scMVAE: a Mixed-Curvature Variational Autoencoder for Single-Cell RNA Data

Colin Doumont
Dpt. of Mathematics
ETH Zürich
Zürich, Switzerland
cdoumont@student.ethz.ch

Christophe Muller
Dpt. of Mathematics
ETH Zürich
Zürich, Switzerland
mullec@student.ethz.ch

Andrei Papkou
Dpt. of Evolutionary Biology
University of Zürich
Zürich, Switzerland
andrei.papkou@uzh.ch

Félix Vittori
Dpt. of Mathematics
ETH Zürich
Zürich, Switzerland
fvittori@student.ethz.ch

***Abstract*—Single-cell RNA sequencing (scRNA-seq) produces data that can be difficult to analyze due to high sparsity, missing observations, and technical biases. Variational autoencoders (VAEs) have become a popular method for scRNA-seq data analysis, but can suffer from posterior collapse, limiting downstream applications. To address this issue, scPhere, a VAE with a spherical latent space, was proposed and shown to improve scRNA-seq data representations. In this paper, we extend scPhere by replacing its VAE with a mixed-curvature VAE (M-VAE), that can also consider non-Euclidean spaces or mixed-spaces as latent space, and that learns their optimal curvature. Our experiments show that non-Euclidean spaces outperform Euclidean spaces in low dimensions, and that the universal model, which learns both the optimal latent space and its curvature, is a reasonable choice in all tested cases. However, learning the curvature and considering mixed-spaces did not prove beneficial.**

I. Introduction

Single-cell RNA sequencing (scRNA-seq) is a collection of new techniques in biology, allowing to study gene activity at single-cell resolution [1]. However, the data produced by these technologies is challenging to analyse for multiple reasons, including high sparsity, missing observations, biological complexity and technical biases [2]. As the size of data grows and the accessibility of GPU hardware increases, deep learning is becoming a method of choice for single-cell data analysis [3]. More specifically, for such tasks as lower-dimensional representation, batch correction and data imputation, Variational Autoencoders (VAE) are the current state-of-the-art and outperform other analytical methods [4, 5, 6, 3, 7, 8, 9, 10].

However, a known issue of VAE's is the collapse of the latent space [11], meaning the latent variables are not used for the model, and the posterior becomes the same as the prior. This is particularly problematic for scRNA-seq data, since it severely limits the downstream applications (such as clustering). To solve this posterior collapse problem, different VAE's with more flexible priors were proposed, such as the VAE-VAMP [12] or VAE-IAF [13], improving biological representations of scRNA-seq data [14]. Alternatively, Xu and Durrett proposed a VAE with a spherical latent space, which they showed reduces the posterior collapse problem for text data [15]. Afterwards, Ding et al. used this spherical/hyperbolic VAE to create the scPhere model for single-cell data, leading to a significant improvement of low-dimensional representations [16].

Given the increased performance of using spherical and hyperbolic latent spaces, a natural next step is to see whether extending scPhere to a more general non-Euclidean setting delivers better results. To do so, we modify scPhere by replacing its VAE with a Mixed-Curvature Variational Autoencoder (M-VAE) [17]. In contrary to the VAE presented in scPhere, the M-VAE not only considers non-Euclidean spaces (e.g. $\mathbb{S}^4$), but also considers products of these spaces (e.g. $\mathbb{S}^2 \times \mathbb{P}^2$) and then learns the "optimal" combination of such spaces. Additionally, instead of using a fixed curvature, the M-VAE now learns the "optimal" curvature, generalising scPhere's VAE.

In this paper, we first present the mathematical derivations needed to replace scPhere's VAE by the more general M-VAE (II-A). Afterwards, we detail the methods used to compare the performance of the original model, scPhere, with our extended version, scMVAE (II-B). Lastly, we go over the results of these experiments (III) and discuss their implications for single-cell analysis (IV).

II. Model & Methods

In this section, we first present the model, which consists of the mathematical framework behind scMVAE, as well as the necessary equations for non-Euclidean clustering. Afterwards, we present the methods, namely the nature of the data, specification of the reconstruction loss, solutions to numerical instabilities, choice of hyperparameters and design of experiments.

A. Model

The present study introduces scMVAE, a modified version of the M-VAE model specifically designed for single-cell data analysis. By leveraging the structure of M-VAE, scMVAE is able to learn an appropriate latent space or combination of latent spaces, as well as determine their optimal curvature. Adapting M-VAE to handle single-cell data was achieved through incorporation of the characteristics of the scPhere model.

Similarly to M-VAE, scMVAE incorporates three Riemannian manifolds: spherical space with positive curvature, $\mathbb{S}$, hyperbolic space with negative curvature, $\mathbb{H}$, and Euclidean

space with zero curvature, $\mathbb{E}$. The model also includes projections of these spaces: the projected hypersphere (projection of spherical space), $\mathbb{D}$, and the Poincaré ball (projection of hyperbolic space), $\mathbb{P}$. These projections are defined by the formulas 1 and 2 in Appendix A and possess the useful property of having a gyrospace-distance metric, allowing for adaptation of curvature while preserving a consistent distance[1]. The spaces are characterized by their curvature K. As stated above, the model is also able to consider any combination of the aforementioned spaces, such as $\mathbb{P}^2 \times \mathbb{E}^2 \times \mathbb{H}^2$, for example.

scMVAE employs the same prior and posterior distributions as M-VAE, namely Wrapped Normal distributions for non-Euclidean spaces and normal distributions for Euclidean spaces. In this regard, scMVAE diverges from scPhere, which utilizes a hyperspherical uniform distribution as prior and a von Mises–Fisher distribution as posterior for spherical spaces. Despite this divergence, scMVAE does incorporate certain characteristics of scPhere, to tailor the model for single-cell data analysis. For example, the model utilizes scPhere's batch variable implementation, which involves concatenating batch variables in the input of the encoder, as well as in the input of the decoder. Additionally, scMVAE employs a Negative Binomial distribution to compute the log-likelihood of the reconstruction, as is typically done with scRNA-seq data. Lastly, scMVAE also contains the added penalty term introduced in scPhere, that is a sum of squared errors.

