Scalable Amortized GPLVMs for Single Cell Transcriptomics Data

Sarah Zhao
Department of Statistics
Stanford University
Stanford, CA 94305, USA
smxzhao@stanford.edu

Aditya Ravuri
Department of Computer Science
University of Cambridge
Cambridge, United Kingdom
ar847@cam.ac.uk

Vidhi Lalchand
Eric and Wendy Schmidt Center
Broad Institute of MIT and Harvard
Cambridge, MA 02142, USA
vidrl@mit.edu

Neil D. Lawrence
Department of Computer Science
University of Cambridge
Cambridge, United Kingdom
ndl21@cam.ac.uk

Abstract

Dimensionality reduction is crucial for analyzing large-scale single-cell RNA-seq data. Gaussian Process Latent Variable Models (GPLVMs) offer an interpretable dimensionality reduction method, but current scalable models lack effectiveness in clustering cell types. We introduce an improved model, the amortized stochastic variational Bayesian GPLVM (BGPLVM), tailored for single-cell RNA-seq with specialized encoder, kernel, and likelihood designs. This model matches the performance of the leading single-cell variational inference (scVI) approach on synthetic and real-world COVID datasets and effectively incorporates cell-cycle and batch information to reveal more interpretable latent structures as we demonstrate on an innate immunity dataset.

1 Introduction

Single-cell transcriptomics sequencing (scRNA-seq) has enabled the study of gene expression at the individual cell level. This high-resolution analysis has helped discover new cell types and cell states, reveal developmental lineages, and identify cell type-specific gene expression profiles (Montoro et al., 2018; Plasschaert et al., 2018; Luecken & Theis, 2019). This high-level resolution, however, comes with a cost. scRNA-seq data are often extremely sparse and prone to various technical and biological noise, such as sequencing depth, batch effects, and cell-cycle phases (Svensson et al., 2018; Tanay & Regev, 2017; Luecken & Theis, 2019; Hie et al., 2020). Various dimensionality reduction techniques have been developed to leverage intrinsic structures in the data (Heimberg et al., 2016) to map to a lower-dimensional latent space. These methods help facilitate downstream tasks like clustering and visualization, while avoiding the curse of dimensionality. Our work emphasizes probabilistic dimensionality reduction methods, which, through providing explicit probabilistic models for the data, allows for more interpretable models and uncertainty measures in the learned latent space.

In particular, we study a class of latent variable models known as Gaussian Process Latent Variable Models (GPLVMs) (Lawrence, 2004), which have recently been applied to scRNA-seq data (Campbell & Yau, 2015; Buettner et al., 2015; Ahmed et al., 2019; Verma & Engelhardt, 2020; Lalchand et al., 2022a). These models, which use Gaussian processes (GPs) to define nonlinear mappings from the latent space to data space, can incorporate prior information in the GP kernel function, motivating its use in single-cell transcriptomics data to model known or approximated covariate random effects, such as batch IDs and cell cycle phases. This approach is made scalable via mini-batching; however, the resulting Bayesian GPLVM model (BGPLVM) struggles to learn informative latent spaces for certain datasets (Lalchand et al., 2022a).

In this work, we present an amortized BGPLVM better fit to scRNA-seq data by leveraging design choices made in a leading probabilistic dimensionality reduction method called single cell variational inference (scVI) (Lopez et al., 2018). While scVI has seen impressive performance in a variety of downstream tasks, it does not easily allow for interpretable incorporation of prior domain knowledge.

In Sections 2 and 3, we describe this model, providing a concise background on BGPLVMs and highlighting the model modifications. Section 4 then discusses (1) an ablation study demonstrating each components contribution to the model's performance via a synthetic dataset; (2) comparable performance to scVI for both the synthetic dataset and a real-world COVID-19 dataset (Stephenson et al., 2021); and (3) promising results for interpretably incorporating prior domain knowledge about cell-cycle phases in an innate immunity dataset (Kumasaka et al., 2021). Our work shines a light on key considerations in developing a scalable, interpretable, and informative probabilistic dimensionality method for scRNA-seq data.

2 Background

This section provides a concise introduction to existing BGPLVM models from the literature.

2.1 Amortized Stochastic Variational Bayesian GPLVM

Given a training set comprised of N D-dimensional observations $\mathbf{Y} = [\boldsymbol{y}_1 \dots \boldsymbol{y}_N]^T \in \mathbb{R}^{N \times D}$, we seek to represent our data with Q-dimensional embeddings $\mathbf{X} = [\boldsymbol{x}_1 \dots \boldsymbol{x}_N]^T \in \mathbb{R}^{N \times Q}$ which are latent and stochastic and $Q \ll D$ provides the dimensionality reduction. The probabilistic model describing the data can be written as follows:

$$\text{Latent prior: } p(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{0}, \mathbf{I}_Q) \quad \text{GP Prior: } p(\mathbf{F}|\mathbf{X}, \theta) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{f}_d|\mathbf{0}, \mathbf{K}_{NN}), \tag{1}$$

$$\text{Likelihood: } p(\mathbf{Y}|\mathbf{F}, \sigma_y^2) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{y}_d|\mathbf{f}_d, \sigma_y^2 \mathbf{I}_N) = \prod_{n=1}^{N}\prod_{d=1}^{D} \mathcal{N}(y_{nd}|f_d(\mathbf{x}_n), \sigma_y^2), \tag{2}$$

$\mathbf{F} \equiv \{\boldsymbol{f}_d\}_{d=1}^{D}$ denotes the collection of latent functions where $\boldsymbol{f}_d$ is associated with $\boldsymbol{y}_d$ (the d^{th} column of $\mathbf{Y}$). $\mathbf{K}_{NN}$ is the covariance matrix corresponding to a user chosen positive-definite kernel function $k(\mathbf{x}, \mathbf{x}')$ evaluated on latent points $\{\boldsymbol{x}_n\}_{n=1}^{N}$ and parameterized by hyperparameters $\boldsymbol{\theta}$. The kernel hyperparameters are shared across all dimensions D.

Moreover, to speed up computation and allow for mini-batching, we use inducing variables $\mathbf{U} = \{\mathbf{u}_m \in \mathbb{R}^Q\}_{m=1}^{M}$ also distributed with a GP prior $\mathbf{u}_d|\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{MM})$, where $\mathbf{K}_{MM}$ is the kernel evaluated at inducing locations $\mathbf{Z} \in \mathbb{R}^{M \times Q}$ as in Hensman et al. (2013); Lalchand et al. (2022b). The introduction of inducing variables gives us the following sparse GP prior:

$$p(\mathbf{F}|\mathbf{U}, \mathbf{X}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{f}_d|\mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{u}_d, \mathbf{K}_{NN} - \mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{MN}). \tag{3}$$

