance  $\tau \in \{0.5, 1, 2, 4\}$  (for a total of 16 different parameter settings). We end up selecting  $k = 5$  and  $\tau = 2$ . Note that it is important to refit across the different values of  $k$  since non-linear embeddings are not typically nested, as discussed in Durif et al. (2017).
4. **Embedding via eSVD:** We apply the eSVD where  $F$  is the the curved Gaussian distribution with  $k = 5$  and  $\tau = 2$  to minimize the loss function (3.3), as prescribed in Subsection 4.1. The initialization procedure is described in Appendix B.1.
5. **Estimating the cell developmental trajectories and uncertainty tube:** Using the thirteen cell sub-types from Marques et al. (2016) as the cluster labels, we apply our modified Slingshot and construct the uncertainty tubes, as alluded to in Section 7 and detailed in Appendix G.

**Screening before rescaling.** In contrast to our order of Steps 1 and 2 above, most single-cell preprocessing pipelines, such as ones recommended by Seurat (Butler et al., 2018), often first rescale the counts for each cell and then screen the genes. However, we found preprocessing the data in this fashion hindered the effectiveness of the downstream trajectory analysis based on the eSVD embedding, although the results were qualitatively the same. This is due to the nature of our dataset, where the library size for older cells are substantially higher than younger cells, as seen in Figure 9. By screening genes before rescaling the counts, we select more genes that more informative in distinguishing among the six mature oligodendrocyte cell sub-types. This is critical for our application, as our goal is to distinguish the lineages among the different mature oligodendrocytes.

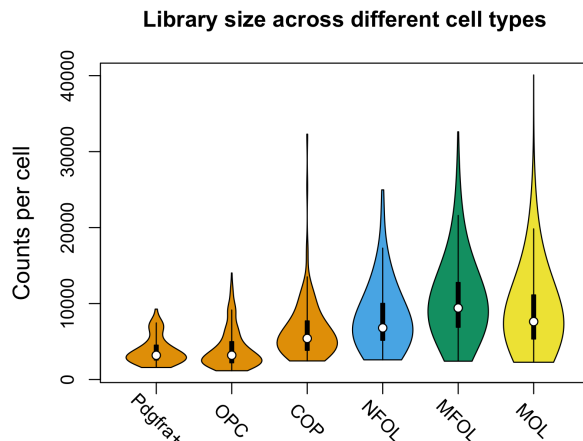


Figure 9: Library size of the raw data across all 5069 cells, split across the six major cell types displayed in Figure 1.

## C.2 Additional details of analysis in Section 2 and SVD-related results in Section 7

We describe additional details of the analysis used to produce the various figures in Section 2, as well as the SVD-related results in Subsection 7.2. The pipeline changes slightly from Appendix C.1 when performing the SVD analysis in the following way:

1. **Different transformation:** As opposed to Step 2’s rescaling, we instead normalize each cell by its library size, and then transform each entry according to  $x \rightarrow \log_2(10,000 \cdot x + 1)$ , as suggested by work such as Seurat (Butler et al., 2018). We call this the  $\log_2$ -transformation in the main text and elsewhere in this supplement.
2. **Different tuning selection:** As opposed to Step 3’s tuning via the eSVD, we instead tune via SoftImpute, a common matrix completion algorithm designed for the constant-variance Gaussian family. This is detailed below under “Details for Figure 2.”
3. **Different embedding:** As opposed to Step 4’s embedding via the eSVD, we use the embedding described in the main text at (2.1) and (2.2), which is the embedding the SVD.

We note the results in Section 2 and Subsection 7.2 are qualitatively the same if we did not use the  $\log_2$ -transformation and only use the SVD on  $A$  directly.

**Details for Figure 1.** The scatter plot is produced by the SVD embedding described in (2.1) and (2.2), where the coloring of the points is based on the cell-type information provided in Marques et al. (2016). The contour of the densities estimated based on the `MASS::kde2d` function (using the default bandwidth), where the level of the contour is chosen to be the 92.5% quantile of the estimated density across a grid of points in this two-dimensional space. This quantile level is chosen solely based on the suitability of the figure, and provides the reader a sense of the density of the points that is otherwise hard to gauge based on only the scatter plot.

**Details for Figure 2.** SoftImpute (Mazumder et al., 2010) requires the dimensionality of the latent space  $k$  and a tuning parameter  $\lambda$  to determine the amount of the spectral regularization. To choose this, we try 50 different value of  $\lambda$  from 1 to the value given in `softImpute::lambda0` (a function that computes the smallest value of  $\lambda$  that yields the all-0 estimated matrix), as well

as  $k \in \{5, 10, 20, 50\}$  for a total of 200 different parameter settings. We then choose  $\lambda$  and  $k$  based on the matrix-completion procedure outlined in Subsection 4.2, where  $F$  is set to be the Gaussian distribution with constant variance. Importantly, this results in choosing  $k = 50$ , which is used when fitting the SVD embedding used for downstream analysis in Subsection 7.2. (Note that this is different from  $k = 5$  used for the eSVD embedding shown in Subsection 7.3.)

**Details for Figure 3.** For this plot specifically, let  $A$  denote the preprocessed single-cell dataset *prior* to the  $\log_2$ -transformation. Figure 3A is created by computing the logarithm of the column-wise (i.e., gene-wise) mean and standard deviation of  $A$ , the preprocessed single-cell dataset. The color of the point is based on the ANOVA  $p$ -value that tests if values in a column of  $A$  (i.e., the expression of each gene) is equal in mean across all six major cell types shown in Figure 1, with blacker points denoting a  $p$ -value closer to 0 and more yellow points denoting  $p$ -values closer to 1.

Figure 3B is created by first computing the first principal component of  $A$  (i.e., the leading eigenvector of the empirical covariance matrix of  $A$ ), setting all the negative entries to 0, and then renormalizing all the entries of the resulting vector to sum to 1. We set all the negative values to 0 so the resulting vector can meaningfully represent a weighted average. We then compute the inner product between the resulting vector and each row of  $A$ , and then create a violin plot based on the grouping the resulting inner product by the six major cell types shown in in Figure 1.

## D Statistical theory for estimation of matrix of natural parameters

As mentioned in Section 5, an important aspect of interpreting the convergence rate showed in Proposition 5.2 is knowing under what assumptions can we control the estimation error  $\|\Theta - \hat{\Theta}\|_F$ . This is the goal of this section, where we operate under the correctly-specified model setting, meaning the eSVD uses the log-likelihood of  $F$ , the true data-generating distribution. Throughout this section, conditioned on  $X$  and  $Y$  (the matrix of random latent vectors for each cell and gene that we are trying to estimate), let the SVD of  $\Theta = XY^\top$  be denoted as

$$\Theta = \tilde{U}\tilde{D}\tilde{V}^\top, \quad (\text{D.1})$$

where we denote the singular values as  $\tilde{d}_1 \geq \dots \geq \tilde{d}_k \geq 0$ . Also, whenever we put a bar over a matrix, as in  $\bar{X}$  and  $\bar{Y}$ , we are explicitly requiring such matrices to have spectral norm  $\sqrt{n}$  and  $\sqrt{p}$  respectively.

The following assumptions ensure that the exponential-family distribution  $F$  is well-behaved. We define notation to distinguish between sample and population optimizers, borrowing terminology from Balakrishnan et al. (2017). Let  $M_n^X(\bar{Y}^{(t)})$  and  $M_n^Y(\bar{X}^{(t+1)})$  denote  $X^{(t+1)}$  and  $Y^{(t+1)}$  respectively in (4.1) and (4.3). In the nonconvex literature, these denote the *sample minimization operators*. Likewise, we define the population loss function,

$$\mathcal{L}(X, Y) = \frac{1}{np} \sum_{(i,j)} \left[ g(X_i^\top Y_j) - \mathbb{E}[T(A_{ij})]^\top \eta(X_i^\top Y_j) \right], \quad (\text{D.2})$$

and the corresponding *population minimization operators*,

$$M^X(\bar{Y}) = \operatorname{argmin}_{X \in \mathbb{R}^{n \times k}} \mathcal{L}(X, \bar{Y}) : X_i^\top \bar{Y}_j \in \mathcal{R}, \quad \forall (i, j), \quad (\text{D.3})$$

$$M^Y(\bar{X}) = \operatorname{argmin}_{Y \in \mathbb{R}^{p \times k}} \mathcal{L}(\bar{X}, Y) : \bar{X}_i^\top Y_j \in \mathcal{R}, \quad \forall (i, j). \quad (\text{D.4})$$

To handle identifiability issues, let us define the set of matrix pairs,

$$\{\bar{\mathcal{X}}^*, \mathcal{Y}^*\} = \left\{ \{\bar{X}, Y\} : \Theta = \bar{X}Y^\top \right\}.$$

By the SVD of  $\Theta$  in (D.1), we can see that the set  $\bar{\mathcal{X}}^*$  represents all matrices that are equal to  $\sqrt{n} \cdot \tilde{U}$  up to rotation. Similarly, we can define the pair of spaces  $\{\mathcal{X}^*, \bar{\mathcal{Y}}^*\}$ . Observe that by the above definitions,

$$M^X(\bar{Y}) = X, \quad \text{and} \quad M^Y(\bar{X}) = Y, \quad (\text{D.5})$$

for pairs of matrices  $\{\bar{X}, Y\} \in \{\bar{\mathcal{X}}^*, \mathcal{Y}^*\}$  and  $\{X, \bar{Y}\} \in \{\mathcal{X}^*, \bar{\mathcal{Y}}^*\}$ .

We now lay down the key assumptions needed to prove the estimation of  $\Theta$ . While we call these “assumptions,” Assumptions D.1-D.4 should be thought of more as general conditions restricting what kind of random dot product model is allowable. The technical requirements relating all the four assumptions below are stated in Proposition D.1 to come.

**Assumption D.1** (Strong convexity and gradient Lipschitz). *Assume that the population loss function  $\mathcal{L}(\cdot, \cdot)$  is  $\mu$ -strongly convex and its gradient is  $L$ -Lipschitz for  $L \geq \mu > 0$  after fixing one of the input matrices. Specifically, for any matrices  $X, X' \in \mathbb{R}^{n \times k}$  and  $Y \in \mathbb{R}^{p \times k}$  satisfying Assumption 3.1,*

$$\frac{\mu}{2} \|X - X'\|_F^2 \leq \mathcal{L}(X', Y) - \mathcal{L}(X, Y) - \left\langle X' - X, \nabla_X \mathcal{L}(X, Y) \right\rangle \leq \frac{L}{2} \|X - X'\|_F^2,$$

and a similar assumption holds for any matrices  $Y, Y' \in \mathbb{R}^{p \times k}$  and  $X \in \mathbb{R}^{n \times k}$ .

For the below assumptions, let

$$\mathbb{B}_F(\omega; \bar{\mathcal{Y}}^*) = \left\{ Y' \in \mathbb{R}^{p \times k} : \text{there exists } \bar{Y}^* \in \bar{\mathcal{Y}}^* \text{ such that } \|Y' - \bar{Y}^*\|_F \leq \omega \right\},$$

in other words, the union of Frobenius balls of matrices around any matrix in  $\bar{\mathcal{Y}}^*$  with radius some determined radius  $\omega$ . We invoke a similar definition for  $\mathbb{B}_F(\omega; \bar{\mathcal{X}}^*)$ .

**Assumption D.2** (Gradient Lipschitz with respect to alternating variable). *Assume that there exists a  $S > 0$  such that for any pairs  $\{X^*, \bar{Y}^*\} \in \{\mathcal{X}^*, \bar{\mathcal{Y}}^*\}$  and  $\{\bar{X}^*, Y^*\} \in \{\bar{\mathcal{X}}^*, \mathcal{Y}^*\}$  and any rescaled orthonormal matrices  $\bar{Y} \in \mathbb{B}_F(\omega; \bar{\mathcal{Y}}^*)$  and  $\bar{X} \in \mathbb{B}_F(\omega; \bar{\mathcal{X}}^*)$  with spectral norm  $\sqrt{p}$  and  $\sqrt{n}$  respectively,*

$$\begin{aligned} \|\nabla_X \mathcal{L}(X^*, \bar{Y}) - \nabla_X \mathcal{L}(X^*, \bar{Y}^*)\|_F &\leq S \|\bar{Y} - \bar{Y}^*\|_F, \\ \|\nabla_Y \mathcal{L}(\bar{X}, Y^*) - \nabla_Y \mathcal{L}(\bar{X}^*, Y^*)\|_F &\leq S \|\bar{X} - \bar{X}^*\|_F. \end{aligned}$$

**Assumption D.3** (Uniform statistical error). *Conditioned on  $X$  and  $Y$ , assume that with probability at least  $1 - c/\min\{n, p\}$  for some universal constant  $c$  that*

$$\begin{aligned} \sup_{Y' \in \mathbb{B}_F(\omega; \bar{\mathcal{Y}}^*)} \|M^X(Y') - M_n^X(Y')\|_F &\leq \varepsilon_{unif}(n, p), \\ \sup_{X' \in \mathbb{B}_F(\omega; \bar{\mathcal{X}}^*)} \|M^Y(X') - M_n^Y(X')\|_F &\leq \varepsilon_{unif}(n, p), \end{aligned}$$

where  $\varepsilon_{unif}(n, p)$  is some function of  $n$  and  $p$  (and possibly other quantities).

In Assumptions D.1 and D.2, the strongly convexity and smoothness enable fast convergence. These assumptions are common in work that study matrix factorization such as Wang et al. (2016) and Yu et al. (2020). Likewise, Assumption D.3 ensures the sample optimizer does not deviate too far from the population optimizer, commonly used in work such as Balakrishnan et al. (2017).

**Assumption D.4** (Initialization condition). *Conditioned on  $X$  and  $Y$ , assume that for some pair of matrices  $\bar{Y}^* \in \bar{\mathcal{Y}}^*$ , the initial estimate  $\bar{Y}^{(0)}$  satisfies*

$$\|\bar{Y}^* - \bar{Y}^{(0)}\|_F \leq \omega.$$

Assumption D.4 is an initialization condition similar to Wang et al. (2016) and Yu et al. (2020) that ensures the alternating minimization steps can allow  $\hat{\Theta}^{(T)}$  to converge towards  $\Theta$ . This is described in the following proposition.

**Proposition D.1.** *Conditioned on  $X$  and  $Y$ , for an appropriate choice of  $\omega$  under Assumptions 3.1 and D.1-D.4 such that*

$$\frac{8(np)^{1/2}}{\tilde{d}_k} < 1, \quad \text{and} \quad S/\mu < 1 \quad \text{and} \quad \varepsilon_{\text{unif}}(n, p) \leq (1 - S/\mu)\omega,$$

*then for the number of iterations  $T$  large enough, with probability at least  $1 - 2c/\min\{n, p\}$ , eSVD described in (4.1)-(4.4) achieves the rate*

$$\|\Theta - \hat{\Theta}^{(T)}\|_F = C \cdot \left( \frac{\max\{n^{1/2}, \tilde{d}_1/n^{1/2}\} \cdot (np)^{1/2}}{\tilde{d}_k - 8(np)^{1/2}S/\mu} \cdot \varepsilon_{\text{unif}}(n, p) \right),$$

*for some universal constants  $c$  and  $C$ .*

In short, the above proposition shows that  $\varepsilon_{\text{unif}}(n, p)$  is the term that is mainly responsible for driving the rate. To apply Proposition D.1 to a particular model, one needs to compute the terms  $\mu$ ,  $L$ ,  $S$ , and  $\varepsilon_{\text{unif}}(n, p)$  for an  $\omega$  that satisfies the conditions in Proposition D.1. This often requires positing assumptions in the context of the particular model that imply Assumptions D.1-D.3. We demonstrate this in the next section with minor modifications to both the estimator and the assumptions.

## E Application of propositions to the curved Gaussian model

In this section, we detail the assumptions needed to apply Proposition D.1 to the curved Gaussian model (4.7). This entails introducing assumptions to help determine the values of  $\mu$ ,  $L$ , and  $S$  used in Assumptions D.1 and D.2. We introduce some notation beforehand. First, observe that we can rescale the normalization constants in (4.1) and (4.3). Hence, we redefine the sample loss functions to be

$$\mathcal{L}_n^{(X)}(X, Y) = \frac{1}{p} \sum_{(i,j)} \left[ -\log(X_i^\top Y_j) - \left[ \begin{array}{c} \tau^2 A_{ij} \\ -\tau^2 A_{ij}^2/2 \end{array} \right]^\top \left[ \begin{array}{c} X_i^\top Y_j \\ (X_i^\top Y_j)^2 \end{array} \right] \right], \quad (\text{E.1})$$

$$\mathcal{L}_n^{(Y)}(X, Y) = \frac{1}{n} \sum_{(i,j)} \left[ -\log(X_i^\top Y_j) - \left[ \begin{array}{c} \tau^2 A_{ij} \\ -\tau^2 A_{ij}^2/2 \end{array} \right]^\top \left[ \begin{array}{c} X_i^\top Y_j \\ (X_i^\top Y_j)^2 \end{array} \right] \right], \quad (\text{E.2})$$

Their corresponding population loss functions that we analyze in this section are

$$\mathcal{L}^{(X)}(X, Y) = \frac{1}{p} \sum_{(i,j)} \left[ -\log(X_i^\top Y_j) - \left[ \begin{array}{c} \tau^2 \mathbb{E}[A_{ij}] \\ -\tau^2 \mathbb{E}[A_{ij}^2]/2 \end{array} \right]^\top \left[ \begin{array}{c} X_i^\top Y_j \\ (X_i^\top Y_j)^2 \end{array} \right] \right], \quad (\text{E.3})$$

$$\mathcal{L}^{(Y)}(X, Y) = \frac{1}{n} \sum_{(i,j)} \left[ -\log(X_i^\top Y_j) - \left[ \begin{array}{c} \tau^2 \mathbb{E}[A_{ij}] \\ -\tau^2 \mathbb{E}[A_{ij}^2]/2 \end{array} \right]^\top \left[ \begin{array}{c} X_i^\top Y_j \\ (X_i^\top Y_j)^2 \end{array} \right] \right], \quad (\text{E.4})$$



where  $A_{ij}$  follows the distribution (4.7), and the above expectations are understood to be conditioned on  $\Theta$ . The change from  $1/(np)$  in (4.1) and (4.3) to  $1/p$  and  $1/n$  is for simplicity and facilitates to control the spectrum of the Hessian appropriately. We define the minimization operators we will use in this section as

$$M_n^X(\bar{Y}) = \operatorname{argmin}_{X \in \mathbb{R}^{n \times k}} \mathcal{L}_n^{(X)}(X, \bar{Y}) : X_i^\top \bar{Y}_j \in \mathcal{R}, \quad \forall(i, j)$$

$$M_n^Y(\bar{X}) = \operatorname{argmin}_{Y \in \mathbb{R}^{p \times k}} \mathcal{L}_n^{(Y)}(\bar{X}, Y) : (\bar{X}_i)^\top Y_j \in \mathcal{R}, \quad \forall(i, j)$$

$$M^X(\bar{Y}) = \operatorname{argmin}_{X \in \mathbb{R}^{n \times k}} \mathcal{L}^{(X)}(X, \bar{Y}) : X_i^\top \bar{Y}_j \in \mathcal{R}, \quad \forall(i, j)$$

$$M^Y(\bar{X}) = \operatorname{argmin}_{Y \in \mathbb{R}^{p \times k}} \mathcal{L}^{(Y)}(\bar{X}, Y) : (\bar{X}_i)^\top Y_j \in \mathcal{R}, \quad \forall(i, j).$$

While we state results for both  $\mathcal{L}^{(X)}(\cdot, \cdot)$  and  $\mathcal{L}^{(Y)}(\cdot, \cdot)$  in this section, we prove only statements for  $\mathcal{L}^{(X)}(\cdot, \cdot)$  since the proofs for both loss functions are identical.