To compare our model (scMVAE) with scPhere, we look at clustering accuracy. More specifically, we fit a k-nearest neighbors (kNN) model on the training set of the data, and then compute the accuracy of the kNN model on the test set. This way, if the different classes of our embeddings overlap, the kNN test accuracy should be very low, and vice versa. Hence, the higher the kNN accuracy, the better our latent representations and thus the better our model. However, before fitting a kNN model on points that lie in a product of non-Euclidean spaces, one must first define how to measure distance between such points. To do so, we adhere to the equations presented in M-VAE. Namely, for points $\boldsymbol{x}$ and $\boldsymbol{y}$ on mixed-space $\mathcal{M}' = \bigtimes_{i=1}^{k} \mathcal{M}_{K_i}^{n_i}$, where n is the dimensionality of the space, K is its curvature, and $\mathcal{M} \in \{\mathbb{E}, \mathbb{S}, \mathbb{D}, \mathbb{H}, \mathbb{P}\}$ is the model choice, we define the distance as $d^2_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{k} d^2_{\mathcal{M}_{K_i}^{n_i}}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$, where $\boldsymbol{x}^{(i)}$ represents a vector in $\mathcal{M}_{K_i}^{n_i}$, corresponding to the part of the latent space representation of $\boldsymbol{x}$ belonging to $\mathcal{M}_{K_i}^{n_i}$.

B. Methods

The idea behind scMVAE is to adapt the M-VAE for processing of scRNA-seq data. This unique data consists of RNA counts for M cells and N genes, for which technical/biological batch effects need to be considered. From this point onward, we will let $x_{i,j}$ represent the RNA count of gene j in cell i, and will let $\boldsymbol{x_i}$ represent the counts of all genes in cell i. Additionally, we will aggregate the measures of batch variables for cell i in $\boldsymbol{y_i}$.

The above vectors are then passed into the model as follows. First, the encoder takes $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$ as input and maps them through a neural network to the parameters of the posterior distribution, which is a Wrapped Normal distribution. Once that is done, a latent vector $\boldsymbol{z_i}$, representing a point in the latent space $\mathcal{M}'$, is sampled from this posterior distribution. Afterwards, the second part of the VAE, namely the decoder, takes $\boldsymbol{z_i}$ and $\boldsymbol{y_i}$ as input and maps them through another neural network to an output vector $\boldsymbol{x_i^*}$. If the model is well-trained, the output should approximate the input $\boldsymbol{x_i}$ (autoencoder structure).

From the approximation produced by the decoder, we define a loss function that consists of two parts: the KL-divergence between the posterior distribution and the prior distribution, and a reconstruction loss. The KL-divergence is scaled by a hyperparameter β and the reconstruction loss is calculated by taking the negative log-probability of observing $\boldsymbol{x_i}$, given a Negative Binomial distribution with $\boldsymbol{x_i^*}$ successes and a probability of success of $p = 0.5$. This probability p was chosen so that $\boldsymbol{x_i^*}$ is the mean of the distribution. In addition to the log-probability, we also include a penalty loss, inspired by the scPhere model, equal to the squared error between the approximation and the input, i.e. $\sum_{j=0}^{N}(\boldsymbol{x_i^*} - \boldsymbol{x_i})^2$. Before applying the reconstruction loss, we set a lower-bound of 10^{-5} to the value of $\boldsymbol{x_i^*}$ to avoid collapse.

During training, we found that our scMVAE model was unstable and suffering from exploding gradients, possibly due to the complexity of scRNA-seq data or due to the optimization challenges associated with learning the curvature. To address this issue, we implemented several adaptations to ensure numerical stability of the model. Specifically, we applied normalization to the data in both a min-max and logarithmic scale. We also explored various reconstruction losses to account for the sparsity of the data. However, these approaches did not significantly improve model stability. Instead, we found that implementing batch normalization in both the encoder and decoder networks had the most significant impact on improving stability. Despite these challenges, we managed to get the rate of numerical instability down from 100% to less than 3%. We also used gradient clipping in a similar fasion as the M-VAE model (`grad_norm` $= 1.0$).

After solving numerical instabilities, we tested the model on various scRNA-seq datasets and compared its performance, i.e. kNN accuracy, with scPhere in several experiments, presented in Section III. We designed each experiment to address different aspects of our model. In particular, Experiment I evaluates the performance of the scMVAE model across different latent spaces and mixed-spaces, holding the curvature of these spaces fixed at $|K| = 0.25$, thus allowing us to test the value of more complex spaces compared to simpler ones. Experiment II assesses the benefit of learning the curvature of the latent space, compared to the corresponding fixed curvature models (such as scPhere). In Experiment III, we use the universal model, in which scMVAE learns the optimal latent space

[1]The spherical and hyperbolic space's distances are inconsistent when the curvature approaches 0, see Appendix A.

or mixed-space, and compare its performance to the other models, where the type of space or mixed-space has been fixed in advance. In Experiment IV, we test whether scMVAE can remove multiple technical and biological batch effects while preserving meaningful information, using a dataset which combines samples from heterogeneous sources.

The default settings of these experiments are displayed in Table III. The `Warmup` parameter refers to the number of initial epochs during which the early stopping mechanism is inactive. This mechanism uses the parameter `Lookahead` to halt training when the reconstruction loss does not decrease within that number of epochs. The `Hidden Units` parameter corresponds to the number of neurons in the encoder and decoder layers, where all of the neurons had a ReLu activation function. Finally, β, from the loss function, starts at `Beta Start` and linearly increases to `Beta End` at `Beta End Epoch`.

The experiments were performed using four datasets. Since the performance of a particular latent space may depend on the size and complexity of the problem, we first used a relatively small dataset: ADIPOSE. ADIPOSE contains scRNA data from 1378 cells of mouse white adipose tissue [18], with high-quality labels identifying six cell types [16]. As a more complex dataset (35699 cells), we used the retinal ganglion cells dataset (RGC), containing 45 neural cells types from mouse retina [19]. The third dataset, CELEGANS, contains an even larger number of cells (86024), as well as 36 major cell types of the nematode *C. elegans* sampled from different points of embryonic development [20]. Including the ADIPOSE, RGC and CELEGANS datasets allowed us to test scMVAE's performance against an increasing size and complexity of input data. Finally, for Experiment IV, we used a dataset containing cells from more than 50 intestinal biopsies from 30 ulcerative colitis patients and healthy individuals [21]. Moreover, the biopsies were sampled from different anatomical locations within the same individual. This dataset contains three major cell groups, from which we have chosen 64457 epithelial cells (with 12 different cell types). We refer to this subset as UC_EPI.