The joint posterior over all unknowns $p(\mathbf{F}, \mathbf{U}, \mathbf{X}|\mathbf{Y})$ is intractable, but admits a tractable lower bound to the marginal likelihood $p(\mathbf{Y}|\boldsymbol{\theta})$ under the variational formulation,

$$q(\mathbf{F}, \mathbf{X}, \mathbf{U}) = \Big[\prod_{d=1}^{D} p(\boldsymbol{f}_d|\boldsymbol{u}_d, \mathbf{X})q(\boldsymbol{u}_d)\Big]q(\mathbf{X}), \tag{4}$$

where the variational distributions are:

$$q(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|H_{\phi_1}(\mathbf{y}_n), \text{diag}(H_{\phi_2}(\mathbf{y}_n)), \quad q(\mathbf{U}) = \prod_{d=1}^{D} q(\mathbf{u}_d) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{u}_d|\mathbf{m}_d, \mathbf{S}_d), \tag{5}$$

where $\{\mathbf{m}_d, \mathbf{S}_d\}_{d=1}^{D}$ denotes the variational parameters. The mean and variance of the variational Gaussian distributions are parameterized as outputs of individual neural networks H_{ϕ_1} and H_{ϕ_2}, which act as encoders. The network weights are amortized and shared across all the data points enabling extension to very large scale datasets as the amortized model side-steps the need to learn the latent variational means and covariances per data point, i.e. $q(\mathbf{x}_n)$, and once trained, allows for constant-time $\mathcal{O}(N)$ inference. The resulting stochastic variational lower bound (Lalchand et al., 2022b), which factorizes across N and D, permitting mini-batching, is given by:

$$\begin{aligned}\mathcal{L}(q(\cdot)) = \sum_{n,d} \mathbb{E}_{q(\boldsymbol{x}_n)} \mathbb{E}_{p(\boldsymbol{f}_d|\boldsymbol{u}_d,\boldsymbol{x}_n)q(\boldsymbol{u}_d)} \left[\log \mathcal{N}(y_{nd}|f_d(\mathbf{x}_n), \sigma_y^2)\right] &- \sum_{n=1}^{N} \mathrm{KL}\left(q(\mathbf{x}_n)||p(\mathbf{x}_n)\right) \\ &- \sum_{d=1}^{D} \mathrm{KL}\left(q(\mathbf{u}_d)||p(\mathbf{u}_d|\mathbf{Z})\right).\end{aligned} \tag{6}$$

2.2 Encoding Domain Knowledge through Kernels

A key benefit of using GPLVMs is that we can encode prior information into the generative model, especially through the kernel design, allowing for more interpretable latent spaces and less training data. Here, we highlight kernels tailored to scRNA-seq data that correct for batch and cell-cycle nuisance factors as introduced by Lalchand et al. (2022a).

Batch correction kernel formulation In order to correct for confounding batch effects through the GP formulation, Lalchand et al. (2022a) proposed the following kernel structure with an additive linear kernel term to capture random effects:

$$\tilde{\mathbf{f}}_d \sim \mathcal{N}(\mu_f \mathbf{I}_N + \Phi\zeta_d, \underbrace{\mathbf{K}_{NN} + \nu\Phi\Phi^T}_{\tilde{\mathbf{K}}_{NN}}), \tag{7}$$

which implicitly represents the relation $\mathbf{Y} = \mathbf{F} + \Phi\mathbf{B} + \varepsilon$ where $\Phi \in \mathbb{R}^{N \times D_{\text{covar}}}$ is the design matrix where each row represents the known covariates for each cell; $\mathbf{B} \in \mathbb{R}^{D_{\text{covar}} \times D_{\text{gene}}}$ is a random variable $[\dots, B_d, \dots]$ (B_d denotes a column) representing the random effect of each known covariate on gene expression $B_d \sim \mathcal{N}(\zeta_d, \nu \mathbf{I}_{D_{covar}})$, $\zeta_d \in \mathbb{R}^{D_{covar}}$, $\nu \in \mathbb{R}$, $\varepsilon \in \mathbb{R}^{N \times D_{\text{gene}}}$ represents the noise model and $\mu_f \in \mathbb{R}$ is a constant mean for the latent functions. For most of this work, we use an SE-ARD kernel with additive linear kernel, henceforth denoted as SE-ARD+Linear.

Cell-cycle phase kernel When certain genes strongly reflect cell-cycle phase effects, obscuring key biological factors, a kernel designed to explicitly address a cell-cycle latent variable can effectively mitigate these effects. This motivates the use of adding a periodic kernel to the above kernel formulation. In particular, we specify the first latent dimension as a proxy for cell-cycle information and model our kernel as:

$$\begin{aligned} k_{\tilde{f}}(\boldsymbol{x}, \boldsymbol{x'}) &= \sigma_f^2 \exp\left(\frac{-2\sin^2(|\boldsymbol{x_1} - \boldsymbol{x'_1}|/2)}{l_1^2}\right) \times \exp\left(-\sum_{q=1}^{Q} \frac{(\boldsymbol{x_q} - \boldsymbol{x'_q})^2}{2l_q^2}\right) + \nu\Phi\Phi^T \quad (8) \\ &= k_{\text{per}} \times k_{\text{se-ard}} + k_{\text{lin}}, \quad (9) \end{aligned}$$

where $\boldsymbol{\theta} = \left\{\sigma_f^2, \{l\}_{q=1}^{Q}, \nu, \mu_f, \zeta_d\right\}$ are the hyperparameters of the BGPLVM. In particular, the periodic kernel helps capture the effects of the cell-cycle phases. We will refer to this kernel as PerSE-ARD+Linear, which will be used in our study of the innate immunity dataset discussed in Section 4.

3 Our model

In the sections below, we discuss a set of modifications to the baseline model presented above, which form the main contributions of this work. In particular, we show that row (library) normalizing the data, using an appropriate likelihood, incorporating batch and cell-cycle information via SE-ARD+Linear and PerSE-ARD+Linear (Section 2.2) and implementing a modified encoder significantly improves the BGPLVM's performance. We present the schematic of the modified BGPLVM in Figure 1.

3.1 Pre-Processing and Likelihood

Raw scRNA-seq data are discrete and must be pre-processed to better align with the Gaussian likelihood in the probabilistic model of the baseline discussed above, which we call OBGPLVM (short for Original Bayesian GPLVM). However, the assumption that this pre-processed data are normally distributed is not necessarily justified. Instead of adjusting the data to fit our model, we aim to better adapt our likelihood to the data. In particular, we only normalize the total counts per cell (i.e. library size) to account for technical factors (Lun et al., 2016) and adopt a negative binomial likelihood like that in scVI (detailed in Appendix A.1).

In particular, we use a negative binomial with fixed scaling factor $\ell = 5000$ and $r = 10^6$. This is the simplest likelihood function and approximates a Poisson distribution. To account for sequencing depth differences, the likelihood requires the raw count data to be library size normalized to 5000 first. We call this likelihood *ApproxPoisson*. [1]

$$y_{nd} \sim \text{Negative Binomial}\left(5000 \times \text{softmax}(\tilde{f}_d(x_n)), 10^6\right). \tag{10}$$

In our initial experiments, we found that the more complex the likelihood function was (in terms of parameters to be learned), the worse the resulting BGPLVM-learned latent space was. While one may expect the more complex and expressive likelihoods to perform better, this opposite trend may be because the model is non-identifiable. That is, especially since the loss function does not explicitly optimize for latent space representations, the extra parameters may overfit and cause the model to fail to learn important biological signals. One such ablation study is presented in Appendix B.3.2. Due to this observation, we focus on the simplest (and best performing) negative binomial-based likelihood, *ApproxPoisson*.