**Assumptions.** The following three assumptions are needed to analyze the curved Gaussian setting.

**Assumption E.1** (Refinement of domain). *Assume for the curved Gaussian distribution (4.7), let  $\mathcal{R} = [1/r, r]$  defined in Assumption 3.1, where there exists a fixed constant  $r > 1$ .*

**Assumption E.2** (Inner product error). *For a universal constant  $c$ , conditioned on  $X$  and  $Y$ , assume there exists  $\{X^*, \bar{Y}^*\} \in \{\mathcal{X}^*, \bar{\mathcal{Y}}^*\}$  such that the initialization  $\bar{Y}^{(0)}$  satisfies for all  $(i, j) \in \{1, \dots, n\} \times \{1, \dots, p\}$*

$$|(X_i^*)^\top (\bar{Y}_j^{(0)} - \bar{Y}_j^*)| \leq c \cdot \frac{1}{\log(\min\{n, p\})} |(X_i^*)^\top \bar{Y}_j^*|. \quad (\text{E.5})$$

*In addition, conditioned on  $X$  and  $Y$ , assume for each iterations  $t \in \{1, \dots, T\}$  throughout the algorithm, there exists  $\{X^*, \bar{Y}^*\} \in \{\mathcal{X}^*, \bar{\mathcal{Y}}^*\}$  and  $\{\bar{X}^*, Y^*\} \in \{\bar{\mathcal{X}}^*, \mathcal{Y}^*\}$  where  $\Theta = X^*(\bar{Y}^*)^\top = \bar{X}^*(Y^*)^\top$  such that for all  $(i, j) \in \{1, \dots, n\} \times \{1, \dots, p\}$*

$$|(X_i^*)^\top (\bar{Y}_j^{(t)} - \bar{Y}_j^*)| \leq c \cdot \frac{1}{\log(\min\{n, p\})} |(X_i^*)^\top \bar{Y}_j^*|, \quad (\text{E.6})$$

$$|(\bar{X}_i^{(t)} - \bar{X}_i^*)^\top Y_j^*| \leq c \cdot \frac{1}{\log(\min\{n, p\})} |(\bar{X}_i^*)^\top Y_j^*|. \quad (\text{E.7})$$

Assumption E.1 effectively ensures  $\mathcal{R}$  is bounded away from 0, a condition to ensure the Hessian is well-controlled. Assumption E.2 imposes an additional requirement on the original initialization condition in Assumption D.4, and is similar to the incoherence assumption in Ma et al. (2018) and Chi et al. (2019). There, the authors prove that spectral initialization ensures the requirement analogous to (E.5) is met with high probability, and each iteration retains properties analogous to (E.6) and (E.7).

**Assumption E.3** (Fixed statistical error). *Conditioned on  $X$  and  $Y$ , for any matrices  $X' \in \mathbb{R}^{n \times k}$  or  $Y' \in \mathbb{R}^{p \times k}$ , with probability at least  $1 - c/\min\{n, p\}$  for some universal constant  $c$  that*

$$\max \left\{ \|M^X(Y') - M_n^X(Y')\|_F, \|M^Y(X') - M_n^Y(X')\|_F \right\} \leq \varepsilon_{\text{fixed}}(n, p),$$

where  $\varepsilon_{\text{fixed}}(n, p)$  is some function of  $n$  and  $p$  (and possibly other quantities).

As mentioned in Appendix D, Assumption E.3 is different from Assumption D.3 but enables a simpler analysis. The reason we impose this new assumption is that minimizing objective functions such as (E.1) does not have a closed-form solution, so uniformly bounding the error is difficult for the curved Gaussian model. This simplifying assumption has been used in other work such as Wang et al. (2015). By imposing Assumption E.3 instead of Assumption D.3, eSVD now requires resampling, i.e., a fresh batch of samples every iteration, for all  $T$  iterations. While this yields a different algorithm that is not practical to use, we believe the theoretical properties we prove also roughly hold for the curved Gaussian model without resampling. We derive the appropriate magnitude of  $\varepsilon_{\text{fixed}}(n, p)$  below in Lemma E.3.

**Controlling  $\mu$ ,  $L$ ,  $S$  and  $\varepsilon_{\text{fixed}}(n, p)$ .** The following lemma implies that  $\mathcal{L}^{(X)}(\cdot, \cdot)$  is  $\mu = (2 + \tau^2)/r^2$ -strongly convex and its gradient is  $L = (2 + \tau^2)r^2$ -Lipschitz.

**Lemma E.1** (Spectrum of Hessian). *For the loss function (E.3) and (E.4) where  $A_{ij}$  follows the distribution (4.7), under Assumption E.1, the eigenvalues of the Hessian  $\nabla_X^2 \mathcal{L}^{(X)}(X, \bar{Y})$  and  $\nabla_Y^2 \mathcal{L}^{(Y)}(\bar{X}, Y)$  are bounded between  $(2 + \tau^2)/r^2$  and  $(2 + \tau^2)r^2$ .*

The following lemma analyzes the gradient smoothness of  $\mathcal{L}^{(X)}(\cdot, \cdot)$  with respect to the alternating variable, assessing the magnitude of  $S$ .

**Lemma E.2** (Gradient smoothness with respect to alternating variable). *Conditioned on  $X$  and  $Y$ , for the loss function (E.3) and (E.4) where  $A_{ij}$  follows the distribution (4.7), under Assumptions E.1 and E.2 consider any pairs  $\{X^*, \bar{Y}^*\} \in \{\mathcal{X}^*, \bar{\mathcal{Y}}^*\}$  and  $\{\bar{X}^*, Y^*\} \in \{\bar{\mathcal{X}}^*, \mathcal{Y}^*\}$ . Then, for  $\min\{n, p\}$  large enough, any rescaled orthonormal matrices  $\bar{Y}$  and  $\bar{X}$  with spectral norm  $\sqrt{p}$  and  $\sqrt{n}$  respectively satisfy*

$$\|\nabla_X \mathcal{L}^{(X)}(X^*, \bar{Y}^*) - \nabla_X \mathcal{L}^{(X)}(X^*, \bar{Y})\|_F \leq c\tau^2 r^2 \cdot \left( \frac{(np)^{1/2}}{p \cdot \log(\min\{n, p\})} + \frac{\tilde{d}_1}{p} \right) \|\bar{Y}^* - \bar{Y}\|_F,$$

and

$$\|\nabla_Y \mathcal{L}^{(Y)}(\bar{X}^*, Y^*) - \nabla_Y \mathcal{L}^{(Y)}(\bar{X}^{(t)}, Y^*)\|_F \leq c\tau^2 r^2 \cdot \left( \frac{(np)^{1/2}}{n \cdot \log(\min\{n, p\})} + \frac{\tilde{d}_1}{n} \right) \|\bar{X}^* - \bar{X}^{(t)}\|_F,$$

for some universal constant  $c$ .

We now analyze the difference between the minimizers in (E.3) and (E.1) (or between (E.4) and (E.2)), assessing the magnitude of  $\varepsilon_{\text{fixed}}(n, p)$ .

**Lemma E.3.** *Conditioned on  $X$  and  $Y$ , let  $A_{ij}$  follow the curved Gaussian distribution (4.7). For a fixed  $\bar{Y}$ , under the assumptions in Lemma E.1, for  $p$  large enough, with probability at least  $1 - 6/p$ ,*

$$\frac{1}{n} \|M^X(\bar{Y}) - M_n^X(\bar{Y})\|_F \leq c \left( \frac{\log^{1/4}(np)}{n^{1/2} p^{1/4}} \right),$$

where  $c$  is a constant that depends only on  $k$ ,  $\tau$  and  $r$ . Similarly, for a fixed  $\bar{X}$ , under Assumption E.1, for  $n$  large enough, with probability at least  $1 - 6/n$ ,

$$\frac{1}{p} \|M^Y(\bar{X}) - M_n^Y(\bar{X})\|_F \leq c \left( \frac{\log^{1/4}(np)}{n^{1/4} p^{1/2}} \right),$$

The above corollary means that

$$\varepsilon_{\text{fixed}}(n, p) = c \left( \max \left\{ \frac{n^{1/2}}{p^{1/4}}, \frac{p^{1/2}}{n^{1/4}} \right\} \cdot \log^{1/4}(np) \right),$$

where  $c$  is a constant that depends only on  $k$ ,  $\tau$  and  $r$ . Given the ingredients Lemma E.1 through Lemma E.3, we can derive the following result.

**Corollary E.4** (Application of Proposition D.1 to curved Gaussian model). *Consider the curved Gaussian model (4.7) where  $k$ ,  $\tau$ , and  $r$  are constant, and Assumptions E.1-E.3 hold conditioned on  $X$  and  $Y$ . Assuming  $n = O(p)$  and  $p = O(n)$  are large enough such that  $8(np)^{1/2}/\tilde{d}_k < 1$ ,  $S/\mu < 1$  and the initialization satisfies  $\|\bar{Y}^* - \bar{Y}^{(0)}\|_F \leq \varepsilon_{fixed}(n, p)/(1 - S/\mu)$ , then for the number of iterations  $T$  large enough, the eSVD described in (4.1)-(4.4) where each iteration resamples the observed matrix  $A$  achieves the rate*

$$\|\Theta - \hat{\Theta}\|_F = O_P\left(n^{3/4} \log^{1/4}(n)\right).$$

For future work, we believe techniques in Eldridge et al. (2018) and Chi et al. (2019) can help improve our theoretical analysis and simplify the assumptions in the above proposition, as it is not immediately clear yet from a theoretical perspective how often an initialization can satisfy these assumptions. Nonetheless, the following corollary is proved immediately after combining Corollary E.4 with Proposition 5.2.

**Corollary E.5** (Application of Proposition 5.2 to curved Gaussian model). *Assume all the setting and assumptions in Corollary E.4. Then, the eSVD (after reparameterizations (4.5) and (4.6)) achieves the rate*

$$\frac{1}{n} \|X - \hat{X}\|_F^2 = O_P\left(\frac{\log^{1/2}(n)}{n^{1/2}}\right).$$

## F Additional simulation details/results

Throughout this section, we use the following notation to parameterize different distributions. The negative binomial and Bernoulli distributions are parameterized by Negative Binomial( $r, p$ ) and Bernoulli( $p$ ) respectively where  $r$  is the number of failures (i.e, the dispersion parameter) and  $p$  is the probability of success. The Gamma distribution is parameterized by Gamma( $a, b$ ) where  $a$  and  $b$  are the shape and rate parameters respectively. The Poisson distribution is parameterized by Poisson( $\lambda$ ) where the mean is  $\lambda$ . (Observe that here, the distributions are parameterized by their canonical parameters, not their natural parameters.)

### F.1 Simulation setup

**Generation of natural parameters.** To sample from  $G$  (prior to identifiability conditions), we uniformly sample an equal number of points along 4 connected line segments, where the line segments collectively have endpoints at  $\{(4, 10), (25, 100), (60, 80), (40, 10), (100, 25)\}$  in the Cartesian coordinate system. We then add Gaussian noise with  $\sigma = 5$  to each of the points. This generates  $\tilde{X}_1, \dots, \tilde{X}_n$  (prior to identifiability conditions).

To sample from  $H$  (prior to identifiability conditions), we similarly uniformly sample an equal number of points along 2 disconnected line segments. One line segment goes from  $(1, 4.5)$  to  $(1.25, 5)$  while the other goes from  $(4.5, 1)$  to  $(5, 1.25)$  in the Cartesian coordinate system. We then also add Gaussian noise to each of the points. This generates  $\tilde{Y}_1, \dots, \tilde{Y}_p$  (prior to identifiability conditions). In generating either  $\tilde{X}_1, \dots, \tilde{X}_n$  or  $\tilde{Y}_1, \dots, \tilde{Y}_p$ , we threshold the values to ensure all entries are positive.

We then compute  $\Theta = \tilde{X}\tilde{Y}^\top/c$ , where  $c$  is a fixed constant chosen beforehand to ensure that the resulting values we generate reasonably lie between 0 and 1000 (depending on which distribution  $F$  is chosen). Letting the SVD of this matrix be  $\Theta = UDV^\top$ , we then output the target embedding we wish to estimate,  $X = (n/p)^{1/4} \cdot U \cdot \sqrt{D}$  and  $Y = (p/n)^{1/4} \cdot V \cdot \sqrt{D}$ , where the square root function is interpreted to be entry-wise.

**Other methods.** While SVD, NMF, ICA, UMAP, t-SNE, Isomap and diffusion maps are more common methods in statistical and genomic analyses, ZINB-WaVE and pCMF are methods more specific to single-cell analyses that we briefly overview here.

- **ZINB-WaVE:** ZINB-WaVE relies on the negative binomial distribution. Similar to our model,  $X_i$  represents the fixed lower-dimensional latent vector for each cell, and  $Y_j$  and  $W_j$  represent two sets of fixed lower-dimensional latent vectors for each gene. Let  $r_j$  denote the dispersion parameter for gene  $j$ . The generative model is, for any  $(i, j) \in \{1, \dots, n\} \times \{1, \dots, p\}$ ,

$$\begin{aligned} Z_{ij} &\sim \text{Negative Binomial}\left(r_j, \frac{\exp(X_i^\top Y_j)}{\exp(X_i^\top Y_j) + r_j}\right), \\ D_{ij} &\sim \text{Bernoulli}(1/(1 + \exp(-X_i^\top W_j))) \\ A_{ij} &= Z_{ij} \cdot D_{ij}, \end{aligned} \tag{F.1}$$

where all the latent variables are independent of one another and  $A_{ij}$ 's are all conditionally independent<sup>4</sup>. ZINB-WaVE estimates the parameters  $X$ ,  $Y$  and  $W$  via an alternating minimization scheme based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method. We note that the ZINB-WaVE model is able to handle additional covariate information about each cell or gene.

Note, compared to the eSVD model, the ZINB-WaVE model is similar since  $X_i^\top Y_j$  represents the natural parameter of a negative binomial distribution in both models. The key differences are that ZINB-WaVE allows for each gene to have a different dispersion parameter and also models dropout. However, due to the nature of the fitting procedure, ZINB-WaVE does not have a matrix-completion diagnostic to assess the quality of fit (which we developed for the eSVD in Subsection 4.2), and is limited to only modeling the negative binomial distribution.

- **pCMF:** pCMF relies on the Poisson distribution. Similar to our model,  $X_i$  represents the lower-dimensional latent vector for each cell, and  $Y_j$  represents the lower-dimensional latent vector for each gene, but explicit distributions for  $G$  and  $H$  are used to facilitate a Bayesian approach. Let  $\pi_j$  denote the unknown gene-specific dropout rate. The generative model is,

$$\begin{aligned} X_{i\ell} &\sim \text{Gamma}(\alpha_{\ell,1}, \alpha_{\ell,2}), \quad \text{for } (i, \ell) \in \{1, \dots, n\} \times \{1, \dots, k\} \\ Y_{j\ell} &\sim \text{Gamma}(\beta_{\ell,1}, \beta_{\ell,2}), \quad \text{for } (i, \ell) \in \{1, \dots, n\} \times \{1, \dots, k\} \\ Z_{ij} &\sim \text{Poisson}(X_i^\top Y_j), \quad \text{for } (i, j) \in \{1, \dots, n\} \times \{1, \dots, p\} \\ D_{ij} &\sim \text{Bernoulli}(\pi_j), \quad \text{for } (i, j) \in \{1, \dots, n\} \times \{1, \dots, p\} \\ A_{ij} &= Z_{ij} \cdot D_{ij}, \end{aligned} \tag{F.2}$$

where all the latent variables are independent of one another and  $A_{ij}$ 's are all conditionally independent. pCMF estimates the parameters via a variational EM algorithm.

Note, compared to the eSVD model, the pCMF model sets  $X_i^\top Y_j$  to be the canonical parameter for the Poisson distribution (i.e., the mean), unlike the eSVD which sets  $X_i^\top Y_j$  to be the natural parameter. Additionally, pCMF also models dropout. However, similar to the disadvantages of ZINB-WaVE mentioned above, pCMF does not have a matrix-completion diagnostic to assess the quality of fit and is limited to only modeling the Poisson distribution.

We list the R packages used for comparisons with other methods. We use the `zinbwave` package for ZINB-WaVE, and set the `K` parameter to 2, the `maxiter.optimize` parameter to 100, the

<sup>4</sup>Their paper actually parameterizes the negative binomial distribution by its mean and inverse dispersion parameter, but is equivalent to the one we describe.

`normalizedValues` parameter to `False`, and `commonDispersion` parameter to `False`. We use the `pCMF` package for pCMF and set the `K` parameter to 2 and the `sparsity` parameter to `False`.

For the remaining methods, we use the `dimRed` package for Isomap, ICA (which invokes the `fastICA` package), and NMF (which invokes the `NMF` package), and use the `density` package for diffusion maps, all of which use the default parameters to estimate a two-dimensional embedding. Lastly, for UMAP and t-SNE, we tune the methods' respective parameters in an oracle-fashion in order to maximize the relative embedding correlation described in Section 6. That is, this tuning requires knowing the true embedding, which is unrealistic in practice but demonstrates the performance of these methods under the most favorable conditions. Specifically, we use the `Rtsne` package for t-SNE, and set the `k` parameter to 2 and tune the `perplexity` parameter in an oracle fashion among 10 values between 2 and 50. We use the `umap` package for UMAP, and set the `n_components` parameter to 2 as well as `init` to `random`. Additionally, we tune the `n_neighbors` and `min_dist` parameters in an oracle fashion among the values  $\{2, 3, 5, 15, 30, 50\}$  and  $\{10^{-5}, 10^{-3}, 0.1, 0.3, 0.5, 0.9\}$  respectively (for a total of 36 different parameter settings).

## F.2 First simulation suite: Verification of convergence to $G$

Based on the first simulation suite described in Section 6, we plot the empirical performance of the eSVD when loss function is set to the negative log-likelihood of the negative binomial distribution with  $r = 50$ , which is correctly-specified model. We plot  $\|X - \hat{X}\|_F^2/n$  versus  $n$  in Figure 10.

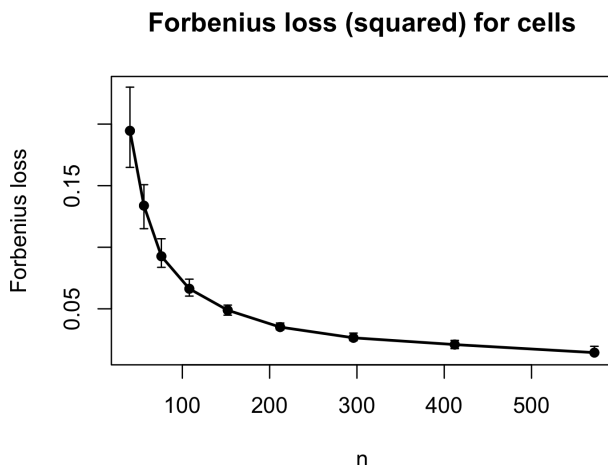


Figure 10:  $\|X - \hat{X}\|_F^2/n$  versus  $n$ , where the solid points represent the median performance over 200 trials and the error bars represent the 25th to 75th quantile.