As mentioned above, the main metric for comparing the performance of our models is the kNN test accuracy. For this kNN accuracy, we used grid search over $k \in \{2, 4, 8\}$ with 5-fold cross validation to determine the optimal number of neighbors k, which we then used to make the final predictions. For the ADIPOSE dataset, since it's relatively small, we ran the algorithm on the whole test set. For the three other (larger) datasets, we subsampled 10 test sets of 1000 cells, computed kNN on each set and kept the average accuracy as kNN score. Additionally, we ran every VAE model multiple times (10+ for ADIPOSE and RGC, 5+ for CELEGANS and UC_EPI) and recorded the kNN score for each of these runs. The multiple kNN samples and model runs were combined into two columns: (avg.) accuracy and its std. deviation (Tables V-VIII).

III. Results

A. Experiment I

When looking at all fixed-curvature non-universal models, we observed three significant trends across all four datasets. *Trend 1:* changing the dimension of the latent space can have a big impact on accuracies, where the best overall dimension sizes seem to be 6 and 12 (Table IX, Fig. 1). *Trend 2:* for the same dimension size, mixed-spaces perform worse than individual spaces, even Euclidean ones. *Trend 3:* for low dimensions, spherical spaces perform best across datasets, followed by hyperbolic spaces and Euclidean spaces, respectively, with the exception of the ADIPOSE case, the smallest dataset.

B. Experiment II

When comparing fixed-curvature non-universal models with learned-curvature non-universal models, one important trend seems to stand out. *Trend 4*: learning the curvature does not significantly impact accuracy (Fig. 2).

C. Experiment III

When examining universal models, which learn both the curvature and optimal latent space, one trend emerges. *Trend 5*: universal models perform comparably to the best other models, but do not significantly outperform them.

D. Experiment IV

ScRNA datasets are often subjected to different biases, due to technical and biological variability. To account for these biases, we extended MVAE to consider one or multiple batch effects (see Methods). To test the ability of scMVAE to remove these batch effects, we trained it using the UC_EPI dataset, which contains three different batch effects: patient id, health status and anatomical location [21]. To assess if these batch effects were still present in the scMVAE representations, we used the silhouette score (as suggested in [22, 4]). We found that all batch effects were close to 0 or negative, suggesting that cells do not cluster by their batch category (Fig. 3). This indicates a successful removal of these effects by scMVAE, while preserving biologically meaningful categories (as suggested by the kNN accuracy of 70% for best performing models in UC_EPI).

E. Notes on Scalability

Table IV shows estimates of the training running-time of the different datasets. This shows that scMVAE can even handle large datasets, while requiring only the equivalent of a modern laptop.

IV. Discussion

Since scMVAE is a generalisation of scPhere, one would expect its results on similar datasets to be equal or better. Interestingly, this is not the case. For the RGC dataset, we obtained accuracies below scPhere, even for analogous models (our best score is 86.7% vs. >95%). Similarly, the best scMVAE accuracy in UC_EPI dataset was lower than

in scPhere (70.8% vs. 85%). Lower accuracies for our models may be explained by the fact that we used small samples (1000 cells) to train the kNN classifier. In addition, we trained and tested the kNN classifier strictly using train and test data for scMVAE models. In this regard, the paper by Ding et al. is ambiguous about the details of their analysis.

One of the main strengths of the scMVAE model is its ability to capture complex patterns of gene expression through the use of more complex latent spaces or mixed-spaces. This enables the model to more accurately represent the structure of the data, and may provide more nuanced insights into the underlying biological processes at work. The implementation of a flexible pipeline for training the model and evaluating its performance is also a significant strength, as it allows for seamless experimentation and optimization.

On the other hand, there are several limitations of the scMVAE model that should be considered. One limitation is that the model relies on certain assumptions that may not hold for all datasets. For example, the Wrapped Normal distribution used as the prior and posterior, as well as the Negative Binomial distribution used to model the scRNA-seq counts, may not be appropriate for all datasets. Another limitation is that the user is required to specify the dimensionality of the latent space, which can be difficult to determine and may impact the model's performance. Additionally, the scMVAE model lacks interpretability, a common limitation among most deep learning models, making it difficult to understand the internal functioning of the model. Finally, the complexity of the latent space can make visualization of the results challenging.

There are a few implications of our results. On the one hand, it appears that non-Euclidean latent spaces are generally more effective for representing single-cell data compared to Euclidean spaces. This suggests that the complex structure of scRNAseq data may be better captured by the non-Euclidean geometry of these spaces. Furthermore, the universal method, which allows the model to learn both the optimal latent space and its curvature, was found to be a reliable choice across all of our experiments. On the other hand, our results showed that mixed-spaces performed worse than individual spaces, indicating that the combination of different types of latent spaces may not be beneficial in this context. Additionally, we found that learning the curvature did not significantly improve model performance. This suggests that, while non-Euclidean latent spaces may be beneficial, the specific curvature of these spaces may not be as important for representing scRNAseq data. Overall, these findings have implications for the design of models for single-cell data analysis, and highlight the potential benefits of using non-Euclidean latent spaces in these models.

There are several directions for extending the work presented in this paper. One direction is to address the main weakness of the scPhere model, namely the fact that the second Negative Binomial parameter is set to 0.5 rather than being learned by the model. Changing this would require modifying the architecture of the decoder to output two variables for each observation. Another potential direction is to explore a wider range of latent spaces and mixed-spaces, including combinations of the components implemented in this work, different numbers of dimensions, or learning the curvature in other non-Euclidean spaces such as $\mathbb{D}$ and $\mathbb{P}$. Additionally, the model could be improved by considering various options that were not explored due to computational and time constraints, such as different numbers of layers and hidden units, alternative activation functions, lower learning rates, different β's, etc., which could all have a big impact on performance.