3.2 Encoder

In the encoder analysis, we compare a simple encoder comprised of linear layers followed by softplus activations (Simple NN) with the scVI's more complex encoder (scVI NN). scVI NN incorporates batch information as input to the nonlinear mapping, so incorporating this encoder into the BGPLVM may help address batch effects observed in the raw count data. Additionally, the scVI encoder architecture includes batch normalizations, contributing to a more stable optimization process, which we leverage for our GPLVM implementation.

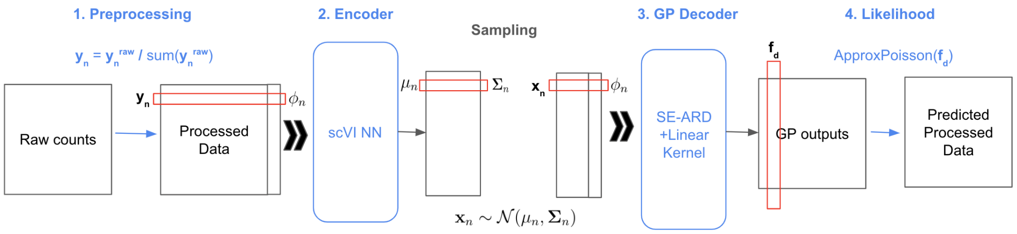


Figure 1: Overview of Modified BGPLVM Model

4 Results and Discussion

We present results for three experiments on an simulated dataset and two real-world datasets, which are detailed in Appendix B.1. Full experiment details and results with latent space metrics are also presented in Appendix B and D.

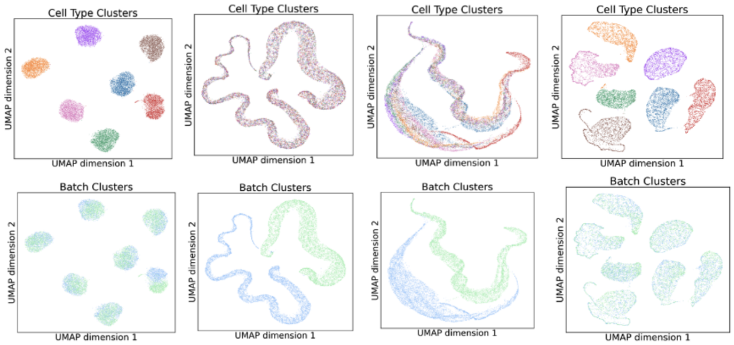
[1] The scVI negative binomial distribution's parameterization is equivalent to the generative model, $y|w \sim \text{Poisson}(w)$ with $w \sim \Gamma(\theta, \theta/\mu)$. Note that $w \overset{\theta \to \infty}{\sim} \mathcal{N}(\mu, \mu/\theta) \to \delta(\mu)$, and thus $y \overset{\text{approx.}}{\sim} \text{Poisson}(\mu)$.

4.1 Each component is crucial to modified model performance

To better understand how each component affects our model performance, we conducted an ablation study with a synthetic scRNA-seq dataset distributed according to a true negative binomial likelihood simulated by Splatter (Zappia et al., 2017). In particular, we reverted each component to a more standard BGPLVM component to evaluate its importance to the model's overall performance. The results for this experiment are detailed in Figure 2 for the simulated dataset. Changing the preprocessing step and likelihood to match a Gaussian distribution as is done in standard GPLVMs completely removes any perceivable cell type separation and results in separated batches (Fig. 2(b)). These observations support our hypothesis that the likelihoods were misaligned with the underlying distribution, at least for the simulated single-cell dataset.

If the SE-ARD+Linear kernel is changed to a fully linear kernel (detailed in Appendix B.3.1), the batches separate while the cell-types begin to mix but are still slightly differentiable, albeit within the separated batches (Fig. 2(c)). These changes may be attributed to the fact that linear kernel is not expressive enough to capture the cell-type information while the nonlinearity of the SE-ARD+Linear kernel permits extra flexibility.

In this reverse ablation study, the encoder exhibits the least impact on the latent space representation, as evidenced by the clear separation of cell types and well-mixed batches in Fig. 2(d). This behavior can be attributed to the encoder playing a smaller role in defining the generative model as it primarily functions as a means of regularization for mappings from the data space to the latent space.



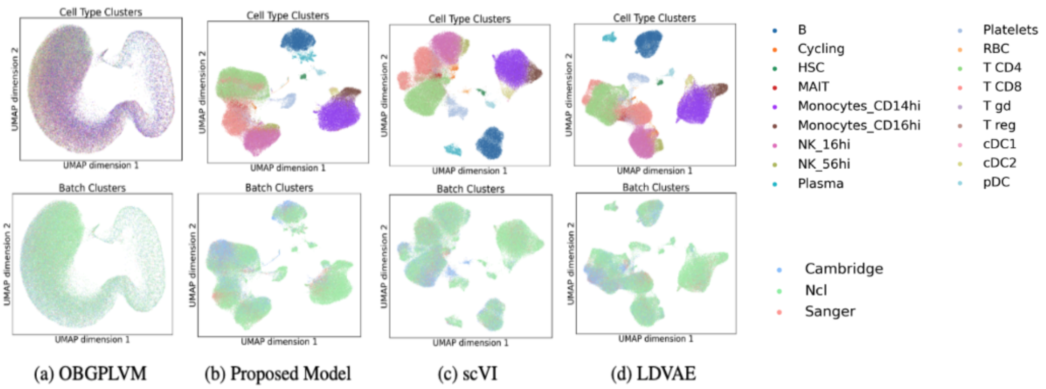
(a) Proposed Model (b) Gaussian Likelihood (c) Linear Kernel (d) Simple NN Encoder

Figure 2: Ablation study with the simulated dataset on the proposed BGPLVM model where we change one component at a time (labeled in subfigures) and visualize the resulting UMAPs. The top row is colored by cell-type and the bottom row by batch.

4.2 Modified model achieves significant improvements over standard Bayesian GPLVM and is comparable to scVI

We compare our proposed model with three benchmark models: OBGPLVM, the current state-of-the-art scVI (Lopez et al., 2018) (Appendix A.1), and a simplified scVI model with a linear decoder (LDVAE) (Svensson et al., 2020) (Appendix A.2) on the synthetic dataset and a real-world COVID-19 dataset (Stephenson et al., 2021). The UMAP plots for the COVID dataset are presented in Figure 3 and the detailed latent space metrics and UMAP plots are given in Appendix D.

Based on the UMAP visualizations, we observe that for both the simulated and COVID datasets, the modified BGPLVM achieves more visually separated cell types and mixed batches compared to the standard Bayesian GPLVM. The model also achieves visually comparable visualizations to scVI and LDVAE (Figures 7 and 3). While the modified model may not achieve better performance when compared to scVI and LDVAE, the GPLVM offers a more intuitive way to encode prior domain knowledge, and exploring such kernels and likelihoods more tailored to specific datasets are left for future work.