## F.3 Second simulation suite: Additional comparison of performance under exponential-family models

In this second simulation suite described in Section 6, we compare the eSVD's performance against the nine other embedding methods where we change both the data-generating distribution and the distribution used by the eSVD simultaneously to be either the Poisson or curved Gaussian distribution. These simulations give us a more holistic understanding of how the data-generating distribution impacts the other nine embedding methods, and also allows us to demonstrate the versatility of the eSVD.

In Figure 11, we see that the eSVD and ZINB-WaVE both perform well in the Poisson simulation, similar to Figure 5. Specifically, despite ZINB-WaVE assuming a negative binomial distribution, it

still performs well in this Poisson setting. While it might be surprising that pCMF does not perform well in this setting, as we noted in Appendix F.1, pCMF assumes that  $X_i^\top Y_j$  is the canonical parameter of the Poisson distribution (i.e., the mean) while our data-generating model here assumes that  $X_i^\top Y_j$  is the natural parameter of the Poisson distribution (i.e., the mean is  $\exp(X_i^\top Y_j)$ , as described in Appendix B.3). This extra exponentiation skews the embedding of the cells outside of the eSVD and ZINB-WaVE, which both operate on the natural-parameter scale.

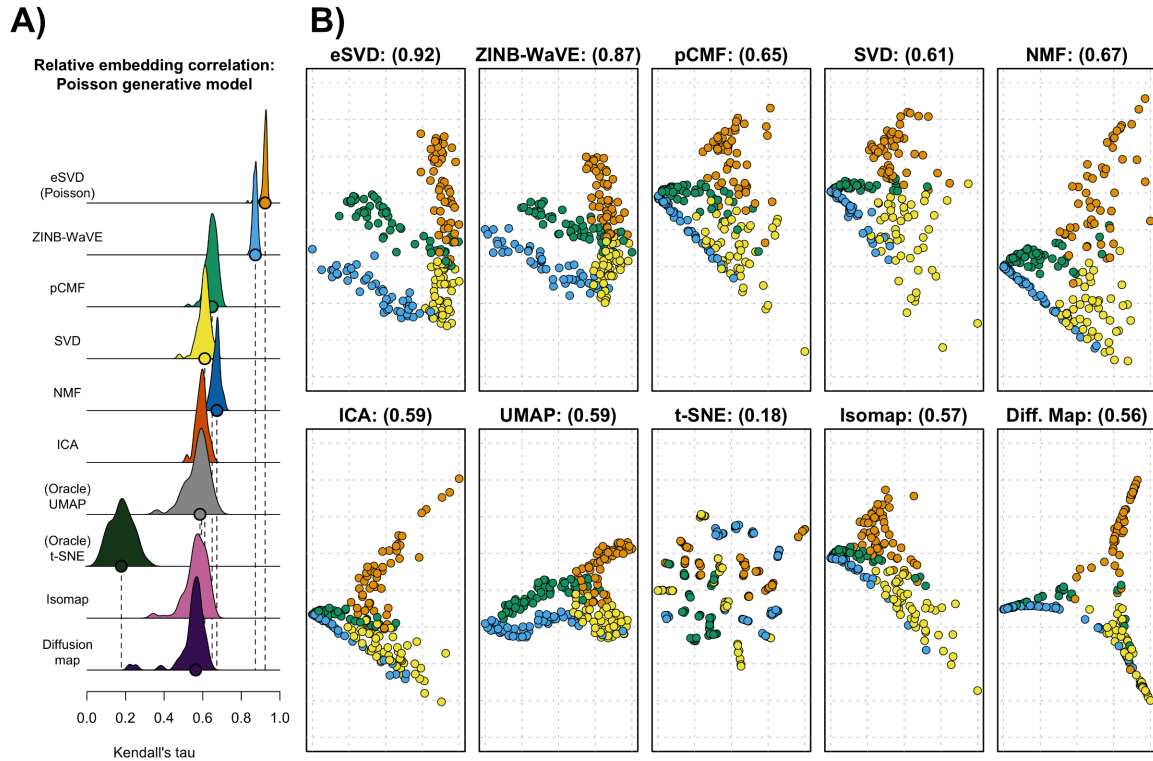


Figure 11: Results for the simulation setting where both the data-generating distribution  $F$  and the eSVD use the Poisson distribution. The layout of these plots are comparable to those in Figure 5.

In Figure 12, we see that only the eSVD performs well in the curved Gaussian simulation. Here, the true data-generating distribution uses  $\alpha = 2$ , while the eSVD uses the tuning procedure mentioned in Section 4 to select the most appropriate value of the tuning parameter  $\alpha$  from the set  $\{1, 2, 4\}$ . We believe the poor performance of the other nine embedding methods is due to the extra variability in the data – the curved Gaussian with  $\alpha = 2$  has more variability than the negative binomial distribution or Poisson distribution demonstrated in Figure 5 and Figure 11 respectively.



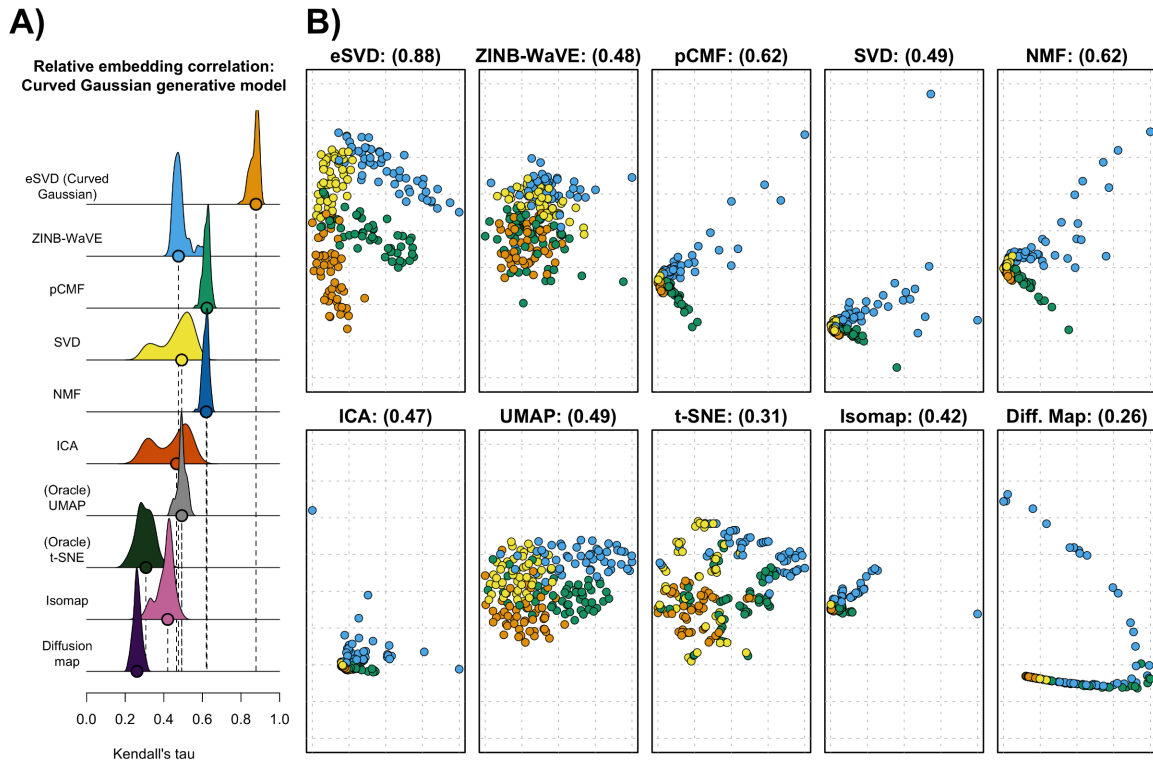


Figure 12: Results for the simulation setting where both the data-generating distribution  $F$  and the eSVD use the curved Gaussian distribution. The layout of these plots are comparable to those in Figure 5.

#### F.4 Third simulation suite: Comparison of performance under misspecified models

We present the additional simulations alluded to as the third simulation suite in Section 6. In this section, we generate models implicitly assumed in ZINB-WaVE or pCMF (see (F.1) or (F.2) respectively). While it is natural to expect that ZINB-WaVE and pCMF will perform the best in their respective simulations, we are additionally interested to see the eSVD can nonetheless be competitive. If so, this gives us empirical evidence that the eSVD's flexibility and tuning diagnostic are sufficient to capture most of the data's nuances assumed in these models, despite being misspecified.

**ZINB-WaVE model.** In this simulation setting, we build on top of the negative binomial model described in Section 6, but include additional complexities that make the simulation more realistic according to the generative model for ZINB-WaVE shown in (F.1). Specifically, using the ZINB-WaVE model (F.1), we vary the dispersion parameter so 25% of the genes have a dispersion parameter of  $r = 80$ , another 25% have a parameter of  $r = 120$ , and the remaining 50% have a parameter of  $r = 600$ . Additionally, a dropout term is now included based on a logistic model, unlike the model described in Section 6. Both of these changes ensure that the negative binomial model that the eSVD fits is sufficiently misspecified. As our simulations reassuringly shows though, the eSVD nonetheless estimates the embedding relatively well when compared to other methods aside from ZINB-WaVE itself.

When we fit the eSVD via a negative binomial model, we use the tuning procedure mentioned in Subsection 4.2 to search for a global dispersion parameter of  $r \in \{50, 100, 500, 1000\}$  and set  $k = 3$ . Notice that we are using *one* dispersion parameter to model the simulated dataset, although our

true generative model uses three different dispersion parameters. Also, we increase the latent space to  $k = 3$  since we found empirically, the addition of the dropout factor can be reasonably captured by an extra latent dimension.

We demonstrate our results in Figure 13. We see that ZINB-WaVE performs the best according to the relative embedding correlation metric, but this is unsurprising as our generative model is correctly specified for ZINB-WaVE. However, even though it is misspecified for the eSVD using the negative binomial distribution, the eSVD’s performance is still quite competitive according to our relative embedding criterion, compared among the nine remaining methods. Hence, we believe that the eSVD using the negative binomial distribution, with our tuning procedure, is comparable in performance to ZINB-WaVE in practice. As we mentioned in the main text however, the benefits of using the eSVD is that the eSVD has a solid theoretical foundation, can be easily extended to other one-parameter exponential-family distributions, and can handle missing values to assess model fit.

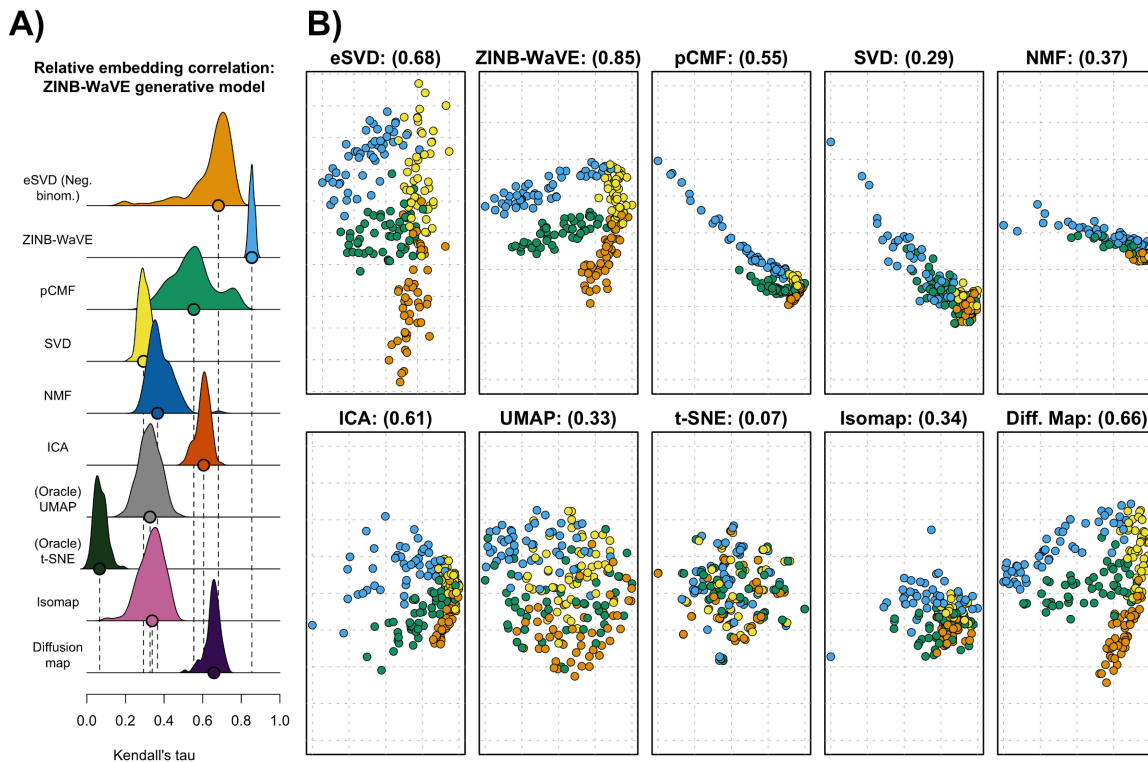


Figure 13: Results for the misspecified simulation setting where the data-generating distribution  $F$  is generated according to the assumed model by ZINB-WaVE (F.1) and the eSVD use the negative binomial distribution. The layout of these plots are comparable to those in Figure 5.

**Effect of  $k$  and appropriateness of tuning procedure.** To demonstrate the appropriateness of choosing  $k = 3$  in the above simulation under the ZINB-WaVE model, we empirically compare how different  $k$  can effect the quality of the model fit. (We have found this investigation more meaningful under the misspecified model since the effect of varying  $k$  was not apparent under any of the settings in the second simulation suite.) Specifically, while there is no “correct” value of  $k$  due to the model misspecification, we are interested if our tuning procedure can select the  $k$  that yields the most accurate embedding and assess how choosing  $k$  poorly can impact the embedding.

For this simulation, we retain the ZINB-WaVE model above and use the eSVD with the negative binomial distribution. However, we now use the tuning procedure mentioned in Subsection 4.2 to

search for a global dispersion parameter of  $r \in \{50, 100, 500\}$  as well as  $k \in \{2, 3, 10\}$ , where the quality of the model fit is computed using the negative log-likelihood as described in Appendix B.2. We use this metric as opposed to the principal angle in this particular simulation since we have already fixed the eSVD to use the negative binomial distribution.

The results in Figure 14 demonstrate that our tuning procedure selects  $k = 3$  92% of the time (among all 100 trials), which also yields the embedding with the highest relative embedding correlation on average, and selects  $k = 2$  among the remaining trials. Specifically, in Figure 14, the selected pair of parameters  $(r, k)$  based on the negative log-likelihood also often has a high relative embedding correlation. All-in-all, Figure 14 demonstrates why we chose  $k = 3$  for the eSVD when producing Figure 13, and demonstrates how our tuning procedure can select  $k$  that can yield the best-performing embedding despite the model misspecification.

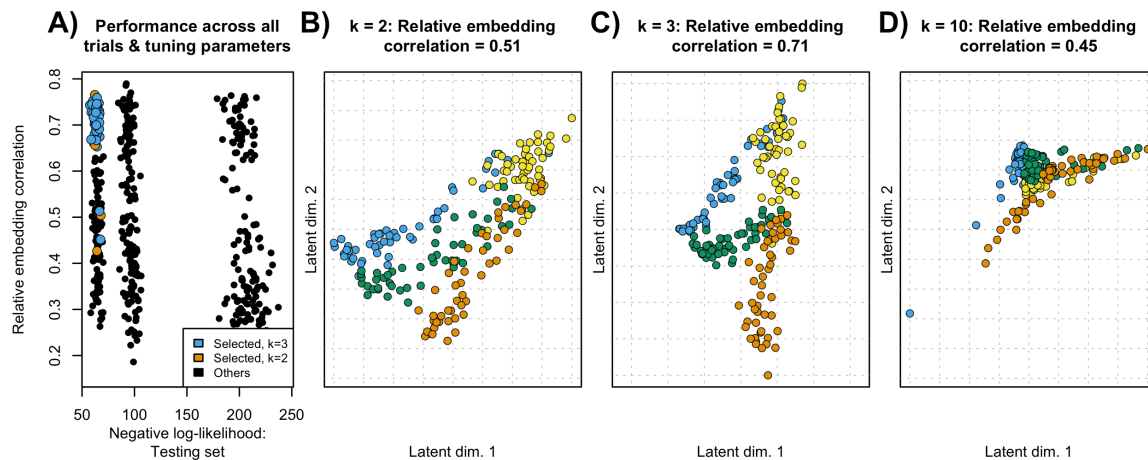


Figure 14: (A) The relative embedding correlation plotted against the negative log-likelihood of the testing set, where each point represents one of 100 trials fitting the eSVD using a different pair of parameters  $(r, k)$  among  $\{50, 100, 500\} \times \{2, 3, 10\}$ , for a total of 900 points shown. Among each of the 100 trials, the selected  $(r, k)$  parameter pair's corresponding point is highlighted in color, blue for if  $k$  is chosen to be 3, and orange if  $k$  is chosen to be 2. (B to D) The embedding for  $k = 2$ ,  $k = 3$  and  $k = 10$  respectively, trial with the median accuracy, and  $r = 50$  (the selected value of  $r$  according to our tuning procedure). The layout of these plots are comparable to those in Figure 5.

**pCMF model.** In this simulation setting, we make the simulation more realistic according to the generative model for pCMF shown in (F.2). Specifically, we set  $\pi_j = 0.3$  for all  $j \in \{1, \dots, p\}$  and use the same way to generate  $X_1, \dots, X_n$  and  $Y_1, \dots, Y_p$  as previously described in Appendix F.1, instead of generating the latent vectors from the prescribed prior distribution. While we set the eSVD to estimate the embedding according to the Poisson setting in this simulation, there are two important aspects that makes this a misspecified setting. First, pCMF assumes that the mean of the Poisson distribution is  $X_i^\top Y_j$ , while the eSVD under the Poisson model assumes that the mean of the Poisson distribution is  $\exp(X_i^\top Y_j)$ . Second, a dropout term is now included. Both of these changes ensures that the Poisson model that the eSVD fits is sufficiently misspecified. As our simulations reassuringly shows though, the eSVD nonetheless estimates the embedding relatively well when compared to other methods aside from ZINB-WaVE itself. Similar to in the ZINB-WaVE simulation above, when we fit the eSVD via a Poisson model, we set  $k = 3$ , where we found that the extra latent dimension helps to capture the dropout effect.

We demonstrate our results in Figure 15. Similar to before, we see that pCMF performs the best, but this is unsurprising. However, more surprisingly, we see that many embedding methods actually work well in this simulation setting, since the Poisson distribution has a small variance when

compared to the negative binomial or curved Gaussian distribution. Hence, the additional variability in the data is insufficient from dramatically distorting many embedding methods, including SVD, NMF, ICA and UMAP. While the eSVD does not rank among the best-performing methods in terms of the relative embedding correlation metric here (Figure 15A), we see that the eSVD’s estimated embedding visually resembles the true layout of latent vectors (Figure 15B, compared to Figure 4). Hence, given this particular simulation as well as all the simulations presented beforehand, we feel the flexibility of the eSVD makes it a competitive embedding method overall.

