On the Role of Embeddings in Diffusion-based Generation of scRNA-seq Data

Martin Rackl[1], Felix Fuchs[2], and Andreas Hotho[1]

[1] CAIDAS, University Würzburg, Germany
[2] University Würzburg, Germany
martin.rackl@informatik.uni-wuerzburg.de
felix.fuchs@stud-mail.uni-wuerzburg.de
hotho@informatik.uni-wuerzburg.de

Abstract. Single-cell RNA sequencing (scRNA-seq) enables fine-grained insight into the heterogeneity of tissues and cellular responses. However, the limited availability of high-quality datasets and the inherent noise in scRNA-seq measurements hinder downstream analyses. Generative models, particularly diffusion models, offer a promising approach to synthesizing realistic scRNA-seq data. This work builds upon the scDiffusion framework and investigates the influence of various embedding strategies on diffusion-based generation. We study three trainable approaches: an autoencoder as in scDiffusion, an scANVI model, and an scTAG model. Further, we investigate a feature selection approach using highly variable genes (HVGs). For the guided diffusion, we use a four-layer MLP as well as an scANVI-based classifier to explore conditional generation. We show that the scANVI model produces the top-performing embedding, and the diffusion model trained on this embedding yields the most realistic data, with a class distribution closely resembling real data. We also show that the used embedding metrics are not sufficient for deciding which embedding is best suited for training another model. These findings highlight the importance of representation choice when training a diffusion model to generate new scRNA-seq data.

Keywords: scRNA-seq · Diffusion Models · Representation Learning

1 Introduction

Single-cell RNA sequencing (scRNA-seq) has emerged as a powerful technology that enables researchers to examine gene expression at the resolution of individual cells. This technology has revolutionized fields such as cancer biology, immunology, and developmental biology by uncovering cellular heterogeneity that bulk RNA sequencing cannot resolve [9, 20]. However, the utility of scRNA-seq is limited by its high cost, technical noise (e.g., dropout events, batch effects), and the labor-intensive nature of annotation [9, 11]. These limitations make it difficult to obtain sufficiently large and well-annotated datasets, particularly for rare cell types.

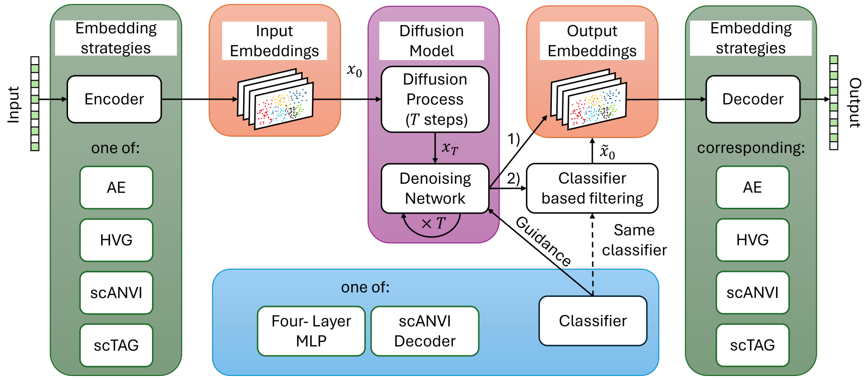

Fig. 1: Adapted scDiffusion framework [12]. Gene expression profiles are embedded using four strategies: Autoencoder, HVGs, scANVI, and scTAG. Each embedding used as training input for a diffusion model. Path 1) shows the unconditional sampling, and path 2) conditional sampling. Classifier guidance [2] is applied using either a four-layer MLP or an scANVI decoder, followed by an additional filtering step based on the classifier predictions.

Recent advances in generative machine learning have opened up new opportunities to address these challenges by generating realistic scRNA-seq data. Generative models can augment training data for supervised learning, aid in benchmarking methods, and support simulation of hypothetical experiments. Among these, deep generative models such as variational autoencoders (VAEs) [8, 6], generative adversarial networks (GANs) [3, 21], and denoising diffusion probabilistic models [7, 19] have been explored for this purpose. The scDiffusion framework [12] introduced a diffusion-based approach specifically tailored for scRNA-seq data and has shown promising results in the unconditional and conditional generation of scRNA-seq samples. Unconditional generation refers to generating samples without aiming at a specific target class, differing from conditional generation, where we want to generate samples of a specific class.

While scDiffusion provides a strong foundation, several questions remain open: How do different latent representations affect the training of a diffusion model and hence the quality of generated samples? Can feature-selection approaches, such as using only a subset of genes, e.g., highly variable genes (HVG), perform comparably as training input for a diffusion model? How are the cell types distributed if unconditionally generated? This study utilizes an adapted scDiffusion framework to address the posed questions (see fig. 1).