(a) OBGPLVM (b) Proposed Model (c) scVI (d) LDVAE

Figure 3: UMAPs generated from the latent spaces of four models: an implementation of the original BGPLVM, the modified BGPLVM for scRNA-seq data, scVI, and a linear decoder scVI (LDVAE) for the COVID data set. The top row is color/shaded by cell type and the bottom by batch.

4.3 Consistency of Latent space with biological factors

An advantage of our model is the ability to incorporate biologically interpretable data to boost latent space interpretability and overall performance. In particular, we compared our learned latent space with previous expert-labelled inferences on the innate immunity dataset in Kumasaka et al. (2021). Pretraining on well-initialized latents and finetuning our model with a PerSE-ARD+Linear kernel allowed us to recover latents consistent with those inferred and biologically motivated in Kotliar et al. (2019) (Figure 4 (top row)) while also separating cells by their treatment conditions (Figure 4 (bottom row)). Moreover, as indicated by the color gradations in the right two UMAP plots in the bottom row, the model's learned latent space is able to distinguish immune response pseudotime directions. This shows how initializations can be done on the amortized BGPLVM encoder-decoder models.

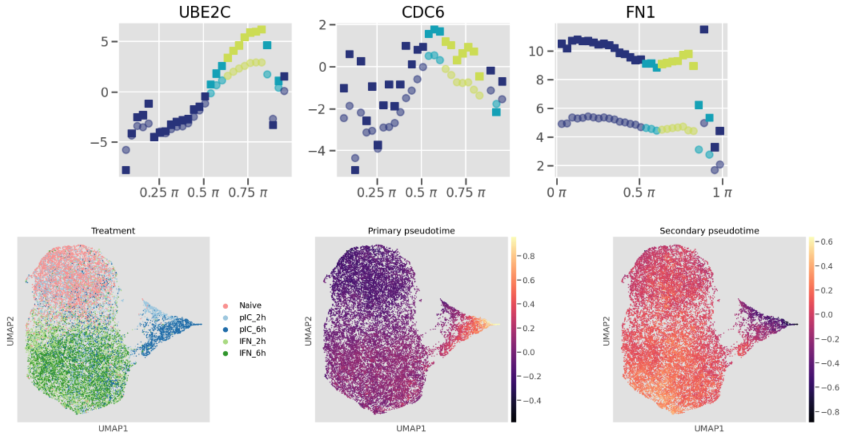


Figure 4: (Top row) Plots of log means and log variances (both parametrized by the same GP) versus learned cell-cycle pseudotime dimension for three specific genes (UBE2C, CDC6, FN1). The squares depict log variances and the circles depict log means of the library normalized data, both colored by the phases annotated in Kumasaka et al. (2021). We see that our model's learned cell-cycle phases correspond roughly to the phases labelled in Kumasaka et al. (2021). (Bottom row) UMAP plots of our model's learned latent space excluding directions identified with hidden technical effects (e.g. batch and plate border effects). Cells are colored by treatment condition (left), primary (middle) and secondary (right) pseudotime directions.

5 Conclusion

This paper identifies a misalignment in the generative model of current GPLVMs used in single-cell data and proposes an amortized BGPLVM better adapted to the scRNA-seq dimensionality reduction setting. In particular, by drawing insight from commonly used single-cell-specific methods, including scVI, LDVAE, and Splatter single-cell simulations, our proposed model tackles three main aspects of single-cell data by (1) accounting for count data with an approximate Poisson likelihood, (2) incorporating batch effect modelling in both the encoder and GP kernel, and (3) normalizing the library size in the data via a pre-processing step. We demonstrate the importance of aligning modelling choices to domain-specific knowledge as the model achieves comparable performance to scVI on both a simulated dataset and real-world COVID dataset in both UMAP visualizations and commonly used latent space metrics.

Acknowledgments

The authors would like to thank Emma Dann, Natsuhiko Kumasaka and the rest of the team at Sanger for help and guidance with our initial project and for providing the data and code, which we based this study on. AR is supported by the accelerate programme for scientific discovery. During the time of this work, SZ was supported by the Churchill Scholarship.

References

Sumon Ahmed, Magnus Rattray, and Alexis Boukouvalas. Grandprix: scaling up the bayesian gplvm for single-cell data. *Bioinformatics*, 35(1):47–54, 2019.

Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, 33(2):155–160, 2015.

Kieran Campbell and Christopher Yau. Bayesian gaussian process latent variable models for pseudotime inference in single-cell rna-seq data. *bioRxiv*, pp. 026872, 2015.

Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M Ibrahim, Andrew J Hill, Fan Zhang, Stefan Mundlos, Lena Christiansen, Frank J Steemers, et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502, 2019.

Graham Heimberg, Rajat Bhatnagar, Hana El-Samad, and Matt Thomson. Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell systems*, 2(4):239–250, 2016.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Brian Hie, Joshua Peters, Sarah K Nyquist, Alex K Shalek, Bonnie Berger, and Bryan D Bryson. Computational methods for single-cell rna sequencing. *Annual Review of Biomedical Data Science*, 3:339–364, 2020.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

Dylan Kotliar, Adrian Veres, M Aurel Nagy, Shervin Tabrizi, Eran Hodis, Douglas A Melton, and Pardis C Sabeti. Identifying gene expression programs of cell-type identity and cellular activity with single-cell rna-seq. *Elife*, 8:e43803, 2019.

Natsuhiko Kumasaka, Raghd Rostom, Ni Huang, Krzysztof Polanski, Kerstin B Meyer, Sharad Patel, Rachel Boyd, Celine Gomez, Sam N Barnett, Nikolaos I Panousis, et al. Mapping interindividual dynamics of innate immune response at single-cell resolution. *bioRxiv*, pp. 2021–09, 2021.

Vidhi Lalchand, Aditya Ravuri, Emma Dann, Natsuhiko Kumasaka, Dinithi Sumanaweera, Rik GH Lindeboom, Shaista Madad, Sarah A Teichmann, and Neil D Lawrence. Modelling technical and biological effects in scrna-seq data with scalable gplvms. *arXiv preprint arXiv:2209.06716*, 2022a.

Vidhi Lalchand, Aditya Ravuri, and Neil D Lawrence. Generalised gplvm with stochastic variational inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 7841–7864. PMLR, 2022b.

Neil D Lawrence. Gaussian process models for visualisation of high dimensional data. *Advances in Neural Information Processing Systems*, 2004.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

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

Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50, 2022.

Aaron T.L. Lun, Karsten Bach, and John C. Marioni. Pooling across cells to normalize single-cell rna sequencing data with many zero counts. *Genome Biology*, 17(75), 2016. doi: https://doi.org/10.1186/s13059-016-0947-7.

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

Daniel T Montoro, Adam L Haber, Moshe Biton, Vladimir Vinarsky, Brian Lin, Susan E Birket, Feng Yuan, Sijia Chen, Hui Min Leung, Jorge Villoria, et al. A revised airway epithelial hierarchy includes cftr-expressing ionocytes. *Nature*, 560(7718):319–324, 2018.