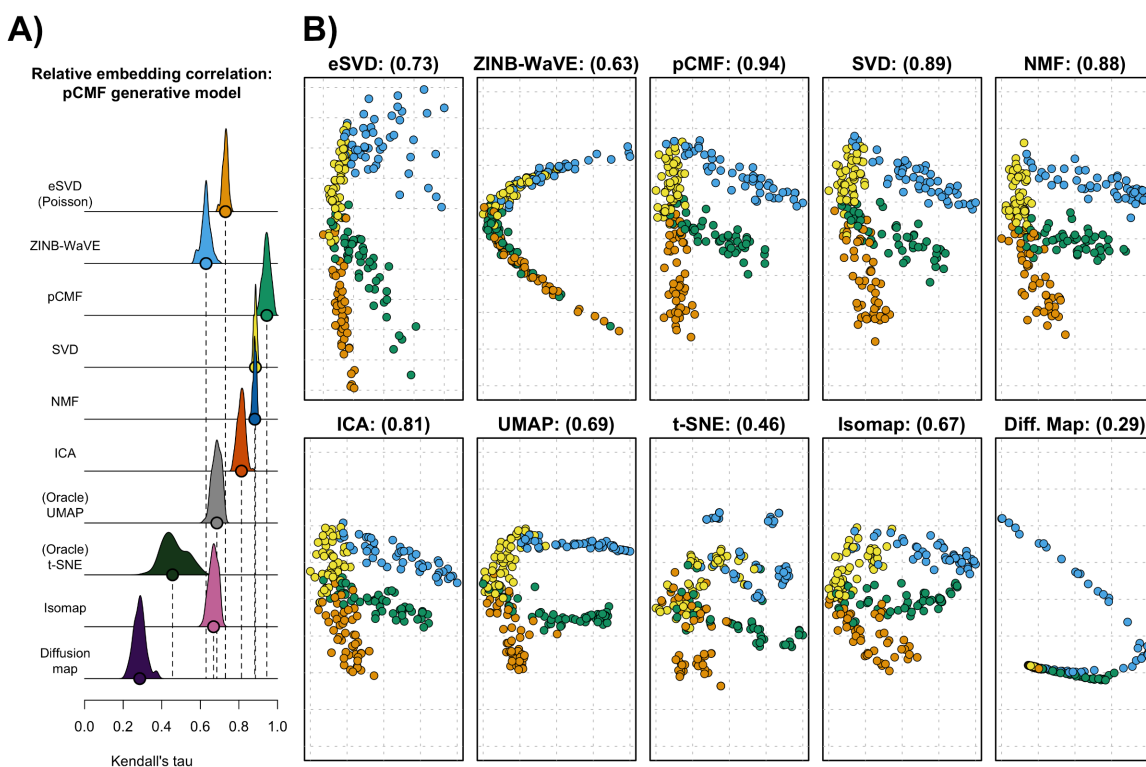


Figure 15: Results for the misspecified simulation setting where the data-generating distribution  $F$  is generated according to the assumed model by pCMF (F.2) and the eSVD use the Poisson distribution. The layout of these plots are comparable to those in Figure 5.

## G Details on Slingshot and uncertainty tube

### G.1 Modifications to Slingshot

Our implementation of Slingshot differs from its original implementation in Street et al. (2018) in a few aspects. The first two modifications is related to how the lineages (i.e., what the different branches are and the ordering of the cell sub-types) are estimated, and the last modification is related to how the trajectories are estimated given the lineages (i.e., what is the numeric curve that interpolates the points in the latent space).

- **Respecting natural order:** Our implementation respects the order among the major cell types (i.e., Pdgfra+ precursor, oligodendrocyte precursor cells, differentiation-committed oligodendrocyte precursors, newly formed oligodendrocytes, myelin-forming oligodendrocytes, and mature oligodendrocytes), but allows the lineage to arbitrarily order or split the cell sub-types within the same cell type.

- **Construction of lineage:** Our implementation determines the lineage via a shortest path tree from the starting cluster, where the distance is determined by the multivariate Hotelling’s  $t$ -test for unequal covariances. This is in contrast with the original Slingshot that uses a minimum spanning tree, where the distance is determined by the multivariate Hotelling’s  $t$ -test for equal covariances.
- **Enable upsampling:** The original Slingshot does not weight clusters, so the lineage curves naturally gravitated towards the larger clusters. To compensate for this phenomenon, we upsampled the cells in each cluster via sampling with replacement so each cluster has the same number of cells. This effectively adds larger weights to these smaller clusters so each cluster is treated equally regardless of its original size. A similar modification is done in Herman and Grün (2018).

## G.2 Construction of uncertainty tube

We construct the uncertainty tubes via a bootstrap-based approach to determine if the different cell developmental trajectories are substantially different. This is done via the following procedure.

1. **Bootstrapping:** For a specific trial, sample with replacement the low-dimensional embedding  $\hat{X}_1, \dots, \hat{X}_n$  among each cell sub-type. This generates a new dataset with the same proportion of cell sub-types. Run Slingshot on this new dataset, having fixed the order of the cell sub-types in each lineage, to obtain a new set of trajectories.
2. **Computing the quantile of  $\ell_2$  distance between lineage curves:** Let  $\hat{T}$  be one of the estimated trajectories based on the original estimated embedding  $\hat{X}_1, \dots, \hat{X}_n$ . For each estimated trajectory  $T_b$  from the newly generated dataset that matches  $\hat{T}$  based on the order of cell sub-types, compute the pointwise  $\ell_2$  distance between the two respective curves. Specifically, letting  $\{T_{b,1}, \dots, T_{b,N}\}$  and  $\{\hat{T}_1, \dots, \hat{T}_M\}$  denote the  $N$  and  $M$  discrete  $k$ -dimensional points in order that represent the trajectory  $T_b$  and  $\hat{T}$  respectively, this pointwise  $\ell_2$  distance is computed as the 95% quantile of the following set

$$\left\{ \min_{j \in \{1, \dots, M\}} \|T_{b,i} - \hat{T}_j\|_2, \forall i \in \{1, \dots, N\} \right\}.$$

After computing this distance for each of the bootstrapped trajectories, let  $\sigma(\hat{T})$  be the the 95th quantile  $\ell_2$  distance among all the bootstrapped trajectories, denoting the “margin of error” for a particular original estimated trajectory  $\hat{T}$ . Let  $\sigma$  be the 95th quantile of such values among all the original estimated trajectories, representing the width of the uncertainty tubes.

3. **Computing the lineage tube:** For each lineage in the original dataset, construct a “tube” of radius  $\sigma$  around the lineage curve.

## H Additional analysis results

### H.1 Additional plots of results

In Figure 16 and Figure 17, we show the two-dimensional and three-dimensional plots of the estimated embeddings to provide more clarity to Figure 1 and Figure 6 in the main text.

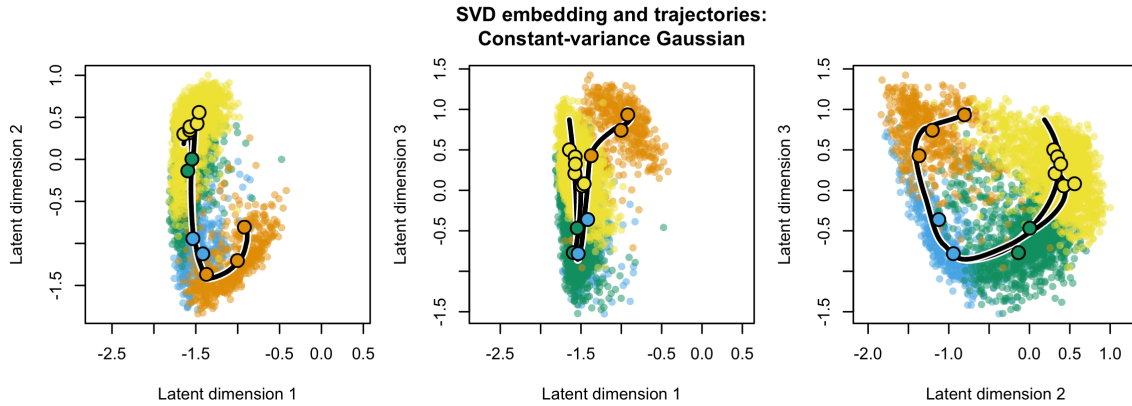


Figure 16: Two-dimensional plots of the SVD embedding, which corresponds to the three-dimensional plots shown in Figure 6. Each cell is color coded using the same color scheme from the aforementioned figure, the large dots represent the cluster centers for each of the thirteen cell sub-types, and the estimated trajectories are overlaid on top, which correspond to the same trajectories in the aforementioned figure. The three plots correspond to the three pairs of latent dimensions, one of the plots being the same as Figure 1.



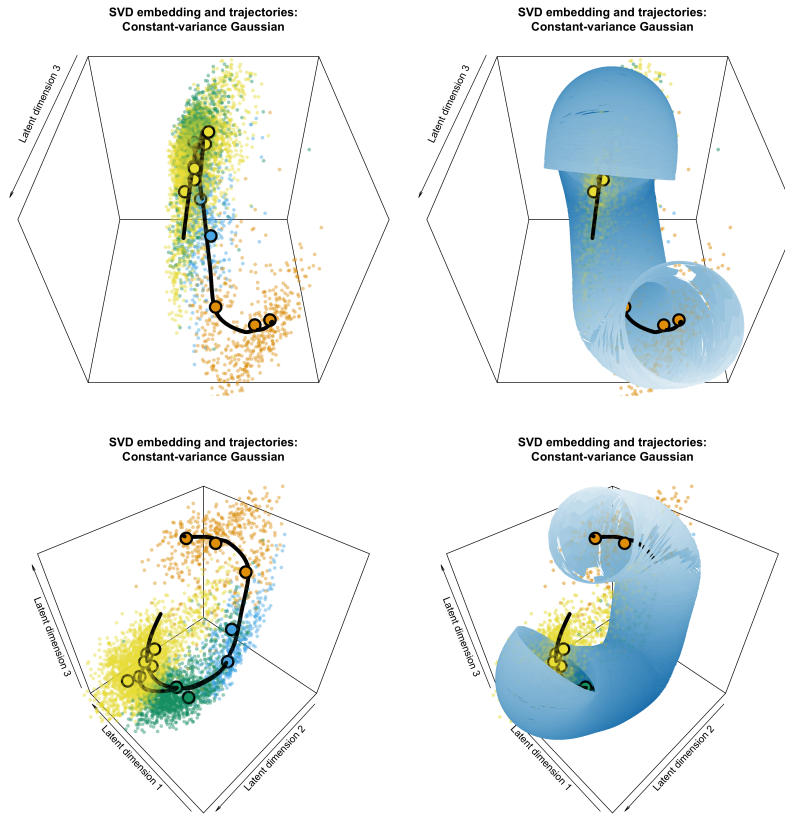


Figure 17: Three-dimensional plots of the estimated latent positions via the SVD embedding, without and with the uncertainty tube overlaid onto. This plot is of the same estimated embedding as shown in Figure 6 but shown from a different perspective (each perspective represented by a different row).

In Figure 18 and Figure 20, we show the two-dimensional and three-dimensional plots of the estimated embeddings to provide more clarity to plots shown in Figure 8 in the main text.

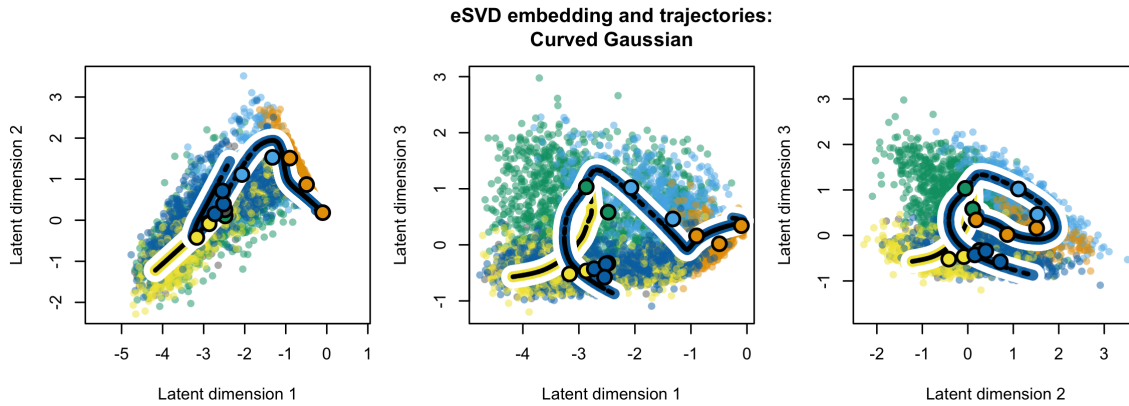


Figure 18: Two-dimensional plots of the eSVD embedding using the curved Gaussian distribution with  $k = 5$  and  $\tau = 2$ , which corresponds to the three-dimensional plots shown in Figure 8. Each cell is color coded using the same color scheme from the aforementioned figure, the large dots represent the cluster centers for each of the thirteen cell sub-types, and the estimated trajectories are overlaid on top, which correspond to the same trajectories in the aforementioned figure.

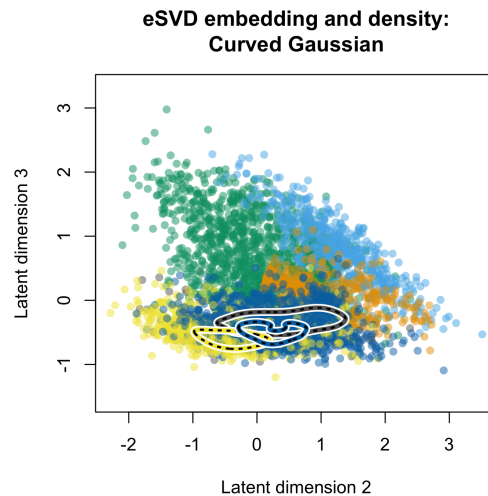


Figure 19: Two-dimensional plot of the eSVD embedding, similar to that shown in Figure 18, but with high-density regions denoted for the mature oligodendrocytes shared between the two trajectories (gray points), and the mature oligodendrocytes unique to each of the estimated trajectories (either blue or yellow points). This plot is directly comparable to Figure 1.

Additionally, Figure 21 shows the same plot as Figure 7, except zoomed in to display the behavior of points near zero. We see that the same trends persists from Figure 7 (mainly, the variance increasing with the mean).

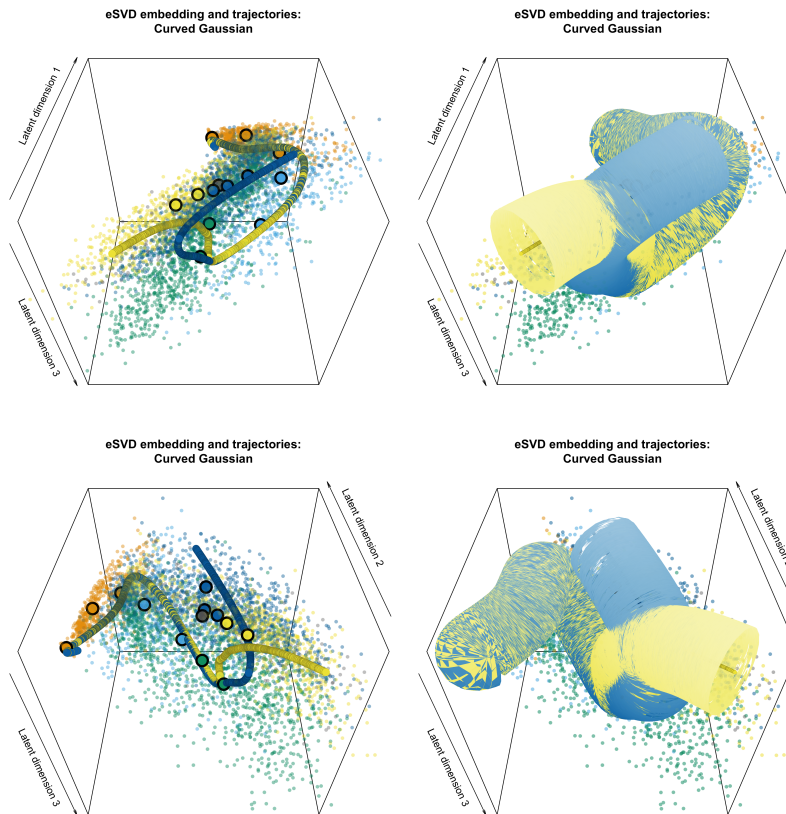


Figure 20: Three-dimensional plots of the estimated latent positions via the eSVD embedding, without and with the uncertainty tubes overlaid onto. This plot is of the same estimated embedding as shown in Figure 8 but shown from a different perspective (each perspective represented by a different row).

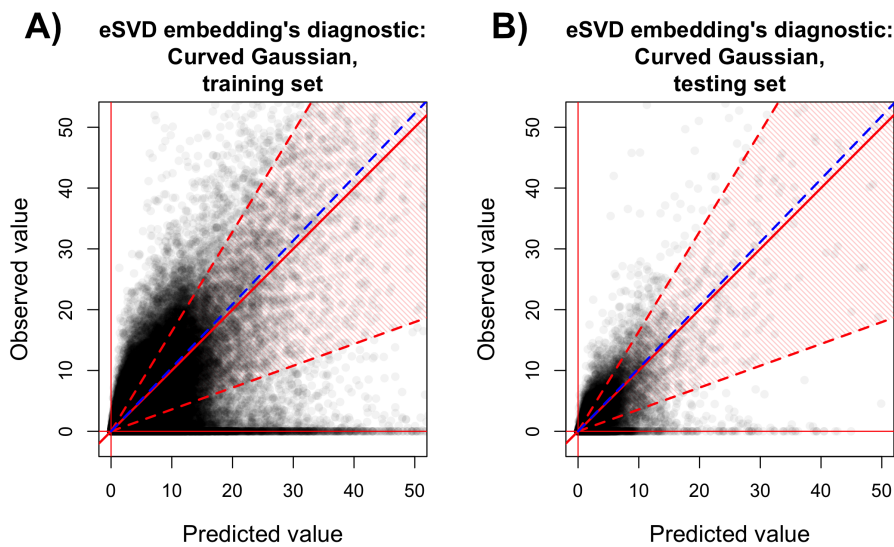


Figure 21: Diagnostic plot showing the same results as in Figure 7, except zoomed in to display values between 0 and 50. (A) The diagnostic plot on the “training set.” (B) The diagnostic plot on the “testing set.”

## H.2 Additional result for the highly informative genes

To supplement the results in Section 7, we find high informative genes across the different six major cell types which also demonstrate the smooth continuum of gene expression along the two estimated trajectories. This is visualized in Figure 22, where heatmaps of this form are common in work that study the gene expression continuum along trajectories (Marques et al. (2016), Haghverdi et al. (2016) and Bergen et al. (2020)). These heatmaps are sometimes referred to as *cascade plots*. Here, we see that when we order the cells from youngest to oldest based on the estimated pseudotime, there is an ordering of genes such that different genes are highly expressed at different pseudotimes. Additionally, we see that when the two trajectories diverge around the rescaled pseudotime of 0.49, there are genes that are highly expressed for mature oligodendrocytes in one trajectory but not the other.