Our main contributions can be summarized as follows:

(1) Evaluation of the influence of training a diffusion model on different input embeddings created by the following embedding strategies: the original autoencoder of [12] as a baseline, scANVI [17], scTAG [18], and highly variable genes (HVG) as a feature-selection approach. **(2)** Analysis of the relationship between the quality of the input embeddings, derived via different clustering metrics, and the quality of the generated scRNA-seq data (Output in fig. 1). **(3)** Investigation of two different classifiers, a four-layer MLP and a decoder of the scANVI model [17], for guided diffusion.

2 Related Work

Recent work on scRNA-seq data synthesis includes statistical simulators such as SPARSim [1] and scDesign3 [15], as well as deep generative models like GRouNdGAN [21] with a GAN architecture and scDiffusion [12] with a diffusion model, as mentioned in the introduction.

In this work, we build on the work of scDiffusion and expand their model. The scDiffusion model employs a diffusion-based architecture that generates scRNA-seq data by first training an autoencoder model and then using the embedding created by this autoencoder to train a diffusion model. As an autoencoder, the pre-trained foundation model SCimilarity [5] was used and then fine-tuned. The diffusion model itself follows a denoising diffusion probabilistic model framework and operates in the learned embedding space. Additionally, they may utilize a distinct classifier, specifically a four-layer MLP, to perform guided diffusion.

Beyond data generation, representation learning is an essential step in analyzing scRNA-seq data because they naturally come with several problems, e.g, technical noise, biological noise, high dimensionality, dropouts, and others. Via representation learning techniques, one can transform the data to a much more suitable form for further downstream analysis [4]. VAE-based approaches, e.g., scVI [10], scANVI [17], and graph-based models like scTAG [18] are commonly used. These models map high-dimensional gene expression profiles into lower-dimensional latent spaces that preserve biological variation while reducing technical noise. Additionally, feature-selection in the gene space using highly variable genes (HVGs) provides interpretable and robust alternatives, especially when labeled data is scarce or model interpretability is important.

We investigate the impact of these representation learning methods on synthesis performance. To the best of our knowledge, we are the first to explore the relation between the quality of input embeddings and the quality of conditionally and unconditionally generated data.

3 Methodology

We employ the scDiffusion framework [12] and exchange key components to investigate their influence on the generated samples. This can be seen in fig. 1. The inputs and outputs are gene expression profiles of the original and the generated data, respectively. To examine how various input embeddings affect a diffusion model's performance, we utilized four different embedding strategies: **(1)** The original autoencoder from the scDiffusion paper as a baseline, where they used a fine-tuned SCimilarity model [5]. **(2)** Highly variable genes (HVGs), as a subset of all genes, as these are often used for training models like scANVI and should, in theory, contain most of the necessary information. Using HVGs only can be viewed as an embedding of the data in the gene space with reduced dimensionality and hence could be a useful representation of the data. **(3)** The embedding from scANVI [17], a supervised variational autoencoder, trained on HVGs. scANVI extends scVI [10] by introducing semi-supervised learning for cell-type annotation. It incorporates labeled and unlabeled data during training, enabling both representation learning and classification within a unified probabilistic framework. **(4)** The embedding generated by scTAG [18], which is an unsupervised model using a graph neural network that uses topological structures of cell graphs derived from scRNA-seq data. It leverages relationships between similar cells through message passing, allowing the model to learn biologically meaningful embeddings from the cellular neighborhood context.

With these four strategies, we create four different input embeddings, which we use to train a diffusion model. Therefore, we employed the original diffusion model from scDiffusion [12]. Inside this diffusion model is a diffusion process where noise is added to the input x_0 in $T = 1000$ steps to obtain the noisy sample x_T. Then, a denoising network is trained to invert this noise in T steps to generate a new sample $\tilde{x}_0$. Sampling new data from the trained diffusion model can be done in two ways: **(1)** Unconditional sampling is performed without any guidance. A random noise sample x_{noise} is given to the diffusion model, which then generates a denoised point in T denoising steps. **(2)** Conditional sample generation, where we employ the classifier guidance method [2], which necessitates the use of a classifier. The core idea is to use the gradient of the classifier during the denoising steps of the diffusion model to guide the denoising towards the target class.

We utilize two distinct classifiers: **(1)** the original classifier as detailed in [12], which is a four-layer MLP, and **(2)** the decoder from the scANVI model [17].

In generating conditional synthetic data, we incorporated a filtering process using the same classifier applied in guided diffusion. A generated sample $\tilde{x}_0$ is retained only if the classifier successfully predicts the target class; otherwise, the sampling is redone.

The embedding models and classifiers are trained independently. The scANVI classifier is completely separate from the scANVI model used for creating the input embeddings. The diffusion model is trained with the embeddings generated by our four approaches.