Lindsey W Plasschaert, Rapolas Žilionis, Rayman Choo-Wing, Virginia Savova, Judith Knehr, Guglielmo Roma, Allon M Klein, and Aron B Jaffe. A single-cell atlas of the airway epithelium reveals the cftr-rich pulmonary ionocyte. *Nature*, 560(7718):377–381, 2018.

Emily Stephenson, Gary Reynolds, Rachel A Botting, Fernando J Calero-Nieto, Michael D Morgan, Zewen Kelvin Tuong, Karsten Bach, Waradon Sungnak, Kaylee B Worlock, Masahiro Yoshida, et al. Single-cell multi-omics analysis of the immune response in covid-19. *Nature medicine*, 27 (5):904–916, 2021.

Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nature protocols*, 13(4):599–604, 2018.

Valentine Svensson, Adam Gayoso, Nir Yosef, and Lior Pachter. Interpretable factor models of single-cell rna-seq via variational autoencoders. *Bioinformatics*, 36(11):3418–3421, 2020.

Amos Tanay and Aviv Regev. Scaling single-cell genomics from phenomenology to mechanism. *Nature*, 541(7637):331–338, 2017.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

Archit Verma and Barbara E Engelhardt. A robust nonlinear low-dimensional manifold for single cell rna-seq data. *BMC bioinformatics*, 21(1):1–15, 2020.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):174, 2017.

A Baseline Models

A.1 scVI

Proposed in 2019 by Lopez et al. (2018), single-cell variational inference (scVI) is a variational autoencoder that is tuned for single-cell data and has been shown to match current state of the art methods in a variety of downstream tasks, including clustering and differential expression (Lopez et al., 2018; Luecken et al., 2022). Furthermore, due to its neural network structure, the model is scalable to large datasets. An overview of the model is presented in Figure 5.

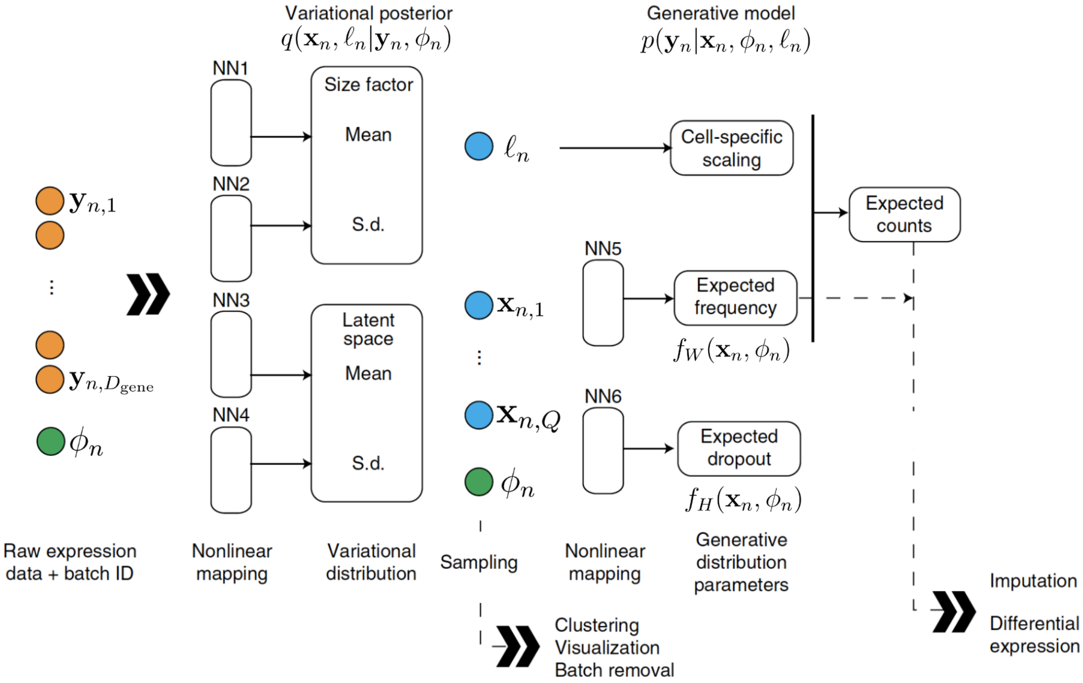


Figure 5: Overview of the scVI architecture adapted from Lopez et al. (2018).

We highlight several key components of the model that target phenomena commonly seen in single-cell data: (1) count data, (2) batch effect, and (3) library size normalization.

Count Data. As scRNA-seq raw count data are discrete, scVI adopts various discrete likelihoods, such as the negative binomial likelihood, for its models. This allows the model to learn a latent space directly from the raw expression data without any conventional pre-processing pipelines. Note that the original paper uses the zero-inflated negative binomial likelihood for the main model to account for dropouts, where gene expressions for a cell are not detected due to technical artifacts (Lopez et al., 2018; Luecken & Theis, 2019).

Accounting for Batch Effects. scVI also models for any effects from different sampling batches by incorporating batch ID information for each cell in both the encoding and decoding portions of the VAE model. While batch information is incorporated as input to the neural network encoder and decoders, it is unclear how exactly the batch effects are modelled.

Library Size Normalization. The third component scVI accounts for is the differences in total gene expression count per cell, or library size, of the data. In the raw count data, each cell has different total gene counts, which may affect comparisons between cells and impact downstream analysis (Hie et al., 2020). As this difference in library size, or sequencing depth, may be a result of technical noise, scVI chooses to model a scaling factor ℓ stand-in for library size. This latent variable is modelled as a log normal as done in Zappia et al. (2017) mappings from the raw counts and batch information to the mean and variance learned by the neural network encoder. To avoid conflating the effects of the scaling factor and of biological effects in the data, a softmax is applied to the output of the decoder before being multiplied by the scaling factor to determine the negative binomial likelihood mean.

With these three key components in mind, scVI's generative model for a given data point $\mathbf{y}_n$ as follows:

$$\text{Prior on latents: } p(\mathbf{x}_n) = \mathcal{N}(\mathbf{x}_n|0, \mathbf{I}_Q) \tag{11}$$

$$\text{Prior on scaling factor: } p(\ell_n) = \text{LogNormal}(\ell_n|\ell_\mu, \ell_{\sigma^2}) \tag{12}$$

$$\text{Likelihood: } p(\mathbf{y}_n|\mathbf{x}_n, \ell_n, \boldsymbol{\phi}_n) = \text{NegativeBinomial}\left(\mathbf{y}_n|\ell_n\text{softmax}\left(f_W(\mathbf{x}_n, \boldsymbol{\phi}_n)\right), r\right), \tag{13}$$

where $\boldsymbol{\phi}_n$ represents the batch information of cell n, $f_W(\mathbf{x}_n, \boldsymbol{\phi}_n)$ is a neural network decoder incorporating batch information, and ℓ_μ and ℓ_{σ^2} are given by the empirical mean and variance of the log library size in the batch containing data point n. Here, the negative binomial is parameterized by the mean and inverse dispersion, so that the model has mean $\ell_n\text{softmax}\left(f_W(\mathbf{x}_n, \boldsymbol{\phi}_n)\right)$ and shape r. In this parameterization, when $r \to \infty$, this distribution approaches a Poisson distribution with mean equivalent to $\ell_n\text{softmax}\left(f_W(\mathbf{x}_n, \boldsymbol{\phi}_n)\right)$.