We briefly describe the procedure to select the genes shown in Figure 22. First, we remove cells that do not easily conform to the estimated trajectories, as by the nature of Slingshot, certain cells belonging to both trajectories might have two distinctly different estimated pseudotimes. We have 4449 cells remaining afterwards, which are used to find highly informative genes from the 983 genes in our analysis. We then order the cells based on their pseudotimes and which trajectories they belong to. This results in two different data matrices, one matrix for the cells shared between both trajectories and the cells unique to the first trajectory, and another matrix for the cells shared between both trajectories and the cells unique to the second trajectory. For each data matrix, we smooth the gene expression across each gene using a univariate kernel nonparametric regression via the function `np::npreg`. Then, for each gene  $j$ , we use a variant of circular binary segmentation (Olshen et al., 2004) to find a region of cells along either data matrix that represents cells which have high expression for gene  $j$ . After all this is done, we find the genes that show the highest change in expression (based on circular binary segmentation on the smoothed gene expression) along each value of the pseudotime for both trajectories. This informs us about which genes are informative for the shared trajectory, only the first trajectory, or only the second trajectory. We additionally include marker genes published in Zhang et al. (2014) for reference, which are originally used in the dataset we are investigating Marques et al. (2016).

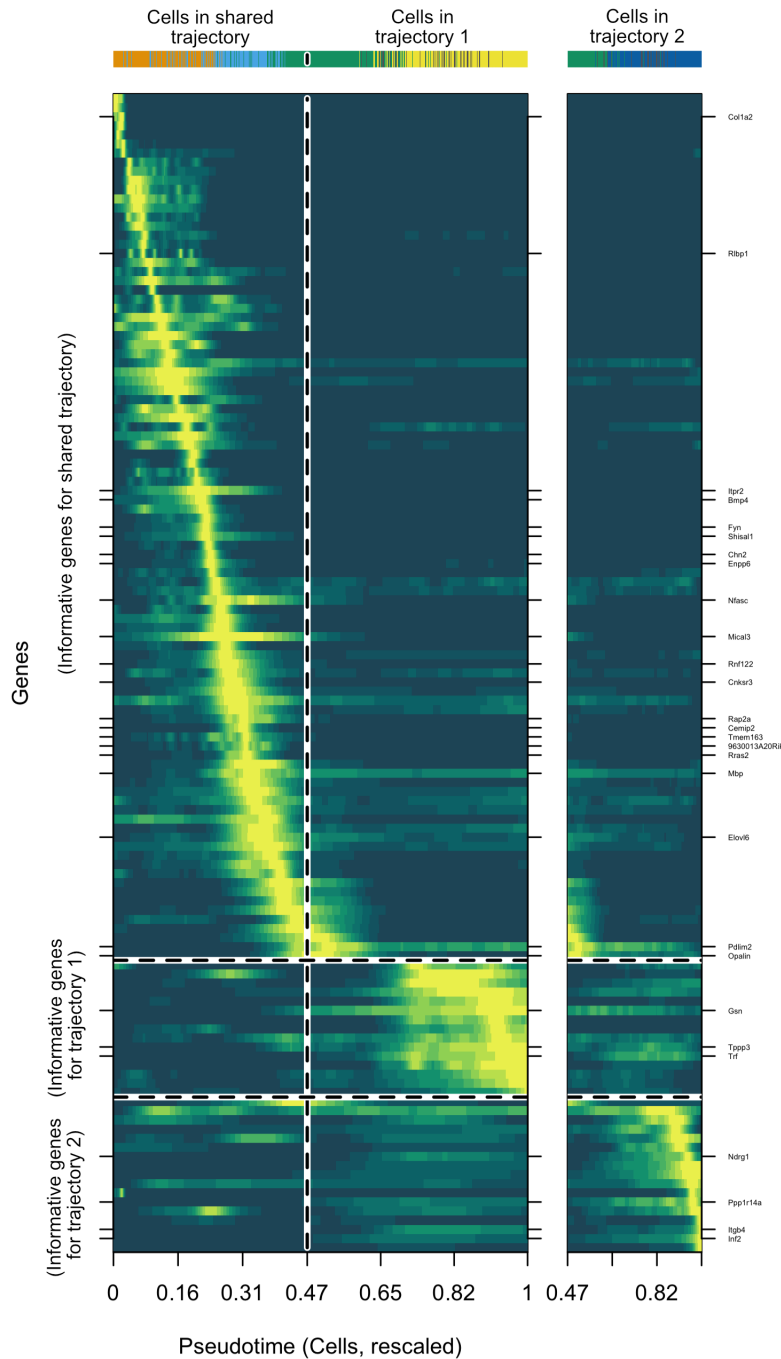


Figure 22: Cascade plot showing the gene expression (rescaled, smoothed) across the cells ordered by the estimated pseudotime. Here, each row denotes a different gene and each column denotes a different cell, where the cell type of each of the corresponding 4449 columns is shown in the top horizontal bar, where the colors correspond to the coloring used in Figure 8. Within this heatmap, yellow denotes high intensity of expression, while dark green denotes low intensity, both relative to all the other cells for a particular gene. The cells prior to a (rescaled) pseudotime of 0.49 are shared between both trajectories, while the cells start to diverge after that pseudotime. The marker genes published in Zhang et al. (2014) for OPCs, NFOLs, and MOLs are explicitly marked in their respective rows for reference.

### H.3 Analysis via eSVD using the negative binomial distribution

For completeness, we include the diagnostic plots on the quality of fit of using the negative binomial distribution to fit the eSVD embedding on the oligodendrocyte data. We tune the negative binomial from a grid of parameters as well, where we choose among the  $k \in \{3, 5, 10\}$  and dispersion parameter  $r \in \{500, 2500, 10^4\}$ . After using our tuning procedure described in Subsection 4.2, we select  $k = 5$  and  $r = 10^4$ . Analogous to Figure 7, we show the resulting plots on the training and testing set in Figure 23. As we can see, compared to analogous plots for the curved Gaussian distribution (Figure 7), the 10th to 90th quantile of the negative binomial model, denoted by the shaded red polygon, are not wide enough to capture as much of the variability in the data in the testing set.

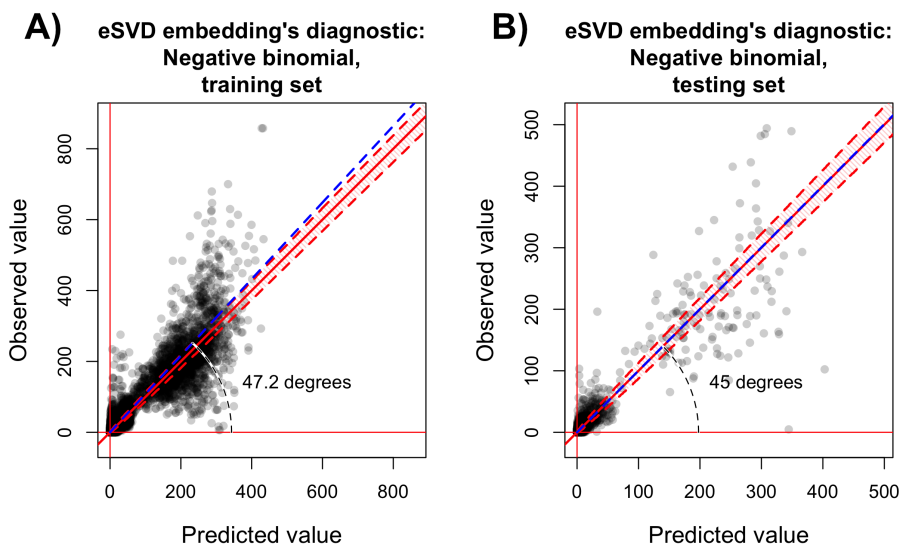


Figure 23: Diagnostic plots analogous to those in Figure 7, except displaying the results of fitting the eSVD embedding with the negative binomial distribution where, after applying the tuning procedure, the number of failures is set to be  $r = 10000$  and the latent dimension space is  $k = 5$ . (A) The diagnostic plot on the “training set.” (B) The diagnostic plot on the “testing set.”

### H.4 Analysis via UMAP and ZINB-WaVE

As mentioned in the main text, two common embedding methods used in the literature are UMAP (McInnes et al., 2018) and ZINB-WaVE (Risso et al., 2018). For completeness, we present these resulting embeddings derived from these methods for comparison against the eSVD embedding with the curved Gaussian distribution. Our goal here is to convey that using any of these embedding results all qualitatively conclude that the mature oligodendrocytes unique to the first estimated trajectory lie opposite to the mature oligodendrocytes unique to the second estimated trajectory.

In Figure 24, we present the results using the UMAP embedding. Here, we use the `umap` R package, where we set `n_neighbors` to be 30. Analogous to Figure 8, we color the mature oligodendrocytes as gray, yellow or dark blue, depending on whether the cell belongs to the mature oligodendrocyte cell sub-type that is shared between the two estimated trajectories, is exclusively in one of the estimated trajectories, or exclusively in the other estimated trajectory respectively. This UMAP embedding is calculated based on the same dataset used to estimate the eSVD embedding shown in Section 7 (i.e., not  $\log_2$ -transformed). Since the UMAP embedding does not rely on a statistical model and does not have known statistical consistency, we avoid applying Slingshot on top of this embedding. Nonetheless, we qualitatively see in Figure 24 that the gray points lie between the dark blue and



yellow points, suggesting a similar phenomenon to our results in Section 7 where we observe two distinct trajectories that split at the mature oligodendrocytes.

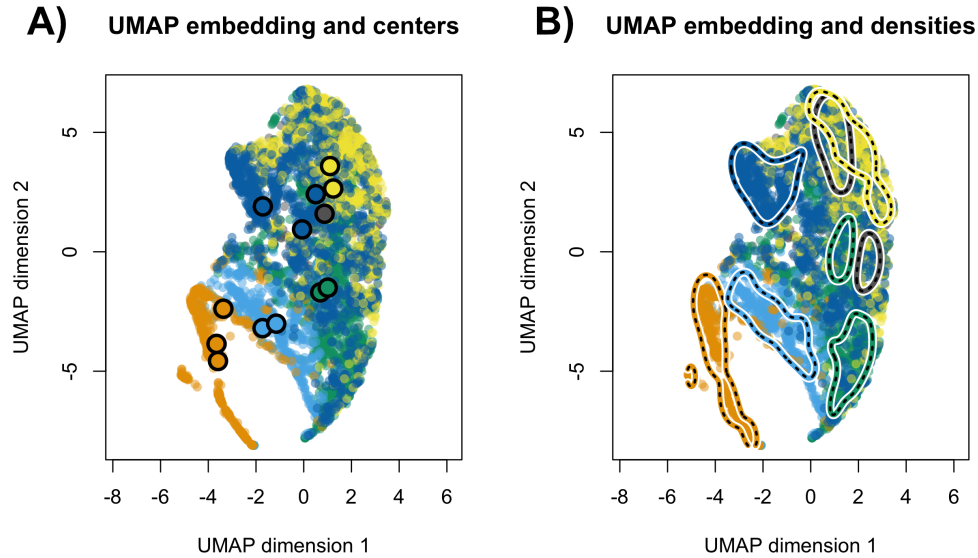


Figure 24: Two-dimensional plot of the UMAP embedding applied to the preprocessed dataset with 5069 cells and 983 genes, (i.e., Steps 1 and 2 as described in Appendix C.1), where the cell types are colored according to Figure 18. (A) UMAP embedding shown alongside the 13 cell cluster centers. (B) UMAP embedding shown alongside with estimated densities to visualize high-density regions (analogous to Figure 19).

Similarly, we present the results using the ZINB-WaVE embedding. This model and its usage is summarized in Appendix F.1. We fit the model using the count data (i.e.,  $n \times p$  matrix with non-negative integer values) that has undergone only the gene screening step (i.e., preprocessing the data using only Step 1 described in Appendix C.1) and the arguments `K=5`, `maxiter.optimize=100`, `normalizedValues=T` and `commonDispersion=F`. Similar results in Section 7, the ZINB-WaVE embedding shown in Figure 25 show the gray points lie between the dark blue points and yellow points. This is most visibly seen in the scatter plot between the first two latent dimensions. Hence, these results also suggest a similar phenomenon to our results in Section 7 where we observe two distinct trajectories that split at the mature oligodendrocytes.

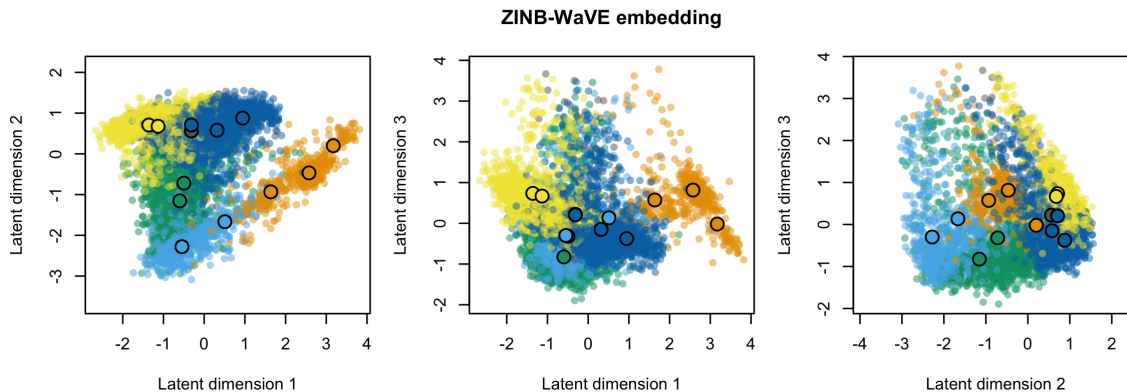


Figure 25: Two-dimensional plots of the ZINB-WaVE embedding applied to the preprocessed count dataset with 5069 cells and 983 genes, (i.e., Step 1 as described in Appendix C.1). The cell types and 13 cell cluster centers are shown to be analogous to those in Figure 18.

## H.5 Additional analysis on a separate dataset

Here, we apply the eSVD to a second dataset to demonstrate its performance on other single-cell datasets outside of the oligodendrocyte dataset studied in the main text. For this, we investigate the eSVD applied to the Zeisel dataset, which is commonly used as a benchmark dataset in other studies (Lun et al., 2016; Zappia et al., 2017; McCarthy et al., 2017; Kiselev et al., 2017; Li and Li, 2018). This dataset studies cells in mice’s nervous systems, and contains 3005 cells across seven different primary cell types (astrocytes ependymal, endothelial mural, interneurons, microglia, oligodendrocytes, pyramidal CA1, and pyramidal SS), of which the least represented cell type has 98 cells. The goal in this particular analysis is to determine if cells of the same cell type are “well-clustered” after embedding each cell into the latent space.

Our analysis strategy proceeds similarly to the one described in Appendix B. Specifically, we first screen the cells in a similar fashion, with exception that now we normalize each cell by its library size prior to screening for genes. After screening, there are 1628 genes remaining. Then, we compare two analyses, comparable to those described in the main text. In the first analysis, we  $\log_2$ -transform all the entries of the preprocessed single-cell data and then embed each cell via the SVD. In the second analysis, we do not  $\log_2$ -transform the entries, but instead rescale the data uniformly so the maximum value is 1000 and embed each cell via the eSVD with a negative binomial distribution. Our choice of analyzing the Zeisel dataset via the negative binomial distribution here is to demonstrate the eSVD’s versatility in how it can be usefully applied to one-parameter exponential-family distributions other than the curved Gaussian distribution used in Section 7, and we also empirically observe that this data has less noise compared to the oligodendrocyte dataset studied in the main text. We tune the latent dimensionality of both embeddings (among  $k \in \{10, 20, 50\}$ ) and the nuisance parameter for the negative binomial distribution (among  $r \in \{250, 500, 1000, 10000, 50000\}$ ) via the tuning procedure described in Subsection 4.2. For the SVD embedding, this resulted in selecting  $k = 50$ , and for the eSVD embedding, this resulted in selecting  $k = 20$  and  $r = 10000$ .

To assess how well the cells are clustered within each embedding, we derive a “purity” statistic based on the the shortest path between cells of the same cell type in the nearest-neighbor graph. We do not use statistics based on K-means clustering or other clustering methods since we wish to not confound the quality of the embedding with the appropriateness of the clustering method as much as possible. To describe this statistic, first construct the nearest-neighbor graph by representing each of the 3005 cells as a vertex, and connecting each vertex to its  $m$  nearest neighbors (for example,  $m = 3$ ). Then, let cell  $i$  and  $j$  be of the same cell type  $t$  (for example, interneurons), and consider

the shortest path between vertices  $i$  and  $j$ . We measure the purity statistic between vertices  $i$  and  $j$  by the percentage of vertices in this shortest path that correspond to cells of the same cell type  $t$ , excluding cells  $i$  and  $j$  themselves. This statistic is in  $[0, 1]$  and a larger value is more desirable, as it suggests that more cells of the same cell type are “between” cells  $i$  and  $j$ . If there are multiple shortest paths, we pick the one that maximizes this statistic. For each of the seven different cell types, we sample 5000 random pairs of cells  $i$  and  $j$  to measure this purity statistic. This results in 35000 values that describe the quality of the embedding.

Using the purity statistic, we see that the eSVD embedding (without the  $\log_2$ -transform) preserves the clustering structure among the seven cell types better than the SVD embedding (with the  $\log_2$ -transform). Specifically, if we were to construct a nearest-neighbor graph based on  $m = 3$ , then the average (mean) purity statistics across all 35000 values for the eSVD embedding and the SVD embedding are 0.93 and 0.91 respectively. However, this statistic is already equal to 1 at the 25th quantile among the 35000 values for either embedding. When we focus on values between the 0th and 25th quantile only, the average (mean) purity statistic for the eSVD embedding and the SVD embedding are 0.73 and 0.65 respectively. When we try different values of  $m$  from  $m = 3$  to  $m = 5$ , we observe a similar result regardless of how we construct the nearest-neighbor graph (Figure 26). This suggests that our analysis results are robust to the value of  $m$ . Choosing  $m$  to be smaller than 3 results in a nearest-neighbor graph that is not connected.

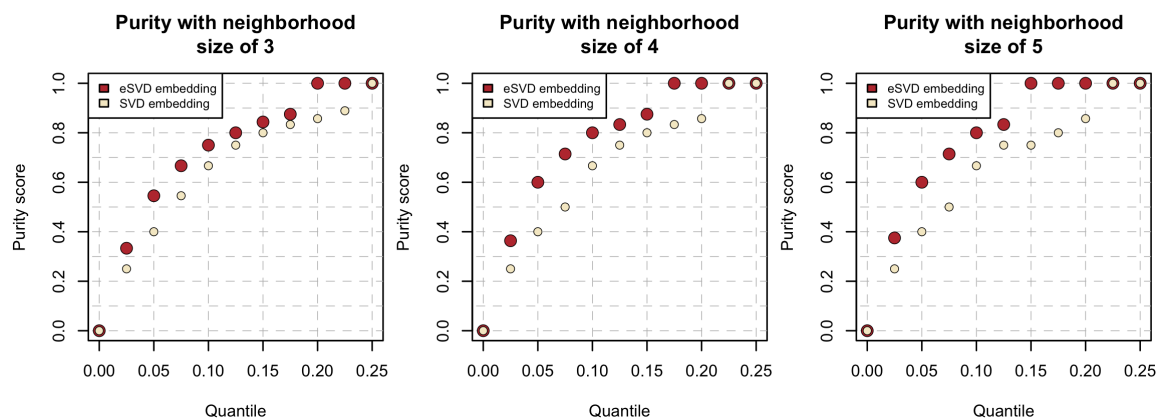


Figure 26: Quantiles of the purity score among the 35000 values for both embeddings when applied to the Zeisel dataset, from the 0th to 25th quantile, across  $m = 3$  to  $m = 5$ .

## I Proofs

In Appendix I.1, we prove Proposition 5.1. In Appendix I.2, we prove Proposition 5.2 and its related lemmas. In Appendix I.3, we prove Proposition D.1 and its related lemmas. In Appendix I.4, we prove the results shown in Appendix E.