V. Summary

Single-cell RNA sequencing (scRNA-seq) data is difficult to analyze due to high sparsity, biological complexity, and technical biases. To address these challenges, deep learning methods, particularly Variational Autoencoders (VAEs), have become popular for tasks like lower-dimensional representation, batch correction, and data imputation. However, VAEs can suffer from the collapse of the latent space, which can limit downstream applications. The Mixed-Curvature Variational Autoencoder (M-VAE) model addresses this issue by considering non-Euclidean spaces and learning the optimal combination of these spaces, as well as the optimal curvature. Here, we adapted the M-VAE model for use in single-cell data analysis by incorporating characteristics from the scPhere model, i.e. accounting for the batch effect(s) and using appropriate distributions to represent the data. The scMVAE model is able to handle three distinct Riemannian manifolds: spherical, hyperbolic, and Euclidean, as well as projections of these spaces.

scMVAE shows some promising results. First, non-Euclidean spaces outperformed their Euclidean counterparts in low dimensions experiments. We also show that the universal model is a reasonable choice in all tested cases. Additionally, the scMVAE model was able to effectively remove multiple batch effects while preserving biologically meaningful categories in the UC_EPI dataset. This is a significant strength as the ability to remove batch effects enables the joint analysis of multiple datasets, which is increasingly important as scRNA-seq databases continue to grow. However, there are some limitations to the scMVAE model that should be considered. These include its reliance on certain assumptions that may not hold for all datasets, the need for the user to specify the dimensionality of the latent space, a lack of interpretability, and the complexity of the latent space, which can make visualization and interpretation more difficult. Also, some key features of scMVAE, namely learning the curvature of the latent-space and considering mixed-spaces, did not prove beneficial. Future work may include exploring new encoder and decoder architectures, decoding a second Negative Binomial variable, and optimizing hyperparameters to further improve the performance of the scMVAE model.

Data and Code availability

All datasets are available from the Single Cell Portal[23]. You can find an implementation of this paper on Github:

https://github.com/Felixiose/scMVAE

References

[1] Oliver Stegle, Sarah A. Teichmann, and John C. Marioni. "Computational and analytical challenges in single-cell transcriptomics". en. In: *Nature Reviews Genetics* 16.3 (Mar. 2015). Number: 3 Publisher: Nature Publishing Group, pp. 133–145. ISSN: 1471-0064. DOI: 10.1038/nrg3833. URL: https://www.nature.com/articles/nrg3833 (visited on 11/10/2022).

[2] Vladimir Yu Kiselev, Tallulah S. Andrews, and Martin Hemberg. "Challenges in unsupervised clustering of single-cell RNA-seq data". en. In: *Nature Reviews Genetics* 20.5 (May 2019). Number: 5 Publisher: Nature Publishing Group, pp. 273–282. ISSN: 1471-0064. DOI: 10.1038/s41576-018-0088-9. URL: https://www.nature.com/articles/s41576-018-0088-9 (visited on 10/26/2022).

[3] Malte D. Luecken et al. "Benchmarking atlas-level data integration in single-cell genomics". en. In: *Nature Methods* 19.1 (Jan. 2022), pp. 41–50. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-021-01336-8. URL: https://www.nature.com/articles/s41592-021-01336-8 (visited on 12/11/2022).

[4] Romain Lopez et al. "Deep generative modeling for single-cell transcriptomics". en. In: *Nature Methods* 15.12 (Dec. 2018). Number: 12 Publisher: Nature Publishing Group, pp. 1053–1058. ISSN: 1548-7105. DOI: 10.1038/s41592-018-0229-2. URL: https://www.nature.com/articles/s41592-018-0229-2 (visited on 12/11/2022).

[5] Christopher Heje Grønbech et al. "scVAE: variational auto-encoders for single-cell gene expression data". In: *Bioinformatics* 36.16 (Aug. 2020), pp. 4415–4422. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa293. URL: https://doi.org/10.1093/bioinformatics/btaa293 (visited on 12/11/2022).

[6] Qiwen Hu and Casey S. Greene. "Parameter tuning is a key part of dimensionality reduction via deep variational autoencoders for single cell RNA transcriptomics". eng. In: *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* 24 (2019), pp. 362–373. ISSN: 2335-6936.

[7] Yue Cao, Pengyi Yang, and Jean Yee Hwa Yang. "A benchmark study of simulation methods for single-cell RNA sequencing data". en. In: *Nature Communications* 12.1 (Nov. 2021), p. 6911. ISSN: 2041-1723. DOI: 10.1038/s41467-021-27130-w. URL: https://www.nature.com/articles/s41467-021-27130-w (visited on 12/11/2022).

[8] Gökcen Eraslan et al. "Single-cell RNA-seq denoising using a deep count autoencoder". eng. In: *Nature Communications* 10.1 (Jan. 2019), p. 390. ISSN: 2041-1723. DOI: 10.1038/s41467-018-07931-2.

[9] Cédric Arisdakessian et al. "DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data". eng. In: *Genome Biology* 20.1 (Oct. 2019), p. 211. ISSN: 1474-760X. DOI: 10.1186/s13059-019-1837-6.

[10] Bin Yu et al. "scGMAI: a Gaussian mixture model for clustering single-cell RNA-Seq data based on deep autoencoder". eng. In: *Briefings in Bioinformatics* 22.4 (July 2021), bbaa316. ISSN: 1477-4054. DOI: 10.1093/bib/bbaa316.

[11] Aman Singh and Tokunbo Ogunfunmi. "An Overview of Variational Autoencoders for Source Separation, Finance, and Bio-Signal Applications". eng. In: *Entropy (Basel, Switzerland)* 24.1 (Dec. 2021), p. 55. ISSN: 1099-4300. DOI: 10.3390/e24010055.

[12] Jakub M. Tomczak and Max Welling. *VAE with a VampPrior*. arXiv:1705.07120 [cs, stat]. Feb. 2018. URL: http://arxiv.org/abs/1705.07120 (visited on 12/11/2022).

[13] Diederik P. Kingma et al. *Improving Variational Inference with Inverse Autoregressive Flow*. arXiv:1606.04934 [cs, stat]. Jan. 2017. URL: http://arxiv.org/abs/1606.04934 (visited on 12/11/2022).

[14] Leander Dony et al. "Variational autoencoders with flexible priors enable robust distribution learning on single-cell RNA sequencing data". In: *ICML 2020 Workshop on Computational Biology (WCB) Proceedings Paper*. Vol. 37. 2020.

[15] Jiacheng Xu and Greg Durrett. *Spherical Latent Spaces for Stable Variational Autoencoders*. arXiv:1808.10805 [cs]. Oct. 2018. DOI: 10.48550/arXiv.1808.10805. URL: http://arxiv.org/abs/1808.10805 (visited on 12/11/2022).

[16] Jiarui Ding and Aviv Regev. "Deep generative model embedding of single-cell RNA-Seq profiles on hyperspheres and hyperbolic spaces". en. In: *Nature Communications* 12.1 (May 2021). Number: 1 Publisher: Nature Publishing Group, p. 2554. ISSN: 2041-1723. DOI: 10.1038/s41467-021-22851-4. URL: https://www.nature.com/articles/s41467-021-22851-4 (visited on 12/11/2022).

[17] Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. *Mixed-curvature Variational Autoencoders*. arXiv:1911.08411 [cs, stat]. Feb. 2020. URL: http://arxiv.org/abs/1911.08411 (visited on 12/11/2022).