The corresponding loss term for each data point is given by

$$\mathcal{L}(q(\mathbf{x}, \ell)) = \mathbb{E}_{q(\mathbf{x},\ell|\mathbf{y},\boldsymbol{\phi})}\left[\log p(\mathbf{y}|\mathbf{x}, \ell, \boldsymbol{\phi})\right] - \text{KL}(q(\mathbf{x}|\mathbf{y}, \boldsymbol{\phi})||p(\mathbf{x})) - \text{KL}(q(\ell|\mathbf{y}, \boldsymbol{\phi})||p(\ell)), \tag{14}$$

where the parameters to be optimized are the weights of the neural network encoders and decoders as well as the inverse dispersion factor r of the negative binomial likelihood. The way in which the loss can be decomposed into terms for datapoint allows the model to be trained with mini-batching (Hoffman et al., 2013).

While scVI has been shown to perform well in a variety of downstream tasks (Lopez et al., 2018; Luecken et al., 2022), its complex architecture (as seen in Figure 5) and opaque incorporation of known nuisance variables like batch effects make the model and its inferences difficult to interpret.

A.2 LDVAE

In response to this lack of interpretability in the original scVI, Svensson et al. (2020) proposed a linear version of scVI, where the neural network decoder is replaced with a linear mapping. In particular the LDVAE model is defined in the generative way as follows:

$$\text{Prior on the latent variables: } p(\mathbf{x}_n) = \mathcal{N}(\mathbf{x}_n|0, \mathbf{I}_Q), \tag{15}$$

$$\text{Prior on scaling factor: } p(\ell_n) = \text{LogNormal}(\ell_n|\ell_\mu, \ell_{\sigma^2}), \tag{16}$$

$$\text{Likelihood: } p(\mathbf{y}_n|\mathbf{x}_n, \ell_n) = \text{NegativeBinomial}\left(\mathbf{y}_n|\ell_n\text{softmax}\left(\mathbf{x}_n\mathbf{W}^T\right), r\right), \tag{17}$$

where $\mathbf{W}$ represents the linear mapping. Note that the mapping from latent space to data space is not completely linear as a nonlinearity is introduced in the softmax function. Moreover, Svensson et al. explored applying a BatchNorm layer to the linearly decoded parameters and found it matched or improved model performance in reconstruction error and learning the latent space in a mouse embryo development dataset (Svensson et al., 2020; Cao et al., 2019). This BatchNorm layer is thus adopted in the LDVAE model, which further obscures a straightforward interpretation of the mapping defined by the decoder.

Thus, while the LDVAE model allows for a more interpretable mapping from the latent space to the dimension space when compared to scVI, the use of a library size surrogate and a not clearly defined incorporation of batch information through NNs make both models less interpretable.

B Experiment Details

B.1 Data

We evaluate these models with two datasets: (1) a simulated dataset using the single-cell simulation framework Splatter (Zappia et al., 2017) and (2) a COVID-19 dataset (Stephenson et al., 2021).

Simulated Data. As the focus of our work is to dissect the assumptions made in single-cell data, we build our model based on a synthetic scRNA-seq dataset generated by the Splatter Splat scRNA-seq simulation (Zappia et al., 2017). The data are modelled off of a negative binomial distribution based on a hierarchical Gamma-Poisson model, where the parameters are drawn from the dataset (Kotliar et al., 2019). The data are simulated with seven cell types and two batches, with 10000 cells in each batch and 10000 genes per cell. We then remove cells with fewer than 200 total gene expression counts and genes that are expressed in three or fewer cells. This results in a synthetic dataset having 16016 cells and 8819 genes.

COVID-19 Data. The COVID-19 dataset (Stephenson et al., 2021) is a real world dataset comprised of gene expression counts obtained from peripheral blood mononuclear cells. This dataset includes samples from 107 patients exhibiting different degrees of COVID-19 severity, as well as samples from 23 healthy individuals. There are three main sampling locations – Sanger, Cambridge, and Newcastle – and the dataset also includes sample ID (143 batches total), where the sample IDs have unique codes for the sampling locations. There are 143 such sample IDs and 18 cell types considered. For this project, we take a subsample of this dataset that takes 100 000 cells and 5000 most variable genes as determined by Scanpy (Wolf et al., 2018).

Innate Immunity Data. The innate immunity dataset of Kumasaka et al. (2021) is comprised of 22,188 primary dermal fibroblasts from 68 donors who were either in the control group or were exposed to two stimulants to mimic innate immune response: (1) dsRNA Poly(I:C) for primary antiviral and inflammatory responses and (2) IFN-beta for secondary antiviral response. There were a total of 4999 genes and 7 latent dimensions (including cell-cycle latents).

B.2 Experimental Set-Up

For each of the experiments, we train the model with batch size 300, learning rate 0.05, and three different seeds: 0, 42, and 123. For the synthetic dataset, we train with 50 epochs and for the COVID-19 dataset, we use 15 epochs, which is sufficient for convergence for the corresponding datasets. The latent space dimension is set to $Q = 10$ for all models. For evaluation, we use seed 1 for all UMAP visualizations, and the latent metrics are reported with the average and standard deviation (each up to two decimal digits) over the three training runs for each model. We use the CSD3 high-performance computers for model training.

B.3 Extra Modifications and Experiments

B.3.1 Linear Kernel

For the ablation study, we also consider a linear kernel that models the augmented latent space information

$$k_{\text{linear}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}') = \nu \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}'^T, \tag{18}$$

where $\tilde{\boldsymbol{x}} = [\boldsymbol{x}^T \ \boldsymbol{\phi}^T]^T \in \mathbb{R}^{Q+D_{\text{covar}}}$ is the augmented latent variable that includes covariate information $\boldsymbol{\phi}$. ν is a variance parameter.

The corresponding augmented GP with linear mean and linear kernel is given by:

$$p(\tilde{f}_d | \tilde{X}) = \mathcal{N}(\tilde{f}_d | \mu_f \mathbf{1}_N + \tilde{X} w_d, \tilde{K}_{NN}), \tag{19}$$

where $\mathbf{1}_N \in \mathbb{R}^N$ is a vector of 1s, $w_d \in \mathbb{R}^Q$ defines the linear mean, $\tilde{X} \in \mathbb{R}^{N \times (Q \times D_{\text{covar}})}$ is the matrix of latent variables $\tilde{X}$ augmented by the known covariates $\boldsymbol{\Phi}$, and $\tilde{K}_{NN} = k_{\text{linear}}(\tilde{X}, \tilde{X})$.