Throughout these proofs, for generic matrices  $A$  and  $B$ , let  $\|A\|_{\text{op}}$  denote the spectral norm, i.e., the largest singular value. If  $A$  is a square matrix, let  $\text{tr}(A)$  denote the trace of  $A$ , i.e., the sum of its diagonal elements. We write  $A \otimes B$  to denote the Kronecker product of  $A$  and  $B$ , and if  $A$  and  $B$  are of the same dimensions, we write  $A \preceq B$  if  $B - A$  is positive semidefinite. For two deterministic sequences  $a_n$  and  $b_n$ , let  $a_n = o(b_n)$  denote that  $a_n/b_n \rightarrow 0$ . For two random sequences  $A_n$  and  $B_n$ , let  $A_n = \Omega_p(B_n)$  denote  $B_n/A_n$  is bounded in probability for large enough  $n$ . Also, let  $I_k$  denote the identity matrix of size  $k \times k$  and  $\mathbf{1}_{n \times p}$  represent the  $n \times p$  matrix of all ones. Throughout these proofs, we implicitly use different equivalent definitions of strong convexity and functions with Lipschitz gradients as stated in Zhou (2018).

## I.1 Proof for Proposition 5.1

*Proof of Proposition 5.1.* Define the eigendecompositions of the second moment matrices for all  $i \in \{1, \dots, n\}$  and  $j \in \{1, \dots, p\}$ ,

$$C_X^* = \mathbb{E}[X_i' X_i'^\top] = \Phi \Lambda \Phi^\top, \quad \text{and} \quad C_Y^* = \mathbb{E}[Y_j' Y_j'^\top] = \Psi \Gamma \Psi^\top, \quad (\text{I.1})$$

where  $\Phi$  and  $\Psi$  are both  $k \times k$  unitary matrices. Our construction of the invertible matrix  $R \in \mathbb{R}^{k \times k}$  will be done in two steps. In the first step, we first construct an invertible matrix  $\tilde{R} \in \mathbb{R}^{k \times k}$  such that

$$\tilde{R} C_X^* \tilde{R}^\top = \tilde{R}^{-\top} C_Y^* \tilde{R}^{-1}. \quad (\text{I.2})$$

This would yield a transformation matrix to ensure  $\mathbb{E}[X_i X_i^\top] = \mathbb{E}[Y_i Y_i^\top]$ . In the second step, we adjust  $\tilde{R}$  into the desired matrix  $R$  in order to ensure that both  $\mathbb{E}[X_i X_i^\top]$  and  $\mathbb{E}[Y_i Y_i^\top]$  are diagonal.

**Step 1:** Based on the definition (I.1) and the desired goal shown in (I.2), an equivalent goal is to show that

$$\tilde{R}^\top \tilde{R} C_X^* \tilde{R}^\top \tilde{R} = \Psi \Gamma \Psi^\top. \quad (\text{I.3})$$

Let  $Q \in \mathbb{R}^{k \times k}$  denote any unitary matrix, to be specified later. Since the matrices on both sides of the above display are symmetric, in order to show (I.3), we can instead we can find an  $Q$  and  $\tilde{R}$  such that,

$$Q \Lambda^{1/2} \Phi^\top \tilde{R}^\top \tilde{R} = \Gamma^{1/2} \Psi^\top.$$

Rearranging the above display, we have

$$\tilde{R}^\top \tilde{R} = \underbrace{\Phi \Lambda^{-1/2} Q^\top \Gamma^{1/2} \Psi^\top}_B.$$

Hence, we are done once we construct a unitary matrix  $Q$  that makes  $B$  symmetric. Observe that if the matrix

$$Q^\top \Gamma^{1/2} \Psi^\top \Phi \Lambda^{1/2}$$

were symmetric, then  $B$  would be symmetric. (This can be seen by multiplying the above matrix on the left by  $E = \Phi \Lambda^{-1/2}$  and on the right by  $E^\top$ .) Observe that  $\Gamma^{1/2} \Psi^\top \Lambda^{-1/2} \Phi$  is guaranteed to be full rank (by assumption of  $C_X^*$  and  $C_Y^*$  being full rank), so  $\Gamma^{1/2} \Psi^\top \Lambda^{-1/2} \Phi$  admits a rank- $k$  SVD of  $U D V^\top$ . Since the product of two unitary matrices is still unitary, we set  $Q = U V^\top$ . Hence, we finished our construction of  $\tilde{R}$ .

**Step 2:** Suppose  $\tilde{R} C_X^* \tilde{R}^\top$  (or equivalently,  $\tilde{R}^{-\top} C_Y^* \tilde{R}^{-1}$  based on (I.2)) has eigenvectors  $W_1, \dots, W_k$ . Let  $W$  be the unitary matrix formed by concatenating these  $k$  eigenvectors column-wise. By diagonalization, we know  $W^\top (\tilde{R} C_X^* \tilde{R}^\top) W$  is diagonal. This implies that our final construction is  $R = W^\top \tilde{R}$ .  $\square$

## I.2 Proof for Proposition 5.2

We first introduce some notation. Let the SVD of  $X$  and  $Y$  be denoted as

$$X = \sqrt{n} U \Lambda^{1/2} \Phi^\top, \quad \text{and} \quad Y = \sqrt{p} V \Gamma^{1/2} \Psi^\top,$$

Using this definition, the empirical second moment matrices are

$$C_X = X^\top X / n = \Phi^\top \Lambda \Phi, \quad \text{and} \quad C_Y = Y^\top Y / p = \Psi^\top \Gamma \Psi.$$

Prior to the proof, we offer a high-level description of the proof strategy. Similar to the proof in Lei (2018), we introduce two levels of approximation. Let us focus on estimating  $X_1^{(C)}, \dots, X_n^{(C)}$ , the latent variables that are oriented based on the population covariance matrix  $C_X^*$ . Formally,

$$X^{(C)} = X \Phi^*.$$

(Recall by Assumption 5.2, since  $C^*X$  is diagonal, the columns of  $\Phi^*$  are the standard basis vectors.) To approximate this, we consider  $X_1^{(S)}, \dots, X_n^{(S)}$ , the latent variables rotated by their own right singular vectors,

$$X^{(S)} = X\Phi = \sqrt{n}U\Lambda^{1/2}.$$

This approximation is driven by  $C_X$  being close to  $\widehat{C}_X$ . This is in turn estimated by  $X_1^{(\Theta)}, \dots, X_n^{(\Theta)}$  based on the SVD of  $\Theta$ ,

$$X^{(\Theta)} = \left(\frac{n}{p}\right)^{1/4} \widetilde{U} \widetilde{D}^{1/2}.$$

This approximation is driven by  $C_X^*$  being equal to  $C_Y^*$ . Finally, this is approximated by a quantity we can actually compute from data, our estimates  $\widehat{X}_1, \dots, \widehat{X}_n$  as in (4.5),

$$\widehat{X} = \left(\frac{n}{p}\right)^{1/4} \widehat{U} \widehat{D}^{1/2},$$

where  $\widehat{U}$  and  $\widehat{D}$  are obtained by an SVD of our estimate  $\widehat{\Theta}$  (recalling that we denote the SVD of  $\widehat{\Theta}$  as  $\widehat{\Theta} = \widehat{U} \widehat{D} \widehat{V}^\top$ ). This approximation is driven by  $\widehat{\Theta}$  being close to  $\Theta$  (i.e., our assumption that  $\|\Theta - \widehat{\Theta}\|_F < \epsilon$ ).

*Proof of Proposition 5.2. Step 1:* (Decomposition of error). By the triangle inequality, the rate is dictated by the term,

$$\frac{1}{n} \max \left\{ \|X^{(C)} - X^{(S)}\|_F^2, \|X^{(S)} - X^{(\Theta)}\|_F^2, \|X^{(\Theta)} - \widehat{X}\|_F^2 \right\}.$$

In the following three steps, we bound each term individually, of which the maximum of all three terms concludes the proof.

**Step 2:** (Approximation from  $X^{(C)}$  to  $X^{(S)}$ ). First, we deduce the relations of the eigenvalue  $\lambda_j$  and eigenvectors  $\Phi_j$ . By applying Lemma J.1, we obtain

$$\|C_X^* - C_X\|_{\text{op}} = O_P((k/n)^{1/2}). \quad (\text{I.4})$$

By using Weyl's inequality and the Davis-Kahan theorem along with Assumption 5.2, this immediately implies

$$\|\Phi_j^* - \Phi_j\|_2 = O_P(j^\beta (k/n)^{1/2}), \quad \forall j \in \{1, \dots, k\}, \quad (\text{I.5})$$

$$|\lambda_j^* - \lambda_j| = O_P((k/n)^{1/2}), \quad \forall j \in \{1, \dots, k\}. \quad (\text{I.6})$$

Since  $k = o(\min\{n, p\})$  by assumption, this implies

$$\lambda_j = O_P(j^{-\alpha}), \quad \forall j \in \{1, \dots, k\}. \quad (\text{I.7})$$

Thus, using the trace function,

$$\begin{aligned} \frac{1}{n} \|X^{(S)} - X^{(C)}\|_F^2 &= \frac{1}{n} \|X(\Phi^* - \Phi)\|_F^2 \\ &= \text{tr} \left\{ (\Phi^* - \Phi)^\top C_X (\Phi^* - \Phi) \right\} \\ &= \text{tr} \left\{ (\Phi^* - \Phi)^\top (C_X - C_X^*) (\Phi^* - \Phi) \right\} + \\ &\quad \text{tr} \left\{ (\Phi^* - \Phi)^\top C_X^* (\Phi^* - \Phi) \right\}. \end{aligned}$$

For the first term,

$$\begin{aligned}
& \left| \operatorname{tr} \left\{ (\Phi^* - \Phi)^\top (C_X - C_X^*) (\Phi^* - \Phi) \right\} \right| \\
& \leq \sum_{j=1}^k \left| (\Phi_j^* - \Phi_j)^\top (C_X - C_X^*) (\Phi_j^* - \Phi_j) \right| \\
& \leq \sum_{j=1}^k \|C_X - C_X^*\|_{\text{op}} \|\Phi_j^* - \Phi_j\|_2^2 = O_P(k^{2\beta+5/2}/n^{3/2}).
\end{aligned}$$

For the second term, recalling that  $\alpha \leq \beta$  and Parseval's identity,

$$\begin{aligned}
& \operatorname{tr} \left\{ (\Phi^* - \Phi)^\top C_X (\Phi^* - \Phi) \right\} \\
& = \sum_{i=1}^k (\Phi_i^* - \Phi_i)^\top \left[ \sum_{j=1}^k \lambda_j^* \Phi_j^* \Phi_j^{*\top} \right] (\Phi_i^* - \Phi_i) \\
& = \sum_{j=1}^k \lambda_j^* \left\{ \sum_{i=1}^k [(\Phi_i^* - \Phi_i)^\top \Phi_j^*]^2 \right\} \\
& = \sum_{j=1}^k \lambda_j^* \left\{ [(\Phi_j^* - \Phi_j)^\top \Phi_j^*]^2 + \sum_{i \neq j, i \leq k} [(\Phi_i^* - \Phi_i)^\top \Phi_j^*]^2 \right\} \\
& = \sum_{j=1}^k \lambda_j^* \left\{ [(\Phi_j^* - \Phi_j)^\top \Phi_j^*]^2 + \sum_{p \neq j, i \leq k} [(\Phi_j^* - \Phi_j)^\top \Phi_i]^2 \right\} \\
& \leq 2 \sum_{j=1}^k \lambda_j^* \|\Phi_j^* - \Phi_j\|_2^2 = O_P(k^{2\beta-\alpha+2}/n).
\end{aligned}$$

**Step 3:** (Approximation from  $X^{(S)}$  to  $X^{(\Theta)}$ ). First, observe that  $\Theta = XY^\top = \sqrt{np}U\Lambda^{1/2}\Phi^\top\Psi\Gamma^{1/2}V^\top$ , and we would like show it is close to the matrix  $\sqrt{np}U\Lambda^{1/2}\Gamma^{1/2}V^\top$ , which has an SVD of  $U(\sqrt{np}\Lambda^{1/2}\Gamma^{1/2})V^\top$ . Specifically,

$$\begin{aligned}
& \|\sqrt{np}U\Lambda^{1/2}\Gamma^{1/2}V^\top - \sqrt{np}U\Lambda^{1/2}\Phi^\top\Psi\Gamma^{1/2}V^\top\|_{\text{op}} \leq \\
& \quad \sqrt{np}\|\Lambda^{1/2}\|_{\text{op}}\|\Gamma^{1/2}\|_{\text{op}}\|I_k - \Phi^\top\Psi\|_{\text{op}},
\end{aligned} \tag{I.8}$$

where we can bound the last term by using the submultiplicative property of the spectral norm and  $C_X^* = C_Y^*$ ,

$$\|I_k - \Phi^\top\Psi\|_{\text{op}} = \|\Phi^\top(\Phi - \Psi)\|_{\text{op}} \leq \|\Phi - \Phi^*\|_{\text{op}} + \|\Psi^* - \Psi\|_{\text{op}} = O_P\left(\frac{k^{\beta+3/2}}{(\min\{n, p\})^{1/2}}\right).$$

where the last inequality is due to Davis-Kahan. Hence, plugging the above display into (I.8),

$$\|\sqrt{np}U\Lambda^{1/2}\Gamma^{1/2}V^\top - \sqrt{np}U\Lambda^{1/2}\Phi^\top\Psi\Gamma^{1/2}V^\top\|_{\text{op}} = O_P\left(\frac{(np)^{1/2} \cdot k^{\beta+3/2}}{(\min\{n, p\})^{1/2}}\right). \tag{I.9}$$

First, we derive the difference in eigenvalues based on (I.9). We apply Weyl's inequality to the above display to conclude

$$|(np\lambda_j\gamma_j)^{1/2} - \tilde{d}_j| = O_P\left(\frac{(np)^{1/2} \cdot k^{\beta+3/2}}{(\min\{n, p\})^{1/2}}\right), \quad \forall j \in \{1, \dots, k\}, \tag{I.10}$$



recalling that  $\tilde{d}_j$  is the  $j$ th singular value of  $\Theta$ . Note from (I.6) that we also have for all  $j \in \{1, \dots, k\}$ ,

$$|\lambda_j - \gamma_j| \leq |\lambda_j - \lambda_j^*| + |\gamma_j^* - \gamma_j| = O_P(k^{1/2} / \min\{n, p\}^{1/2}). \quad (\text{I.11})$$

Hence, combining (I.10) and (I.11), noting that  $\max\{|\lambda_j - \lambda_j^{1/2} \gamma_j^{1/2}|, |\gamma_j - \lambda_j^{1/2} \gamma_j^{1/2}|\} \leq |\lambda_j - \gamma_j|$ , we derive

$$|n\lambda_j - (n/p)^{1/2} \tilde{d}_j| = O_P\left(\frac{n \cdot k^{\beta+3/2}}{(\min\{n, p\})^{1/2}}\right), \quad \forall j \in \{1, \dots, k\}. \quad (\text{I.12})$$

Combining the above display with Assumption 5.2 and (I.7) and given  $k = o(\min\{n, p\})$ , this implies

$$\tilde{d}_j = O_P((np)^{1/2} j^{-\alpha}), \quad \forall j \in \{1, \dots, k\}. \quad (\text{I.13})$$

Second, we derive the difference in eigenvectors based on (I.9). Observe that based on (I.6) and  $k = o(\min\{n, p\})$ , we can derive that  $|(\lambda_j^*)^{1/2} (\gamma_j^*)^{1/2} - \lambda_j^{1/2} \gamma_j^{1/2}|$  is dominated by  $\Omega_P(j^{-\beta})$ . Hence, we can derive the spacing of the singular values of  $\sqrt{np} U \Lambda^{1/2} \Gamma^{1/2} V^\top$ ,

$$(np)^{1/2} (|\lambda^{1/2} \gamma^{1/2} - \lambda_{j+1}^{1/2} \gamma_{j+1}^{1/2}|) = \Omega_P((np)^{1/2} j^{-\beta}), \quad \forall j \in \{1, \dots, k-1\}.$$

Hence, using the Davis-Kahan theorem by combining (I.9) with the above display, we conclude

$$\|U_j - \tilde{U}_j\|_2 = O_P\left(\frac{j^\beta k^{\beta+3/2}}{(\min\{n, p\})^{1/2}}\right), \quad \forall k \in \{1, \dots, k\}. \quad (\text{I.14})$$

Using (I.7), (I.12), and (I.14), along with Lemma J.3,

$$\begin{aligned} \frac{1}{n} \|X^{(S)} - X^{(\Theta)}\|_F^2 &= \frac{1}{n} \sum_{j=1}^k \|(n\lambda_j)^{1/2} U_j - (n/p)^{1/4} \tilde{d}_j^{1/2} \tilde{U}_j\|_2^2, \\ &\leq \frac{2}{n} \sum_{j=1}^k \|(n\lambda_j)^{1/2} (U_j - \tilde{U}_j)\|_2^2 + \|(n\lambda_j)^{1/2} - (n/p)^{1/4} \tilde{d}_j^{1/2}\|_2^2 \|\tilde{U}_j\|_2^2, \\ &\leq \frac{2}{n} \sum_{j=1}^k n\lambda_j \|U_j - \tilde{U}_j\|_2^2 + O\left(\left(\frac{|n\lambda_j - (n/p)^{1/2} \tilde{d}_j|}{(n\lambda_j)^{1/2}}\right)^2\right) \\ &= O_P(k^{4\beta-\alpha+4} / \min\{n, p\}). \end{aligned}$$

**Step 4:** (Approximation from  $X^{(\Theta)}$  to  $\hat{X}$ ). By assumption, we have

$$\|\Theta - \hat{\Theta}\|_F \leq \epsilon. \quad (\text{I.15})$$

This implies  $\|\Theta - \hat{\Theta}\|_{\text{op}} \leq (2k)^{1/2} \epsilon$ . By Weyl's inequality, we conclude

$$|(n/p)^{1/2} \tilde{d}_j - (n/p)^{1/2} \hat{d}_j| = O_P((kn)^{1/2} \epsilon / p^{1/2}), \quad \forall j \in \{1, \dots, k\}. \quad (\text{I.16})$$

In addition, following a similar logic as above, using the Davis-Kahan theorem along with (I.12) to control the spacing of the eigenvalues, we can derive

$$\|\tilde{U}_j - \hat{U}_j\|_2 = O_P(j^\beta k^{1/2} \epsilon / (np)^{1/2}), \quad \forall j \in \{1, \dots, k\}. \quad (\text{I.17})$$

Hence, analogous to the derivation above, using (I.13), (I.16), and (I.17), along with Lemma J.3,

$$\begin{aligned}
\frac{1}{n} \|X^{(\Theta)} - \widehat{X}\|_F^2 &= \frac{1}{n} \sum_{j=1}^k \|(n/p)^{1/4} \widetilde{d}_j^{1/2} \widetilde{U}_j - (n/p)^{1/4} \widehat{d}_j^{1/2} \widehat{U}_j\|_2^2 \\
&\leq \frac{2}{n} \sum_{j=1}^k \|(n/p)^{1/4} \widetilde{d}_j^{1/2} (\widetilde{U}_j - \widehat{U}_j)\|_2^2 + \|(n/p)^{1/4} (\widetilde{d}_j^{1/2} - \widehat{d}_j^{1/2}) \widehat{U}_j\|_2^2 \\
&\leq \frac{2}{n} \sum_{j=1}^k (n/p)^{1/2} \widetilde{d}_j \|\widetilde{U}_j - \widehat{U}_j\|_2^2 + O\left(\left(\frac{|(n/p)^{1/2} \widetilde{d}_j - (n/p)^{1/2} \widehat{d}_j|}{(n/p)^{1/4} \widetilde{d}_j^{1/2}}\right)^2\right) \\
&= O_P(k^{2\beta-\alpha+2} \max(\epsilon^2, \epsilon)/(np)).
\end{aligned}$$

□

### I.3 Proofs for Proposition D.1

Similar to Balakrishnan et al. (2017), we first analyze the behavior of an iteration of eSVD when using the population loss function  $\mathcal{L}(\cdot, \cdot)$ . Then, by leveraging Assumption D.3, we can analyze the behavior of eSVD when applied on the sample loss function  $\mathcal{L}_n(\cdot, \cdot)$ .