[18] Chelsea Hepler et al. "Identification of functionally distinct fibro-inflammatory and adipogenic stromal subpopulations in visceral adipose tissue of adult mice". eng. In: *eLife* 7 (Sept. 2018), e39636. ISSN: 2050-084X. DOI: 10.7554/eLife.39636.

[19] Nicholas M. Tran et al. "Single-Cell Profiles of Retinal Ganglion Cells Differing in Resilience to Injury Reveal Neuroprotective Genes". eng. In: *Neuron* 104.6 (Dec. 2019), 1039–1055.e12. ISSN: 1097-4199. DOI: 10.1016/j.neuron.2019.11.006.

[20] Jonathan S. Packer et al. "A lineage-resolved molecular atlas of C. elegans embryogenesis at single-cell resolution". In: *Science* 365.6459 (Sept. 2019). Publisher: American Association for the Advancement of Science, eaax1971. DOI: 10.1126/science.aax1971. URL: https://

www.science.org/doi/10.1126/science.aax1971 (visited on 12/11/2022).

[21] Christopher S. Smillie et al. “Intra- and Inter-cellular Rewiring of the Human Colon during Ulcerative Colitis”. eng. In: *Cell* 178.3 (July 2019), 714–730.e22. ISSN: 1097-4172. DOI: 10.1016/j.cell.2019.06.029.

[22] Michael B. Cole et al. “Performance Assessment and Selection of Normalization Procedures for Single-Cell RNA-Seq”. In: *Cell Systems* 8.4 (Apr. 24, 2019). Publisher: Elsevier, 315–328.e8. ISSN: 2405-4712. DOI: 10.1016/j.cels.2019.03.010. URL: https://www.cell.com/cell-systems/abstract/S2405-4712(19)30080-8 (visited on 01/01/2023).

[23] *scPhere - Single Cell Portal*. URL: https://singlecell.broadinstitute.org/single_cell/study/SCP551/scphere (visited on 12/11/2022).

APPENDIX A
GEOMETRY OF RIEMANNIAN MANIFOLDS

The projection of a n-dimensional Spherical space of curvature $K > 0$, $\mathbb{S}^n_K$, to a Projected Hypersphere space $\mathbb{D}^n_K$ is done by Formula 1. And the projection of a Hyperbolic space of dimension n and curvature $K < 0$, $\mathbb{H}^n_K$, to a Poincaré space $\mathbb{P}^n_K$ is Formula 2.

$$\mathbb{D}^n_K = \rho_K(\mathbb{S}^n_K \ \{-\mu_0\}) \quad \text{where} \quad \mu_0 = (1/\sqrt{|K|}, 0, \cdots, 0)^T \in \mathbb{S}_{\mathbb{K}} \tag{1}$$

$$\mathbb{P}^n_K = \rho_K(\mathbb{H}^n_K) = \{x \in \mathbb{R}^n : \langle x, x\rangle_2 < -\frac{1}{K}\} \tag{2}$$

With ρ_K being the projection function of a point $(\xi; x^T)T \in \mathbb{R}^{n+1}$ and curvature $K \in \mathbb{R}$, where $\xi \in \mathbb{R}$, $x, y \in \mathbb{R}$:

$$\rho_K((\xi; x^T)^T) = \frac{x}{1 + \sqrt{|K|} \cdot \xi}$$

spaces	symbol	formula
Euclidean	$\mathbb{E}$	$\|\boldsymbol{x} - \boldsymbol{y}\|_2$
Spherical, Hyperbolic	$\mathbb{S}, \mathbb{H}$	$\frac{1}{\sqrt{\|K\|}} \cos_K^{-1}(\|K\| \langle \boldsymbol{x}, \boldsymbol{y}\rangle_K)$
Projected Hypersphere, Poincarré	$\mathbb{D}, \mathbb{P}$	$\frac{2}{\sqrt{\|K\|}} \tan_K^{-1}(\sqrt{K}\| - \boldsymbol{x} \oplus_K \boldsymbol{y}\|_2)$

TABLE I: Distance functions. The detailed notations can be found in Table II

notation	formula
$\boldsymbol{x} \oplus_K \boldsymbol{y}$	$\frac{(1-2K\langle \boldsymbol{x},\boldsymbol{y}\rangle_2 - K\|\boldsymbol{y}\|_2^2)\boldsymbol{x} + (1+K\|\boldsymbol{x}\|_2^2)\boldsymbol{y}}{1-2K\langle \boldsymbol{x},\boldsymbol{y}\rangle_2 + K^2\|x\|_2^2\|y\|_2^2}$
$\langle \boldsymbol{x}, \boldsymbol{y}\rangle_K$	$\begin{cases} \sum_{i=1}^p x_i \cdot y_i & \text{if } K > 0 \\ -x_1 \cdot y_1 + \sum_{i=2}^p x_i \cdot y_i & \text{if } K < 0 \end{cases}$
$\cos_K$	$\begin{cases} \cos & \text{if } K > 0 \\ \cosh & \text{if } K < 0 \end{cases}$
$\tan_K$	$\begin{cases} \tan & \text{if } K > 0 \\ \tanh & \text{if } K < 0 \end{cases}$

TABLE II: Formula of Möbius addition, $\oplus_K$, curvature-aware scalar product, $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_K$, cosine, $\cos_K$, and sine, $\sin_K$, of $x, y \in \mathbb{R}^p$

One can see that the distance in space $\mathbb{S}$ and $\mathbb{H}$ is not defined for a curvature approaching 0:

We have that $|K|\langle x, y\rangle_K \to 0$ as K approaches 0. And, as $t \to 0$, $cos_K^{-1}(t) \to \pi/2$ if $K > 0$ or 0 if $K < 0$. Then, as $K \to 0$, $\frac{1}{\sqrt{|K|}} \cos_K^{-1}(|K| \langle \boldsymbol{x}, \boldsymbol{y}\rangle_K) \to \infty$ and the distance is thus not defined.