B.3.2 Likelihoods

In our ablation studies, more complex likelihoods (for example, a negative binomial likelihood where the library size of each row was learned) were observed to perform poorly, and likelihood simplifications like using the approximate Poisson likelihood led to improved performance (see Fig. 6). This phenomenon could be explained by an issue with the identifiability of the model. The extra parameters in the model allow more flexibility in these likelihoods, but may also be learning and abstracting away pertinent cell-type information from the latent space variables. When the library size parameter is learned slowly, the model may be biased towards high-count cells, potentially disregarding the rest of the data and attributing latent space factors to technical noise rather than relevant biological differences. By constraining our likelihoods to slightly misaligned models, we may be encouraging the BGPLVM model to learn the 0s and smaller count values extremely well.

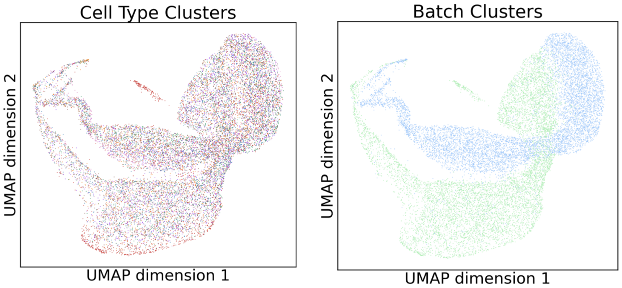


Figure 6: UMAP plots for an extended ablation study on the proposed model's likelihood. When library size is learned, the cell types become fully mixed (left) and the batches become separated (right).

C Latent Space Metrics

In this work, we compare these latent spaces both qualitatively and quantitatively. For qualitative measurements, we refer to UMAP 2-D visualizations (McInnes et al., 2018). However, since UMAP is a stochastic mapping and the visualized distances between datapoints are not reflective of the true distances in the latent space (McInnes et al., 2018), we also turn to quantitative latent space measurements pertinent to single-cell data. In particular, we focus on five quantitative metrics used in single-cell integration benchmarking that measure how well the latent space clusters cell types and how well the latent space mixes samples by batch (Luecken et al., 2022). The measurements for cell-type separation in latent space are detailed in Bio Conservation Metrics (Section C.1) and measurements for batch mixing are detailed in Batch Correction Metrics (C.2).

In our experiments, following convention in Luecken et al. (2022), we average the batch variables to obtain an average batch metric score, and we average the bio-conservation metrics to obtain an average bio metric score. While Luecken et al. (2022) propose an overall latent space score obtained through a weighted average of these two metrics, we deviate from this approach. We observed that models that failed to learn meaningful information could still yield high batch mixing scores (due to indiscriminate mixing) and consequently lead to misleading total scores. Hence, we choose to report only the average batch metric score and average bio metric score, separately.

C.1 Bio Conservation Metrics

We measure the latent space's ability to separate by cell-type with three different bio metrics: normalized mutual information (NMI), adjusted rand index (ARI), and cell-type average silhouette width (cellASW).

The NMI and ARI metrics require comparing cell-type information with learned clusterings from the latent space. To help make the metrics comparable for different models, we define the learned clusters with the Leiden clustering method with default resolution = 1 (Traag et al., 2019) on the latent space projections. We also considered k-means clustering on the latent space projections but found that the resulting metrics were sometimes not reflective of the perceived clusters (e.g. when clearly-defined clusters are long and thin and close together width-wise, k-means outputs poor metrics).

Normalized Mutual Information (NMI). NMI compares the overlap of two clusterings, taking on values between 0 and 1 where 0 indicates no overlap and 1 indicates perfect overlap.

More formally, let T define the true cell type labels with $\#T$ distinct clusters and C denote the predicted clusterings with $\#C$ distinct clusters. Furthermore, let $\{T_i\}_{i=1}^{\#T}$ denote the different clusters in T and $\{C_j\}_{j=1}^{\#C}$ denote the different clusters in C, and for each cluster $|C_j|$ is the number of samples in cluster C_j. N is the total number of samples being clustered. Then, NMI is defined as follows:

$$NMI(T,P) = \frac{2I(T;C)}{H(T)+H(C)}, \tag{20}$$

where

$$I(T;C) = \sum_{i=1}^{\#T}\sum_{j=1}^{\#C} P(T_i \cap C_j)\log\left(\frac{P(T_i \cap C_j)}{P(T_i)P(C_j)}\right)$$
$$= \sum_{i=1}^{\#T}\sum_{j=1}^{\#C} \frac{|T_i \cap C_j|}{N}\log\left(\frac{N|T_i \cap C_j|}{|T_i||C_j|}\right) \tag{21}$$

is the mutual information of T and C and

$$H(T) = -\sum_{i=1}^{\#T}\frac{|T_i|}{N}\log\frac{|T_i|}{N} \tag{22}$$

denotes the entropy of T (and is similarly defined for C).

Adjusted Rand Index (ARI). Adjusted Rand Index (ARI) also compares two clusterings but ARI (1) counts the pairwise agreements between the clusterings instead element-wise comparisons as done in NMI; and (2) adjusts for chance. The measurement usually takes on values between 0 and 1, and may extend to -0.5 for very different clusterings (Luecken et al., 2022).

For a given sample S of N samples, Rand index by itself captures the proportion of samples upon which the two clusterings X and Y capture similar information. More formally,

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{N}{2}}, \tag{23}$$

where

- a is the number of pairs that are in the same cluster in T and in the same cluster in C
- b is the number of pairs that are in different clusters in T and in different clusters in C
- c is the number of pairs that are in same cluster in T and in different clusters in C
- d is the number of pairs that are in different clusters in T and in the same cluster in C

ARI is the corrected-for-chance version of RI.

$$ARI = \frac{\text{RI} - \text{Expected RI}}{\max(\text{RI}) - \text{Expected RI}}. \tag{24}$$

Cell Average Silhouette Width (Cell type ASW). The cell-type average silhouette width measures how compact the predicted clusters are by comparing the intra-cluster distances with inter-cluster distances. A score of 1 indicates well-separated and compact clusters while a score of 0 indicates misaligned or overlapping clusters. The clusters in this case are defined by the cell-types.

For a cell n of cell type C_j, its silhouette score is defined as:

$$s(n) = \frac{b(n) - a(n)}{\max(a(n), b(n))}, \tag{25}$$

where $a(n)$ is the average (Euclidean) distance between cell n and the other cells of the same cell-type and $b(n)$ is the minimum average (Euclidean) distance between cell n and a cell of a different cell type. More formally,

$$a(k) = \frac{1}{|C_j| - 1}\sum_{l \in C_j} d(k,l), \tag{26}$$

$$b(k) = \min_{j' \neq j}\frac{1}{|C_{j'}|}\sum_{l \in C_{j'}} d(k,l). \tag{27}$$

Then, the average silhoutte width for each cell-type cluster C_j is defined as the average silhoutte scores for each cell of that type:

$$\text{ASW}_j = \frac{1}{|C_j|}\sum_{n \in C_j} s(n). \tag{28}$$

Cell type ASW simply scales the average silhouette width over all cell-types so that instead of taking values between -1 and 1, it takes on values between 0 and 1:

$$\text{cell type ASW}_j = \frac{1}{2}\left(1 + \frac{1}{M}\sum_{j=1}^{M} ASW_j\right),$$

where M is the total number of cell types.