**Lemma I.1.** *Assume the population loss function  $\mathcal{L}(\cdot, \cdot)$  satisfies Assumptions D.1 and D.2. Conditioned on  $X$  and  $Y$ , then for any iteration  $t \in \{1, \dots, T\}$ , there exists matrices  $\{X^*, \overline{Y}^*\} \in \{\mathcal{X}^*, \overline{\mathcal{Y}}^*\}$  and  $\{\overline{X}^*, Y^*\} \in \{\overline{\mathcal{X}}^*, \mathcal{Y}^*\}$  such that*

$$\begin{aligned}
\|X^* - M^X(\overline{Y}^{(t)})\|_F &\leq \frac{S}{\mu} \|\overline{Y}^* - \overline{Y}^{(t)}\|_F, \\
\|Y^* - M^Y(\overline{X}^{(t)})\|_F &\leq \frac{S}{\mu} \|\overline{X}^* - \overline{X}^{(t)}\|_F.
\end{aligned}$$

*Proof.* Observe that by the first-order optimality conditions, for any  $\overline{Y}^* \in \overline{\mathcal{Y}}^*$ , we have

$$\left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^*), X' - M^X(\overline{Y}^*) \right\rangle \geq 0, \quad \text{for all } X' \text{ such that } (X'_i)^\top \overline{Y}_j^* \in \mathcal{R} \quad \forall (i, j)$$

and similarly,

$$\left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^{(t)}), \overline{Y}^{(t)}), X' - M^X(\overline{Y}^{(t)}) \right\rangle \geq 0, \quad \text{for all } X' \text{ such that } (X'_i)^\top \overline{Y}_j^{(t)} \in \mathcal{R} \quad \forall (i, j)$$

By combining the two inequalities by setting  $X' = M^X(\overline{Y}^{(t)})$  in the first display and  $X' = M^X(\overline{Y}^*)$  in the second display with some algebra, we get

$$\begin{aligned}
&\left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^{(t)}), \overline{Y}^{(t)}) - \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^{(t)}), M^X(\overline{Y}^{(t)}) - M^X(\overline{Y}^*) \right\rangle \\
&\leq \left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^*) - \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^{(t)}), M^X(\overline{Y}^{(t)}) - M^X(\overline{Y}^*) \right\rangle. \tag{I.18}
\end{aligned}$$

By manipulating the properties of strong convexity assumed in Assumption D.1, we can lower bound the left-hand term in (I.18) by

$$\begin{aligned}
&\left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^{(t)}), \overline{Y}^{(t)}) - \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^{(t)}), M^X(\overline{Y}^{(t)}) - M^X(\overline{Y}^*) \right\rangle \\
&\geq \mu \|M^X(\overline{Y}^*) - M^X(\overline{Y}^{(t)})\|_F^2 = \mu \|X^* - X^{(t+1)}\|_F^2, \tag{I.19}
\end{aligned}$$

where we plugged in the definition of  $X^* \in \mathcal{X}^*$  and  $X^{(t+1)}$  in the last display (see (D.5)).

Similarly, by the Lipschitz smoothness assumed in Assumption D.1 with Cauchy-Schwarz inequality, we can upper bound the right-hand term in (I.18) by

$$\begin{aligned} & \left\langle \nabla_X \mathcal{L}(M^X(\bar{Y}^*), \bar{Y}^*) - \nabla_X \mathcal{L}(M^X(\bar{Y}^{(t)}), \bar{Y}^{(t)}), M^X(\bar{Y}^{(t)}) - M^X(\bar{Y}^*) \right\rangle \\ & \leq S \|X^* - X^{(t+1)}\|_F \|\bar{Y}^* - \bar{Y}^{(t)}\|_F. \end{aligned} \quad (\text{I.20})$$

Combining (I.19) and (I.20) into the original equation (I.18) yields

$$\mu \|X^* - X^{(t+1)}\|_F^2 \leq S \|X^* - X^{(t+1)}\|_F \|\bar{Y}^* - \bar{Y}^{(t)}\|_F,$$

which completes the proof after simple rearrangements.  $\square$

Similar to Zhao et al. (2015), we now prove Proposition D.1 using the above lemma in conjunction with Lemma J.2 which describes the effect of the LeftSVD operator.

*Proof of Proposition D.1.* For simplicity, let  $\kappa = S/\mu$ . By triangle inequality and Lemma I.1, we have that there exists some  $\{X^{*(t)}, \bar{Y}^{*(t)}\} \in \{\mathcal{X}^*, \bar{\mathcal{Y}}^*\}$  and  $\{\bar{X}^{*(t)}, Y^{*(t)}\} \in \{\bar{\mathcal{X}}^*, \mathcal{Y}^*\}$  for all  $t \in \{0, \dots, T\}$ ,

$$\begin{aligned} \|X^{*(t)} - M_n^Y(\bar{Y}^{(t)})\|_F & \leq \|X^{*(t)} - M^X(\bar{Y}^{(t)})\|_F + \|M^X(\bar{Y}^{(t)}) - M_n^X(\bar{Y}^{(t)})\|_F \\ & \leq \kappa \|\bar{Y}^* - \bar{Y}^{(t)}\|_F + \varepsilon_{\text{unif}}(n, p). \end{aligned} \quad (\text{I.21})$$

Applying Lemma J.2 and recalling that the smallest non-zero singular value of  $X^{*(t)}$  is  $\tilde{d}_k/p^{1/2}$ , we obtain

$$\|\bar{X}^{*(t+1)} - \bar{X}^{(t+1)}\|_F \leq \frac{8(np)^{1/2}}{\tilde{d}_k} \left( \varepsilon_{\text{unif}}(n, p) + \kappa \|\bar{Y}^{*(t)} - \bar{Y}^{(t)}\|_F \right). \quad (\text{I.22})$$

Similarly,

$$\|\bar{Y}^{*(t)} - \bar{Y}^{(t)}\|_F \leq \frac{8(np)^{1/2}}{\tilde{d}_k} \left( \varepsilon_{\text{unif}}(n, p) + \kappa \|\bar{X}^{*(t)} - \bar{X}^{(t)}\|_F \right). \quad (\text{I.23})$$

Hence, observe that by the conditions of Proposition D.1, we are ensured that if  $\|\bar{Y}^{*(0)} - \bar{Y}^{(0)}\|_F \leq \omega$  (Assumption D.4), then by iteratively applying (I.21) through (I.23), then  $\max\{\|\bar{X}^{*(t)} - \bar{X}^{(t)}\|_F, \|\bar{Y}^{*(t)} - \bar{Y}^{(t)}\|_F\} \leq \omega$  for all  $t \in \{1, \dots, T\}$ . Therefore, by infinite geometric summation,

$$\begin{aligned} \|\bar{X}^{*(T)} - \bar{X}^{(T)}\|_F & \leq \frac{8(np)^{1/2}}{\tilde{d}_k} \left( \varepsilon_{\text{unif}}(n, p) + \kappa \|\bar{Y}^{*(T-1)} - \bar{Y}^{(T-1)}\|_F \right) \\ & \leq \frac{8(np)^{1/2}}{\tilde{d}_k} \left( \varepsilon_{\text{unif}}(n, p) + \kappa \cdot \frac{8(np)^{1/2}}{\tilde{d}_k} \left( \varepsilon_{\text{unif}}(n, p) + \kappa \|\bar{X}^{*(T-1)} - \bar{X}^{(T-1)}\|_F \right) \right) \\ & \leq \dots \leq \frac{8(np)^{1/2}}{\tilde{d}_k - 8(np)^{1/2}\kappa} \varepsilon_{\text{unif}}(n, p) + \left( \frac{8(np)^{1/2}\kappa}{\tilde{d}_k} \right)^{(2T-1)} \|\bar{Y}^{*(0)} - \bar{Y}^{(0)}\|_F. \end{aligned}$$

Once again, by the conditions of Proposition D.1, we are ensured that

$$\frac{8(np)^{1/2}}{\tilde{d}_k - 8(np)^{1/2}\kappa} < 1, \quad \text{and} \quad \lim_{T \rightarrow \infty} \left( \frac{8(np)^{1/2}\kappa}{\tilde{d}_k} \right)^{(2T-1)} = 0.$$

Therefore, for large enough  $T$ , we conclude for some universal constant  $C$ ,

$$\begin{aligned}
\|\Theta - \widehat{\Theta}\|_F &= \|\overline{X}^{*(T)}(Y^{*(T)})^\top - \overline{X}^{(T)}(Y^{(T)})^\top\|_F \\
&= \|\overline{X}^{*(T)}(Y^{*(T)})^\top - \overline{X}^{(T)}(Y^{*(T)})^\top + \overline{X}^{(T)}(Y^{*(T)})^\top - \overline{X}^{(T)}(Y^{(T)})^\top\|_F \\
&\leq \underbrace{\|Y^{*(T)}\|_2}_{\widetilde{d}_1/n^{1/2}} \|\overline{X}^{*(T)} - \overline{X}^{(T)}\|_F + \underbrace{\|\overline{X}^{(T)}\|_2}_{n^{1/2}} \|Y^{*(T)} - Y^{(T)}\|_F \\
&= C \cdot \left( \frac{\max\{n^{1/2}, \widetilde{d}_1/n^{1/2}\} \cdot (np)^{1/2}}{\widetilde{d}_k - 8 \max\{n, p\}^{1/2} \kappa} \cdot \varepsilon_{\text{unif}}(n, p) \right).
\end{aligned}$$

□

## I.4 Proofs for Lemma E.1 through Corollary E.5

**Useful facts** It is useful to have the following forms of the gradients and Heissan written down.

- The gradient  $\nabla_X \mathcal{L}^{(X)}(X, \overline{Y}) \in \mathbb{R}^{n \times k}$  has the  $i$ th row equal to, for  $i \in \{1, \dots, n\}$ ,

$$\frac{1}{p} \sum_{j=1}^p \left( -\frac{1}{X_i^\top \overline{Y}_j} - \tau^2 \mathbb{E}[A_{ij}] + \tau^2 \mathbb{E}[A_{ij}^2] (X_i^\top \overline{Y}_j) \right) \overline{Y}_j.$$

- Let  $\nabla_Y [\nabla_X \mathcal{L}^{(X)}(X, \overline{Y})] \in \mathbb{R}^{(nk) \times (pk)}$  denote the gradient of above function with respect to  $\overline{Y}$ . If we focus on a particular block of  $k$  rows corresponding to a specific  $X_i$  and a particular block of  $k$  columns corresponding to a specific  $\overline{Y}_j$ , we have

$$\frac{1}{p} \left[ \left( -\frac{1}{X_i^\top \overline{Y}_j} + \tau^2 \mathbb{E}[A_{ij}] - \tau^2 \mathbb{E}[A_{ij}^2] (X_i^\top \overline{Y}_j) \right) I_k + \left( \frac{1}{(X_i^\top \overline{Y}_j)^2} + \tau^2 \mathbb{E}[A_{ij}^2] \right) X_i \overline{Y}_j^\top \right].$$

- The Hessian matrix  $\nabla_X^2 \mathcal{L}^{(X)}(X, \overline{Y}) \in \mathbb{R}^{(nk) \times (nk)}$  is all 0, except for  $n$   $k \times k$  blocks along the diagonal. The  $i$ th block is equal to, for  $i \in \{1, \dots, n\}$ ,

$$\frac{1}{p} \sum_{j=1}^p \left( \frac{1}{(X_i^\top \overline{Y}_j)^2} + \tau^2 \mathbb{E}[A_{ij}^2] \right) \overline{Y}_j (\overline{Y}_j)^\top.$$

We list properties about the curved Gaussian distribution assumed in (3.1) that will be needed. Recall that  $A_{ij} \sim N(\mu_{ij}, \mu_{ij}^2/\tau^2)$ , where  $\mu_{ij} = -1/\theta_{ij}$ .

- (First and second moment):

$$\mathbb{E}[A_{ij}] = 1/\theta_{ij}, \quad \text{and} \quad \mathbb{E}[A_{ij}^2] = \left( \frac{1}{\tau^2} + 1 \right) \frac{1}{\theta_{ij}^2}. \tag{I.24}$$

- (Squared random variable):

$$A_{ij}^2 \stackrel{d}{=} \frac{1}{(\tau \theta_{ij})^2} Z_{ij}, \quad \text{where} \quad Z_{ij} \sim \chi^2(k=1, \lambda = \tau^2).$$

Hence, the variance of  $A_{ij}^2$  is upper-bounded by

$$\kappa = \frac{2(1 + 2\tau^2) \cdot r^4}{\tau^4}. \tag{I.25}$$

*Proof of Lemma E.1.* We start with the lower bound. Observe that minimum eigenvalue of one of the  $n \times k$  blocks will be the minimum eigenvalue of the overall Hessian matrix  $\nabla_X^2 \mathcal{L}^{(X)}(X, \bar{Y})$ . Hence, inspecting one particular block for a specific  $i \in \{1, \dots, n\}$ , observe that

$$\frac{1}{p} \sum_{j=1}^p \left( \frac{1}{(X_i^\top \bar{Y}_j)^2} + \tau^2 \mathbb{E}[A_{ij}^2] \right) \bar{Y}_j (\bar{Y}_j)^\top \succeq \frac{2 + \tau^2}{r^2} \cdot \frac{1}{p} \sum_{j=1}^p \bar{Y}_j (\bar{Y}_j)^\top,$$

where  $\sum_j \bar{Y}_j (\bar{Y}_j)^\top = pI_k$  since the columns of  $\bar{Y}$  are orthogonal. For the upper bound, we apply the same logic.  $\square$

*Proof of Lemma E.2.* Recall by the multivariate Taylor expansion (Feng et al., 2013), we have

$$\nabla_X \mathcal{L}^{(X)}(X^*, \bar{Y}^*) - \nabla_X \mathcal{L}^{(X)}(X^*, \bar{Y}) = \int_0^1 \nabla_Y [\nabla_X \mathcal{L}(X^*, \bar{Y}^* + tu)] du \cdot t,$$

where  $t = \bar{Y}^* - \bar{Y}$ . This leads to the following inequality,

$$\|\nabla_X \mathcal{L}^{(X)}(X^*, \bar{Y}) - \nabla_X \mathcal{L}^{(X)}(X^*, \bar{Y}^*)\|_F \leq \sup_{u \in [0,1]} \left\| \underbrace{\nabla_Y [\nabla_X \mathcal{L}(X^*, \bar{Y}^* + tu)]}_{B(u)} \right\|_{\text{op}} \|\bar{Y}^* - \bar{Y}\|_F.$$

Observe we can split the matrix  $B(u)$  into two matrices, one sparse matrix and one dense matrix. That is,  $B(u) = B_1(u) + B_2(u)$ , all matrices of dimension  $(nk) \times (pk)$ . The  $(i, j)$ th  $k \times k$  block of  $B_1(u)$  has entries equal to

$$\frac{1}{p} \left( - \frac{1}{(X_i^*)^\top (\bar{Y}_j^* + u(\bar{Y}_j - \bar{Y}_j^*))} - \tau^2 \mathbb{E}[A_{ij}] + \tau^2 \mathbb{E}[A_{ij}^2] ((X_i^*)^\top (\bar{Y}_j^* + u(\bar{Y}_j - \bar{Y}_j^*))) \right) I_k,$$

while the  $(i, j)$ th  $k \times k$  block of  $B_2(u)$  has entries equal to

$$\frac{1}{p} \left( \frac{1}{((X_i^*)^\top (\bar{Y}_j^* + u(\bar{Y}_j - \bar{Y}_j^*)))^2} + \tau^2 \mathbb{E}[A_{ij}^2] \right) X_i^* (\bar{Y}_j^* + u(\bar{Y}_j - \bar{Y}_j^*))^\top. \quad (\text{I.26})$$

We first analyze  $B_1(u)$ . Plugging in (I.24), we obtain

$$\frac{1}{p} \underbrace{\left( - \frac{1}{(X_i^*)^\top (\bar{Y}_j^* + u(\bar{Y}_j - \bar{Y}_j^*))} - \frac{\tau^2}{(X_i^*)^\top \bar{Y}_j^*} + (\tau^2 + 1) \cdot \frac{(X_i^*)^\top (\bar{Y}_j^* + u(\bar{Y}_j - \bar{Y}_j^*))}{((X_i^*)^\top \bar{Y}_j^*)^2} \right)}_{C_1} I_k,$$

To analyze  $C_1$  in the perspective of Assumption E.2, we analyze

$$\max_t \left| - \frac{r}{(1+t)} - \tau^2 \cdot r + (\tau^2 + 1)(1+t) \cdot r \right| : t \in \left[ - \frac{1}{\log(\min\{n, p\})}, \frac{1}{\log(\min\{n, p\})} \right].$$

By analyzing the Lagrangian of the above display, we see that for  $\min\{n, p\}$  large enough,

$$C_1 \leq c \cdot \frac{\tau^2 r}{\log(\min\{n, p\})}$$

for some universal constant  $c$ . Hence, we derive that

$$B_1(u) \preceq c\tau^2 r \cdot \frac{1}{p \cdot \log(\min\{n, p\})} \cdot I_k \otimes \mathbf{1}_{n \times p}.$$

Therefore,

$$\|B_1(u)\|_{\text{op}} \leq c\tau^2 r \cdot \frac{(np)^{1/2}}{p \cdot \log(\min\{n, p\})}. \quad (\text{I.27})$$

We next analyze  $B_2(u)$ . Plugging in (I.24), we obtain

$$\frac{1}{p} \left( \underbrace{\frac{1}{((X_i^*)^\top (\bar{Y}_j + u(\bar{Y}_j - \bar{Y}_j^*)))^2}}_{C_2} + \frac{\tau^2 + 1}{((X_i^*)^\top \bar{Y}_j^*)^2} \right) X_i^* (\bar{Y}_j + u(\bar{Y}_j - \bar{Y}_j^*))^\top.$$

By analyzing  $C_2$  similar to how we analyzed  $C_1$ , we derive that for  $\min\{n, p\}$  large enough,

$$B_2(u) \preceq (\tau^2 + 2)r^2 \cdot \frac{1}{p} \cdot X^* \otimes (\bar{Y}^* + u(\bar{Y} - \bar{Y}^*))^\top.$$

Therefore,

$$\|B_2(u)\|_{\text{op}} \leq (\tau^2 + 2)r^2 \cdot \frac{2\tilde{d}_1}{p}. \quad (\text{I.28})$$

We conclude by combining (I.27) and (I.28) by a triangle inequality.  $\square$

**Proof of Lemma E.3** Prior to proving Lemma E.3, we need the following set of concentration bounds. Recall that  $\kappa$  upper-bounds the variance of  $A_{ij}^2$  under the curved Gaussian model (see (I.25)).