APPENDIX B
EXPERIMENT SET-UP AND RESULTS

hyperparameter	default value
Batch Size	100
Learning Rate	0.001
Epochs	500
Warmup	100
Lookahead	50
Hidden Units	400
Beta Start	1
Beta End	1
Beta End Epoch	1

TABLE III: Default hyperparameters used during all experiments

dataset	cells $\times$ features	running time
ADIPOSE	1378, 1351	$<$ 10 minutes
RGC	35699, 1809	$<$ 6 hours
CELEGANS	86024, 2766	$<$ 24 hours
UC-EPI	64457, 1361	$<$ 24 hours

TABLE IV: Running times on an average, modern laptop

id	model	fixed curvature	universal	accuracy (%)	std. deviation (x10^{-2})
$m001$	$\mathbb{E}2$	TRUE	FALSE	85.2	1.41
$m002$	$\mathbb{E}6$	TRUE	FALSE	90.6	1.28
$m003$	$\mathbb{E}12$	TRUE	FALSE	89.6	1.59
$m004$	$\mathbb{E}24$	TRUE	FALSE	89.3	1.14
$m005$	$\mathbb{H}2$	TRUE	FALSE	84.8	1.23
$m006$	$\mathbb{H}6$	TRUE	FALSE	90.0	1.48
$m007$	$\mathbb{H}12$	TRUE	FALSE	89.0	1.34
$m008$	$\mathbb{H}24$	TRUE	FALSE	86.5	1.31
$m009$	$\mathbb{S}2$	TRUE	FALSE	85.2	1.22
$m010$	$\mathbb{S}6$	TRUE	FALSE	90.1	0.96
$m011$	$\mathbb{S}12$	TRUE	FALSE	89.8	1.09
$m012$	$\mathbb{S}24$	TRUE	FALSE	89.0	0.86
$m013$	$\mathbb{E}2 \times \mathbb{H}2 \times \mathbb{S}2$	TRUE	FALSE	89.2	1.28
$m014$	$\mathbb{E}4 \times \mathbb{H}4 \times \mathbb{S}4$	TRUE	FALSE	88.5	1.40
$m015$	$\mathbb{E}8 \times \mathbb{H}8 \times \mathbb{S}8$	TRUE	FALSE	88.3	0.80
$m016$	$\mathbb{S}12$	FALSE	FALSE	90.2	0.80
$m017$	$\mathbb{H}12$	FALSE	FALSE	89.7	1.20
$m018$	$\mathbb{E}4 \times \mathbb{H}4 \times 4$	FALSE	FALSE	88.9	1.33
$m019$	$\mathbb{U}4 \times \mathbb{U}4 \times \mathbb{U}4$	FALSE	TRUE	89.8	1.21

TABLE V: Experiment accuracies for ADIPOSE Dataset

id	model	fixed curvature	universal	accuracy (%)	std. deviation (x10^{-2})
$m020$	$\mathbb{E}2$	TRUE	FALSE	38.9	2.15
$m021$	$\mathbb{E}6$	TRUE	FALSE	77.2	1.54
$m022$	$\mathbb{E}12$	TRUE	FALSE	82.6	1.39
$m023$	$\mathbb{E}24$	TRUE	FALSE	81.7	1.68
$m024$	$\mathbb{H}2$	TRUE	FALSE	40.3	1.76
$m025$	$\mathbb{H}6$	TRUE	FALSE	83.1	1.58
$m026$	$\mathbb{H}12$	TRUE	FALSE	85.0	1.78
$m027$	$\mathbb{H}24$	TRUE	FALSE	79.7	1.60
$m028$	$\mathbb{S}2$	TRUE	FALSE	44.6	2.09
$m029$	$\mathbb{S}6$	TRUE	FALSE	84.8	1.34
$m030$	$\mathbb{S}12$	TRUE	FALSE	86.7	1.25
$m031$	$\mathbb{S}24$	TRUE	FALSE	85.6	1.30
$m032$	$\mathbb{E}2 \times \mathbb{H}2 \times \mathbb{S}2$	TRUE	FALSE	75.1	4.02
$m033$	$\mathbb{E}4 \times \mathbb{H}4 \times \mathbb{S}4$	TRUE	FALSE	72.5	3.28
$m034$	$\mathbb{E}8 \times \mathbb{H}8 \times \mathbb{S}8$	TRUE	FALSE	67.0	2.59
$m035$	$\mathbb{S}12$	FALSE	FALSE	85.7	1.32
$m036$	$\mathbb{H}12$	FALSE	FALSE	85.8	1.26
$m037$	$\mathbb{E}4 \times \mathbb{H}4 \times \mathbb{S}4$	FALSE	FALSE	77.5	2.89
$m038$	$\mathbb{U}4 \times \mathbb{U}4 \times \mathbb{U}4$	TRUE	TRUE	82.6	1.90

TABLE VI: Experiment accuracies for RGC Dataset

id	model	fixed curvature	universal	accuracy (%)	std. deviation (x10^{-2})
$m039$	$\mathbb{E}2$	TRUE	FALSE	67.7	1.78
$m040$	$\mathbb{E}6$	TRUE	FALSE	78.6	1.74
$m041$	$\mathbb{E}12$	TRUE	FALSE	80.2	1.23
$m042$	$\mathbb{E}24$	TRUE	FALSE	72.0	1.77
$m043$	$\mathbb{H}2$	TRUE	FALSE	70.1	1.66
$m044$	$\mathbb{H}6$	TRUE	FALSE	79.1	1.53
$m045$	$\mathbb{H}12$	TRUE	FALSE	78.4	1.61
$m046$	$\mathbb{H}24$	TRUE	FALSE	70.6	2.12
$m047$	$\mathbb{S}2$	TRUE	FALSE	70.2	1.69
$m048$	$\mathbb{S}6$	TRUE	FALSE	80.1	1.22
$m049$	$\mathbb{S}12$	TRUE	FALSE	81.8	1.33
$m050$	$\mathbb{S}24$	TRUE	FALSE	78.6	1.51
$m051$	$\mathbb{E}2 \times \mathbb{H}2 \times \mathbb{S}2$	TRUE	FALSE	78.0	1.51
$m052$	$\mathbb{E}4 \times \mathbb{H}4 \times \mathbb{S}4$	TRUE	FALSE	75.4	1.67
$m053$	$\mathbb{E}8 \times \mathbb{H}8 \times \mathbb{S}8$	TRUE	FALSE	61.7	3.54
$m054$	$\mathbb{S}12$	FALSE	FALSE	81.3	1.36
$m055$	$\mathbb{H}12$	FALSE	FALSE	80.6	1.22
$m056$	$\mathbb{E}4 \times \mathbb{H}4 \times \mathbb{S}4$	FALSE	FALSE	76.5	1.67
$m057$	$\mathbb{U}4 \times \mathbb{U}4 \times \mathbb{U}4$	TRUE	TRUE	79.1	1.57