C.2 Batch Correction Metrics

Batch Average Silhouette Width. Much like Cell-type ASW, Batch ASW also measures the compactness of the predicted clusters. However, for the case of batches, we want the clusters to be spread out, so the Batch ASW formula must be adjusted accordingly so that a score of 1 reflects well-mixed batches and a score of 0 reflects poorly mixed batches. This is done by first introducing the absolute silhouette width for a cell n,

$$s_{\text{batch}}(n) = |s(n)|, \tag{29}$$

so that 0 represents a perfectly-mixed batch and any other value represents some deviation from being well-mixed.

The Batch ASW for a cell-type j is then

$$\text{batch ASW}_j = \frac{1}{|C_j|}\sum n \in C_j 1 - s_{\text{batch}}(n). \tag{30}$$

The overall Batch ASW is given by

$$\text{batch ASW} = \frac{1}{M}\sum_{j=1}^{M} \text{batch ASW}_j. \tag{31}$$

Graph Connectivity The graph connectivity score represents how well the kNN graph connects cells of the same type. If there is good batch mixing, we would expect the cells of the same type to be clustered together, representing well connected same cell-type subgraphs. Conversely, when batches are not corrected for, cells of the same type could be dispersed across the latent space and not connected by the kNN graph.

This idea is formally represented by the following graph connectivity metric:

$$\text{GC} = \frac{1}{M}\sum_{j=1}^{M} \frac{|\text{LCC}(G(N_j; E_j)}{|N_j|}, \tag{32}$$

where $G(N_j; E_j)$ is the subgraph containing only cells of cell type j, N_j is the set of nodes of cell-type j, and LCC$(G(N_j; E_j))$ is the Largest Connected Component of the subgraph $G(N_j; E_j)$.

D Detailed Metrics

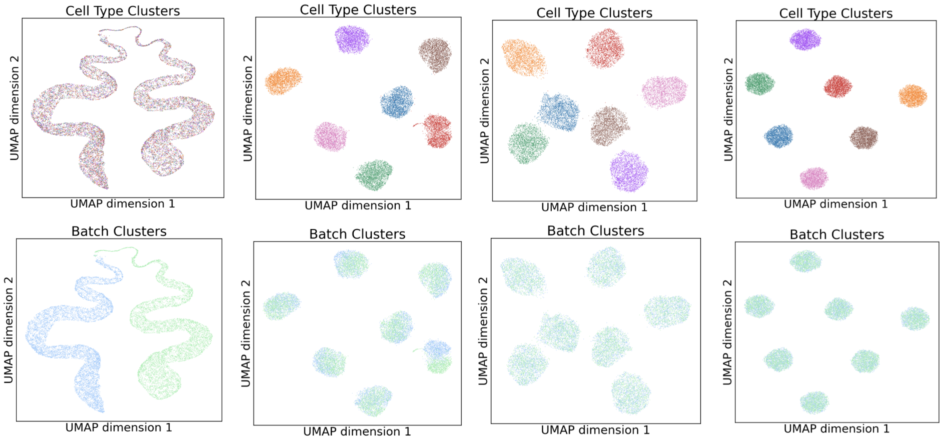
We report the latent metrics for the first two experiments, taking the mean and standard deviation across trained models from three different seeds. Blue columns correspond to batch metrics and Green columns correspond to cell-type metrics.

D.1 Ablation study

Model Change	BatchASW	iLisi	kBET	Graph Connectivity	NMI Leiden	ARI Leiden	cellASW	cLisi
Simple NN	0.843 ± 0.003	0.881 ± 0.095	0.141 ± 0.245	0.338 ± 0.571	0.237 ± 0.394	0.110 ± 0.189	0.557 ± 0.157	0.511 ± 0.424
Gaussian Likelihood	0.447 ± 0.164	0.000 ± 0.000	0.015 ± 0.005	0.262 ± 0.158	0.003 ± 0.001	0.000 ± 0.000	0.429 ± 0.010	0.335 ± 0.010
Linear Kernel	0.088 ± 0.005	0.000 ± 0.000	0.000 ± 0.000	0.517 ± 0.001	0.627 ± 0.069	0.352 ± 0.100	0.486 ± 0.005	0.988 ± 0.005

Table 1: Latent space metrics for the ablation study on simulated dataset.

D.2 Benchmarking



(a) OBGPLVM (b) Proposed Model (c) scVI (d) LDVAE

Figure 7: UMAPs generated from the latent spaces of four models: an implementation of the original BGPLVM, the modified BGPLVM for scRNA-seq data, scVI, and a linear decoder scVI (LDVAE) for the simulated dataset. The top row is color/shaded by cell type and the bottom by batch.

Model	BatchASW	iLisi	kBET	Graph Connectivity	NMI	ARI	cellASW	cLisi
OBGPLVM	0.527 ± 0.472	0.319 ± 0.495	0.284 ± 0.424	0.582 ± 0.200	0.00247 ± 0.00036	0.00026 ± 0.00020	0.475 ± 0.004	0.321 ± 0.002
Proposed model	0.877 ± 0.049	0.570 ± 0.199	0.260 ± 0.149	0.998 ± 0.001	0.912 ± 0.061	0.849 ± 0.119	0.643 ± 0.026	1.0 ± 0
LDVAE	0.978 ± 0.049	0.913 ± 0.199	0.854 ± 0.149	1.000 ± 0.001	0.999 ± 0.061	1.000 ± 0.119	0.700 ± 0.026	1.0 ± 0.0
scVI	0.983 ± 0.002	0.916 ± 0.005	0.885 ± 0.028	1.000 ± 0.001	0.903 ± 0.061	0.805 ± 0.158	0.601 ± 0.012	1.000 ± 0.001

Table 2: Latent space metrics for benchmarking on the simulated dataset.

Model	**BatchASW**	**iLisi**	**kBET**	**Graph Connectivity**	**NMI**	**ARI**	**cellASW**	**cLisi**
OBGPLVM	0.795 ± 0.088	0.377 ± 0.055	0.465 ± 0.214	0.398 ± 0.258	0.265 ± 0.230	0.055 ± 0.065	0.459 ± 0.095	0.894 ± 0.086
Proposed model	0.848 ± 0.040	0.230 ± 0.017	0.884 ± 0.009	0.903 ± 0.028	0.606 ± 0.049	0.369 ± 0.053	0.492 ± 0.053	0.989 ± 0.002
linear scVI	0.918 ± 0.001	0.319 ± 0.003	0.861 ± 0.011	0.925 ± 0.001	0.690 ± 0.004	0.458 ± 0.007	0.578 ± 0.001	0.996 ± 0.000
scVI	0.945 ± 0.002	0.335 ± 0.004	0.881 ± 0.005	0.947 ± 0.002	0.722 ± 0.019	0.538 ± 0.053	0.544 ± 0.006	0.995 ± 0.000

Table 3: Latent space metrics for benchmarking on the COVID-19 dataset