**Lemma I.2.** *Conditioned on  $\Theta$ , let  $A_{ij}$  follow the distribution (4.7). Let  $\kappa = (2(1 + 2\tau^2))r^4/\tau^4$ . Under Assumption E.1, for a fixed  $i \in \{1, \dots, n\}$ , for some universal constant  $c$ , for fixed matrices  $X$  and  $\bar{Y}$ , each of the following four inequalities hold separately.*

$$\mathbb{P}\left(\left|\frac{1}{p} \sum_{j=1}^p -\tau^2 X_i^\top \bar{Y}_j (A_{ij} - \mathbb{E}[A_{ij}])\right| \geq t\right) \leq e \exp\left[\frac{-ct^2}{\kappa^2 \tau^4 r^2} \cdot p\right], \quad (\text{I.29})$$

$$\mathbb{P}\left(\left\|\frac{1}{p} \sum_{j=1}^p -\tau^2 \bar{Y}_j (A_{ij} - \mathbb{E}[A_{ij}])\right\|_2 \geq t\right) \leq ke \exp\left[\frac{-ct^2}{\kappa^2 \tau^4} \cdot \frac{p}{k^2}\right], \quad (\text{I.30})$$

$$\mathbb{P}\left(\left|\frac{1}{p} \sum_{j=1}^p \frac{\tau^2 (X_i^\top \bar{Y}_j)^2}{2} (A_{ij}^2 - \mathbb{E}[A_{ij}^2])\right| \geq t\right) \leq 2 \exp\left[-c \min\left(\frac{4t^2}{\kappa^2 \tau^4 r^4}, \frac{2t}{\kappa \tau^2 r^2}\right) \cdot p\right], \quad (\text{I.31})$$

$$\mathbb{P}\left(\left\|\frac{1}{p} \sum_{j=1}^p (\tau^2 (X_i^\top \bar{Y}_j) (A_{ij}^2 - \mathbb{E}[A_{ij}^2])) \bar{Y}_j\right\|_2 \geq t\right) \leq 2k \exp\left[-c \min\left(\frac{t^2}{\kappa^2 \tau^4 r^2}, \frac{t}{\kappa \tau^2 r}\right) \cdot \frac{p}{k^2}\right]. \quad (\text{I.32})$$

*Proof.* (I.29) and (I.31) are direct applications of Hoeffding's and Bernstein's inequality (Vershynin (2010), Proposition 5.10 and 5.16) since  $X_i^\top \bar{Y}_j \leq r$ . To show (I.30), observe that for a particular coordinate  $\ell = 1, \dots, k$ , we can apply Hoeffding's inequality to show

$$\mathbb{P}\left(\left|\frac{1}{p} \sum_{j=1}^p -\tau^2 \bar{Y}_{j\ell} (A_{ij} - \mathbb{E}[A_{ij}])\right| \geq t\right) \leq e \exp\left(\frac{-ct^2}{\kappa^2 \tau^4} \cdot p\right),$$

since the  $\ell_2$ -norm of any column of  $\bar{Y}$  is  $\sqrt{p}$ . Applying a maximal inequality and then summing over all  $k$  dimensions, we have

$$\mathbb{P}\left(\left\|\frac{1}{p} \sum_{j=1}^p -\tau^2 \bar{Y}_j (A_{ij} - \mathbb{E}[A_{ij}])\right\|_1 \geq t\right) \leq ke \exp\left(\frac{-ct^2}{\kappa^2 \tau^4} \cdot \frac{p}{k^2}\right),$$



which upper-bounds the RHS of (I.30). The same technique using Bernstein's inequality can be used to show (I.32).  $\square$

With these concentration statements, we are ready to proceed with the analysis of  $\varepsilon_{\text{fixed}}(n, p)$  row-wise.

**Lemma I.3.** *Assume the conditions in Lemma E.1. Let*

$$\widehat{\Lambda} = \operatorname{argmin}_{\Lambda \in \mathbb{R}^k} \frac{1}{p} \sum_{j=1}^p \left[ -\log(\Lambda^\top \bar{Y}_j) - \left[ \begin{array}{c} \tau^2 A_{ij} \\ -\tau^2 A_{ij}^2/2 \end{array} \right]^\top \left[ \begin{array}{c} \Lambda^\top \bar{Y}_j \\ (\Lambda^\top \bar{Y}_j)^2 \end{array} \right] \right], \quad (\text{I.33})$$

and

$$\Lambda^* = \operatorname{argmin}_{\Lambda \in \mathbb{R}^k} \frac{1}{p} \sum_{j=1}^p \left[ -\log(\Lambda^\top \bar{Y}_j) - \left[ \begin{array}{c} \tau^2 \mathbb{E}[A_{ij}] \\ -\tau^2 \mathbb{E}[A_{ij}^2/2] \end{array} \right]^\top \left[ \begin{array}{c} \Lambda^\top \bar{Y}_j \\ (\Lambda^\top \bar{Y}_j)^2 \end{array} \right] \right]. \quad (\text{I.34})$$

Then, conditioned on  $X$  and  $Y$ , with probability at least  $1 - 6/(np)$ ,

$$\|\Lambda^* - \widehat{\Lambda}\|_2 \leq c \left( \frac{\log(np)}{p} \right)^{1/4},$$

where  $c$  is a constant that depends only on  $k, \mu, L, \tau$  and  $r$ .

*Proof. Step 1:* (Setup). Our proof is inspired by Lemma 4 of Candès et al. (2020). The proof is fairly straightforward and relies mainly on the convexity properties of  $\mathcal{L}$  and the concentration statements established in Lemma I.2.

Throughout this proof, we let  $c_1, c_2, \dots$  denote constants depend on quantities that we will explicitly state, but their explicit form can change from line to line. Let  $\mathcal{L}_n(\cdot)$  and  $\mathcal{L}(\cdot)$  denote functions being minimized in (I.33) and (I.34) respectively throughout this proof only. Since  $\mathcal{L}$  is  $\mu$ -strongly convex, we have

$$\mathcal{L}(\Lambda) \geq \mathcal{L}(\Lambda^*) + \frac{1}{2\mu} \|\Lambda^* - \Lambda\|_2^2.$$

Define  $\|\Lambda^* - \Lambda\|_2 = c_1 v$  and  $h = v^2/(6c_1^2\mu)$  for some constant  $c_1$  that depends on quantity defined later in the proof. Using these definitions, the above display equals

$$\mathcal{L}(\Lambda) \geq \mathcal{L}(\Lambda^*) + 3h. \quad (\text{I.35})$$

Next, consider the sphere centered at  $\Lambda^*$  with radius  $v$ ,  $\mathcal{B}_v(\Lambda^*) = \{\Lambda : \|\Lambda^* - \Lambda\|_2 = v\}$ . Consider the event  $E$  defined as

$$\inf_{\Lambda \in \mathcal{B}(\Lambda^*)} \mathcal{L}_n(\Lambda) > \mathcal{L}(\Lambda^*) + h \quad \text{and} \quad \mathcal{L}_n(\Lambda^*) < \mathcal{L}(\Lambda^*) + h. \quad (\text{I.36})$$

Note that when event  $E$  occurs, by convexity of  $\mathcal{L}_n$ ,  $\widehat{\Lambda}$  (the minimizer of  $\mathcal{L}_n$ ) must lie within the sphere  $\mathcal{B}(\Lambda^*)$ , so hence,  $\|\widehat{\Lambda} - \Lambda^*\|_2 \leq v$ .

**Step 2:** (Decomposition of  $E$ ). Let  $\Lambda_1, \dots, \Lambda_M$  be  $M$  points on  $\mathcal{B}(\Lambda^*)$ , to be specified in the next step. We decompose  $E$  in the above display into three separate events that we will control individually. Let the event  $E_1$  be defined as

$$\max_{i \in \{1, \dots, M\}} |\mathcal{L}(\Lambda_i) - \mathcal{L}_n(\Lambda_i)| \leq h. \quad (\text{I.37})$$

and the event  $E_2$  be defined as

$$\inf_{\Lambda \in \mathcal{B}_v(\Lambda^*)} \mathcal{L}_n(\Lambda) \geq \mathcal{L}_n(\Lambda_i) - h. \quad (\text{I.38})$$

Starting from the event  $E_1$ , we have the following line of implications

$$\begin{aligned} \max_{i \in \{1, \dots, M\}} |\mathcal{L}(\Lambda_i) - \mathcal{L}_n(\Lambda_i)| < h &\implies \forall i \in \{1, \dots, M\}, \quad \mathcal{L}_n(\Lambda_i) \geq \mathcal{L}(\Lambda_i) - h \\ &\implies \forall i \in \{1, \dots, M\}, \quad \mathcal{L}_n(\Lambda_i) - h > \mathcal{L}(\Lambda^*) + h, \\ &\implies \inf_{\Lambda \in \mathcal{B}_v(\Lambda^*)} \mathcal{L}_n(\Lambda) > \mathcal{L}(\Lambda^*) + h, \end{aligned}$$

where the second implication follows from the definition of  $h$  in (I.35), and the last implication holds under the event  $E_2$ . Hence, together  $E_1$  and  $E_2$  would imply the first part of  $E$  in (I.36).

Observe that the following event  $E_3$ , defined as,

$$|\mathcal{L}(\Lambda^*) - \mathcal{L}_n(\Lambda^*)| \leq h \tag{I.39}$$

would imply the second part of  $E$  in (I.36). Hence, since the event  $E_1 \cap E_2 \cap E_3$  implies event  $E$ , we have by union bound

$$\mathbb{P}(E^c) \leq \mathbb{P}(E_1^c) + \mathbb{P}(E_2^c) + \mathbb{P}(E_3^c).$$

Therefore, the remainder of the proof shows that for a rate of  $v$ , the summation of the probability of the complementary events is small.

**Step 3:** (Analysis of  $E_1$ ). We can construct an  $\gamma$ -net of the sphere  $\mathcal{B}(\Lambda^*)$ . Specifically, by Lemma 9.5 of Ledoux and Talagrand (2013), we require at most most  $M = (3/\gamma)^k$  matrices, denoted as  $\Lambda_1, \dots, \Lambda_M$ , that satisfies,

$$\text{for any } \Lambda \in \mathcal{B}(\Lambda^*), \quad \min_{i \in \{1, \dots, M\}} \|\Lambda - \Lambda_i\|_2 \leq \gamma.$$

For the rest of the proof, we will set

$$\gamma = (\log(np)/p)^{1/4}. \tag{I.40}$$

We analyze  $E_1$  in (I.37), the event that  $\mathcal{L}_n(\Lambda_i)$  and  $\mathcal{L}(\Lambda_i)$  are close along the  $\gamma$ -net. Using a union bound along with concentration bounds shown in Lemma I.2, for  $p$  large enough, we have with probability at least  $1 - 2/(np)$  that event  $E_1$  occurs,

$$\max_{i \in \{1, \dots, M\}} |\mathcal{L}(\Lambda_i) - \mathcal{L}_n(\Lambda_i)| \leq c_2 \left( \frac{\log(np) + k \log(3/\gamma)}{p} \right)^{1/2},$$

where  $c_2$  depends on  $\tau$  and  $r$ . Observe that with our choice of  $\gamma$  in (I.40),  $\log(np)$  dominates  $k \log(3/\gamma)$  for  $n$  large enough. Hence, treating  $k$  as a constant, to set the left-hand side of the above display to be equal to  $h$ , we need

$$v = c_3 (\log(np)/p)^{1/4}, \tag{I.41}$$

where  $c_3$  depends on  $k, \mu, \tau$ , and  $r$ .

**Step 4:** (Analysis of  $E_2$ ). Next, we analyze  $E_2$  in (I.38). By convexity, letting  $\Lambda_i$  be the closest vector to  $\Lambda$  in  $\ell_2$  distance,

$$\mathcal{L}_n(\Lambda) \geq \mathcal{L}_n(\Lambda_i) + \langle \nabla \mathcal{L}_n(\Lambda_i), \Lambda - \Lambda_i \rangle \geq \mathcal{L}_n(\Lambda_i) - \|\nabla \mathcal{L}_n(\Lambda_i)\|_2 \|\Lambda - \Lambda_i\|_2. \tag{I.42}$$

We know  $\|\Lambda - \Lambda_i\|_2 \leq \gamma$  by construction. To bound  $\|\nabla \mathcal{L}_n(\Lambda_i)\|_2$ , using a union bound and Lemma I.2 once again, we have with probability at least  $1 - 2/(np)$  for  $p$  large enough, event  $E_2'$  occurs,

$$\max_{i \in \{1, \dots, M\}} \|\nabla \mathcal{L}(\Lambda_i) - \nabla \mathcal{L}_n(\Lambda_i)\|_2 \leq c_2 k \left( \frac{\log(np) + k \log(3/\gamma)}{p} \right)^{1/2}. \tag{I.43}$$

Furthermore, we know that since  $\nabla\mathcal{L}(\Lambda^*) = 0$  and the gradient of  $\mathcal{L}$  is  $L$ -Lipschitz smooth, using the definition of  $v$  in (I.41), we have

$$\|\nabla\mathcal{L}(\Lambda_i) - \nabla\mathcal{L}(\Lambda^*)\|_2 \leq L\|\Lambda_i - \Lambda^*\|_2 = Lv = Lc_3\left(\frac{\log(np)}{p}\right)^{1/4}. \quad (\text{I.44})$$

Therefore, combining (I.43) and (I.44), that on event  $E'_2$ ,

$$\max_{i \in \{1, \dots, M\}} \|\nabla\mathcal{L}_n(\Lambda_i)\|_2 \|\Lambda_i - \widehat{\Lambda}\|_2 \leq \left(c_2\left(\frac{\log(np)}{p}\right)^{1/2} + Lc_3\left(\frac{\log(np)}{p}\right)^{1/4}\right) \cdot \gamma.$$

Recalling the rate of  $\gamma$  set in (I.40) and plugging the above inequality back into (I.42), we have on event  $E_2$ ,

$$\mathcal{L}_n(\Lambda) \geq \mathcal{L}_n(\Lambda_i) - c_4\left(\frac{\log(np)}{p}\right)^{1/2} = \mathcal{L}_n(\Lambda_i) - h, \quad (\text{I.45})$$

where  $c_4$  depends on  $L, k, \mu, \tau$ , and  $r$ . This shows  $E_2$ .

**Step 5:** (Analysis of  $E_3$  and conclusion). From similar calculations above based on Lemma I.2, we can show that for  $p$  large enough, we have with probability at least  $1 - 2/(np)$  that event  $E_3$  occurs,

$$|\mathcal{L}(\Lambda^*) - \mathcal{L}_n(\Lambda^*)| \leq h.$$

Hence, we conclude the proof, where  $c_1$  initially stated is a constant that depends on  $L, k, \mu, \tau$ , and  $r$ . By Lemma E.1, we know  $L$  and  $\mu$  are constants that depend on only  $\tau$  and  $r$ . □

We are now ready to prove Lemma E.3.

*Proof of Lemma E.3.* Let  $\widehat{X} = M_n(\bar{Y})$  and  $X^* = M(\bar{Y})$ . By Lemma I.3, we have with probability  $6/(np)$ , for any  $i \in \{1, \dots, n\}$ ,

$$\|X_i^* - \widehat{X}_i\|_2^2 \geq c\left(\frac{\log(np)}{p}\right)^{1/2}.$$

Hence, applying a union bound over all  $i \in \{1, \dots, n\}$ , we have with probability  $6/p$ ,

$$\max_{i \in \{1, \dots, n\}} \|X_i^* - \widehat{X}_i\|_2^2 \geq c\left(\frac{\log(np)}{p}\right)^{1/2}.$$

This event implies that,

$$\|X^* - \widehat{X}\|_F^2 \geq cn\left(\frac{\log(np)}{p}\right)^{1/2}. \quad \square$$

*Proof of Corollary E.4.* The proof mainly follows from the proof of Proposition D.1 above, where we replace the term  $\varepsilon_{\text{unif}}(n, p)$  with  $\varepsilon_{\text{fixed}}(n, p)$ , which requires us to take a union bound over all  $T$  iterations.

Due to our invocation of Assumptions 5.1 and 5.2, we have shown already in (I.10) that  $\widetilde{d}_j$  scales with  $(np\lambda_j\gamma_j)^{1/2}$  asymptotically, completing the proof. □

## J Auxiliary results and proofs

The first result is a restatement of Proposition 2.1 of Vershynin (2012), which helps us establish the concentration of the estimated second moment matrix.

**Lemma J.1** (Probability bound of operator norm of a covariance matrix). *Let  $X_1, \dots, X_n$  be i.i.d.  $k$ -dimensional random variables that satisfies Assumption 5.1 with a fixed constant  $D$  (also known as the sub-Gaussian norm). Then, for any  $\delta$ ,*

$$\mathbb{P}\left(\left\|\mathbb{E}[X_i X_i^\top] - \frac{1}{n} \sum_{i=1}^n X_i X_i^\top\right\|_{\text{op}} \geq C\left(\sqrt{\frac{k}{n}}\right)\right) \leq 1 - \delta,$$

for  $n$  sufficiently large enough, and where  $C$  is a constant that depends only  $D$  and  $\delta$ .

The following perturbation result can be derived using the perturbation results stated in Theorem 2 of Yu et al. (2014) with Hermitian dilation (see, e.g. Tropp (2012)) along with related notions of distance among orthonormal matrices stated in Lemma 1 of Cai et al. (2018), which helps us establish the concentration of the left singular vectors given concentration of the second moment matrices.

**Lemma J.2** (Davis-Kahan theorem for singular vectors). *For any two  $n \times p$  symmetric matrices  $\Theta$  and  $\hat{\Theta}$  where  $m = \min\{n, p\}$ , let  $\sigma_1 > \dots > \sigma_m$  and  $\hat{\sigma}_1 > \dots > \hat{\sigma}_m$  denote the singular values of each matrix respectively, and  $U_1, \dots, U_k$  and  $\hat{U}_1, \dots, \hat{U}_m$  denote their corresponding left singular vectors. Fix  $d \in \{1, \dots, m\}$  and let  $U = [U_1, \dots, U_d]$  and  $\hat{U} = [\hat{U}_1, \dots, \hat{U}_d]$  denote the concatenated matrix of left singular vectors. Then, letting  $\mathcal{O}_{d,d}$  denote the set of  $d \times d$  unitary matrices,*

$$\inf_{Q \in \mathcal{O}_{d,d}} \|U - \hat{U}Q\|_F \leq \frac{4 \min\left\{d^{1/2}\|\Theta - \hat{\Theta}\|_{\text{op}}, 2^{1/2}\|\Theta - \hat{\Theta}\|_F\right\}}{\sigma_d - \sigma_{d+1}},$$

where we use the convention that  $\sigma_{m+1} = 0$ .

We note that when we apply the above lemma in our proofs throughout Appendix I, the uniqueness of the eigenvalues and the spacing of the eigenvalues assumed in Assumption 5.2 absolves us from needing to worry about the above unitary matrix  $Q$ . Instead, we simply need to worry “up to sign” as alluded to Proposition 5.2, where we need to possibly multiply the columns of  $\hat{U}$  by  $\pm 1$ .

The next result on random sequences can be easily verified by direct calculation.

**Lemma J.3.** *For two positive deterministic sequences  $a_n$  and  $b_n$ , consider a random sequence  $X_n = O_P(a_n)$  and a random positive sequence  $Y_n = \Omega_P(b_n)$ . If  $a_n = o(b_n)$ , then*

$$(Y_n + X_n)^{1/2} - Y_n^{1/2} = O_P(a_n b_n^{-1/2}).$$