TABLE VII: Experiment accuracies for CELEGANS dataset

id	model	fixed curvature	universal	accuracy (%)	std. deviation (x10^{-2})
$m058$	$\mathbb{E}2$	TRUE	FALSE	57.0	1.98
$m059$	$\mathbb{E}6$	TRUE	FALSE	70.4	1.32
$m060$	$\mathbb{E}12$	TRUE	FALSE	69.7	1.74
$m061$	$\mathbb{E}24$	TRUE	FALSE	63.9	1.87
$m062$	$\mathbb{H}2$	TRUE	FALSE	58.4	2.02
$m063$	$\mathbb{H}6$	TRUE	FALSE	7.11	1.49
$m064$	$\mathbb{H}12$	TRUE	FALSE	69.2	2.00
$m065$	$\mathbb{H}24$	TRUE	FALSE	63.6	2.27
$m066$	$\mathbb{S}2$	TRUE	FALSE	59.4	2.23
$m067$	$\mathbb{S}6$	TRUE	FALSE	70.8	1.58
$m068$	$\mathbb{S}12$	TRUE	FALSE	69.8	1.84
$m069$	$\mathbb{S}24$	TRUE	FALSE	66.6	2.30
$m070$	$\mathbb{E}2 \times \mathbb{H}2 \times \mathbb{S}2$	TRUE	FALSE	66.6	1.93
$m071$	$\mathbb{E}4 \times \mathbb{H}4 \times \mathbb{S}4$	TRUE	FALSE	63.5	1.80
$m072$	$\mathbb{E}8 \times \mathbb{H}8 \times \mathbb{S}8$	TRUE	FALSE	52.4	3.77
$m073$	$\mathbb{S}12$	FALSE	FALSE	69.8	1.78
$m074$	$\mathbb{H}12$	FALSE	FALSE	70.1	1.93
$m075$	$\mathbb{E}4 \times \mathbb{H}4 \times \mathbb{S}4$	FALSE	FALSE	63.8	1.68
$m076$	$\mathbb{U}4 \times \mathbb{U}4 \times \mathbb{U}4$	TRUE	TRUE	69.3	1.66

TABLE VIII: Experiment accuracies for UC-EPI dataset

dataset	dimension	accuracy (%)	std. deviation (x10^{-2})
ADIPOSE	2	85.0	1.27
ADIPOSE	6	90.2	1.25
ADIPOSE	12	89.5	1.36
ADIPOSE	24	88.3	1.68
CELEGANS	2	69.3	2.07
CELEGANS	6	79.3	1.63
CELEGANS	12	80.1	1.96
CELEGANS	24	73.7	3.93
RGC	2	41.2	3.15
RGC	6	81.6	3.46
RGC	12	84.7	2.24
RGC	24	82.3	2.89
UC − EPI	2	58.3	2.28
UC − EPI	6	70.8	1.48
UC − EPI	12	69.6	1.87
UC − EPI	24	64.7	2.54

TABLE IX: Experiment accuracies per dataset and Latent Space Dimension, for fixed-curved and non-mixed models

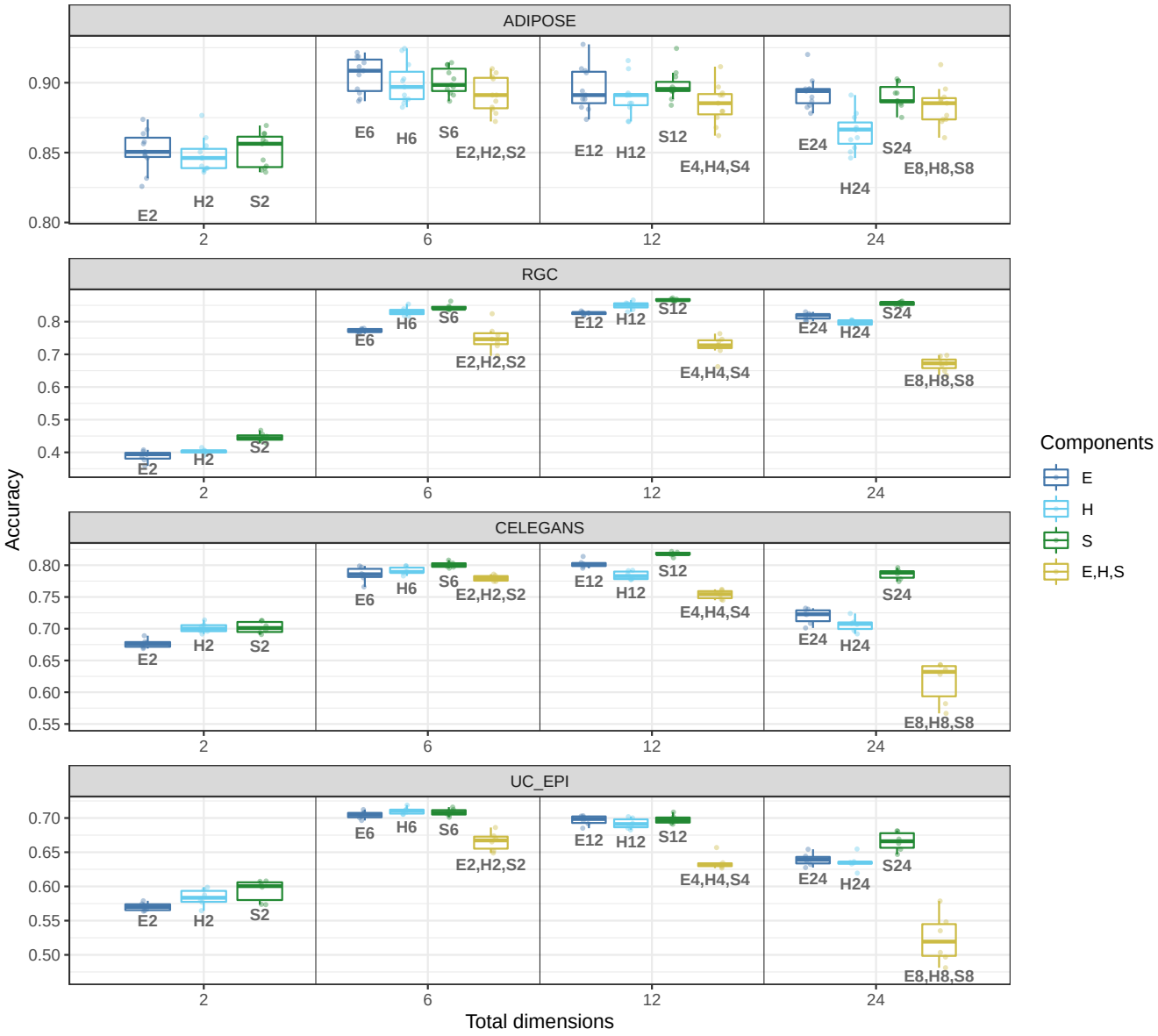

Fig. 1: Accuracies for models with fixed curvature

The results are shown for all four datasets (rows) and for all models which have fixed curvature. The models are arranged by the total number of dimensions in the latent space (columns).

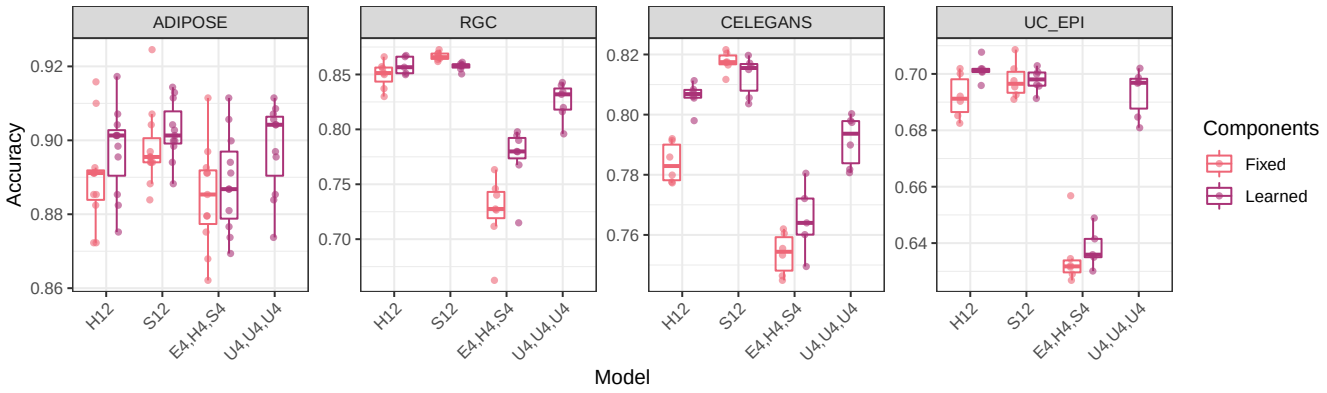

Fig. 2: Comparing accuracies for models with fixed versus learned curvature
The accuracies for models with fixed curvature and learned curvature are shown in different colors. The results for different datasets are arranged in columns.

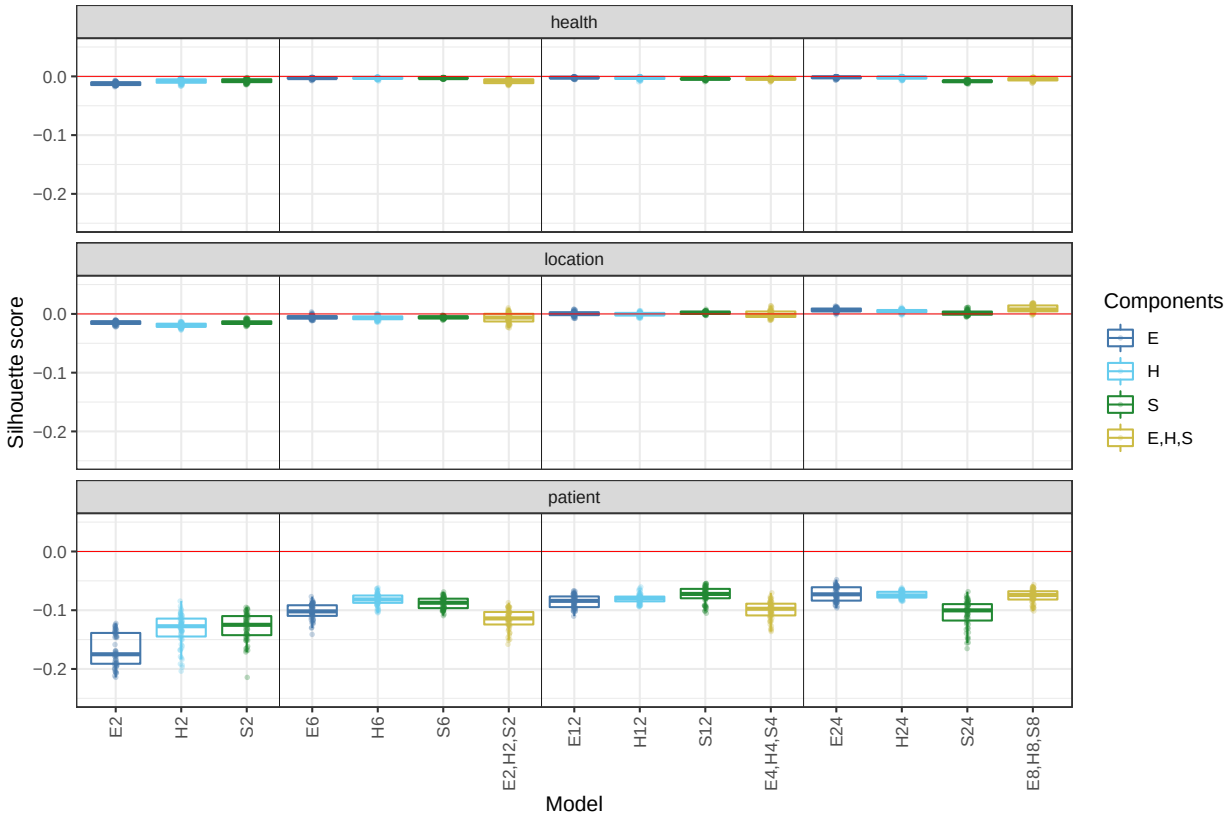

Fig. 3: Silhouette scores for batch effects in UC_EPI dataset
The results are shown for all models with fixed curvature. The models are arranged by the total number of dimensions in the latent space. The silhouette scores are shown for 3 different batch effects: health status ($n = 3$ classes), anatomical location ($n = 2$ classes) and patient ($n = 30$ classes). The silhouette scores were computed using random samples of 5000 cells represented in the latent space ($n = 10$ random samples per model). A clustering of cells by batch effect would have been indicated by positive silhouette scores. Small or negative scores suggest that batch effects are successfully removed by scMVAE.