4 Experimental Setup

In this section, we will briefly introduce the datasets used and explain how we trained the different models.

4.1 Datasets

We used two common datasets for evaluation:

Tabula Muris [14] contains nearly $100,000$ cells across 20 different organs and tissues from mice (Mus Musculus). After processing the data with Cell-Ranger from the 10x Genomics platform, $55,656$ of cells remain after quality control, which are divided into 56 different cell types. In addition to the cell annotation, there are metadata such as mouse id, sex, etc. given in the dataset. The annotation was achieved by experts using marker genes and unbiased clustering.

PBMC68k [20] dataset was created to study immune populations within peripheral blood mononuclear cells (PBMC) and includes $68,579$ samples, which were also sequenced by a droplet-based method. The data of this dataset comes from a healthy donor (Donor A). The PBMCs are an important component of the immune system and help to defend against infections. In this dataset, the cells are separated into 11 different cell types. There are no further annotations present in this dataset. Some of the cells in this dataset have overlapping functions, which makes it very hard to categorize them [20]. Therefore, this dataset is considered to be hard to classify.

4.2 Model Training

Embedding Model Training As a baseline, we use the same autoencoder architecture as SCimilarity [5], but we train the autoencoder from scratch, as well as the other models scANVI and scTAG. The HVGs were obtained using the scanpy package [16]. All embedding models were trained using the hyperparameters from table A.3 in the appendix. The dataset was also split in a train and test set with a ratio of 80% to 20%. The hyperparameters are taken from [12].
Diffusion Model Training For each of the four input embeddings we trained a diffusion model with the same architecture to ensure comparability. The training hyperparameters are detailed in in appendix in table A.4.
Classifier Training We use two classifiers: **(a)** a four-layer MLP of the original implementation, with SiLU as activation functions and a dropout of 0.1, and **(b)** the decoder component of an scANVI model. Both classifiers are trained with incorporated noise of the diffusion process up to time step $T/2$. Earlier stages of the denoised sample are considered too noisy. We also used early stopping with a patience of 5 epochs for training the classifier. Detailed information on the used hyperparameters is provided in table A.5 in the appendix.

4.3 Conditional and Unconditional Sampling

For the unconditional sampling, we generated as many samples as in the original dataset are present, using the four diffusion models trained with different input embeddings (see section 3). We do not give any constraints for the sample generation and evaluate how distinguishable the generated data are from the original ones. Further, we investigate to what extent the generated data represent the same class distribution as the original data.

Conditional sampling is achieved by combining each diffusion model, trained with distinct embedding strategies, with a classifier, resulting in eight different combinations of embedding strategy and classifier. For each class in the initial dataset, we produce an equivalent number of samples as present in the original dataset. Afterwards, we again investigate how hard it is to differentiate between real and generated data. For this, we use a kNN-classifier with $k = 5$.

Table 1: Clustering metrics for different embedding strategies on two datasets.

Embedding	Tabula Muris				PBMC68k			
	ARI ↑	NMI ↑	SS ↑	HMG ↑	ARI ↑	NMI ↑	SS ↑	HMG ↑
AE	0.4099	0.7704	0.1191	0.8621	0.2797	0.4584	0.0855	0.4889
HVG	0.6656	0.7938	-0.0658	0.7244	0.1681	0.4005	-0.0281	0.2803
scANVI	**0.9777**	**0.9698**	0.0915	**0.9683**	**0.4655**	**0.5436**	0.0521	**0.5106**
scTAG	0.2593	0.4639	**0.3097**	0.4376	0.0790	0.1424	**0.4722**	0.1187

5 Results

In the following sections, we will present the results of the four embedding models and the two classifiers, evaluating their performance and how varying representation strategies impacted both unconditional and conditional generation of new scRNA-seq data.

5.1 Embedding Model Comparison

The performance of the different embedding strategies was assessed using four clustering metrics: Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Silhouette Score (SS), and Homogeneity Score (HMG). We use the implementation of these metrics of the scikit-learn library [13]. Arrows indicate if higher values are better. The results on both the Tabula Muris and PBMC68k datasets are summarized in table 1.

In the Tabula Muris dataset, scANVI embeddings outperformed in most metrics, achieving top scores: ARI at 0.9777, NMI at 0.9698, and HMG at 0.9683, indicating a significant agreement between scANVI-derived cell clusters and the ground truth labels. The baseline autoencoder showed results comparable to the HVG method. Among the four evaluated strategies, the unsupervised scTAG presented the lowest ARI, NMI, and HMG values, yet it showed superior results in SS, suggesting scTAG's capacity to form distinct clusters unrelated to actual cell types. Therefore, it is crucial to consider SS along with other metrics. This analysis strongly suggests scANVI as the best embedding method in our experiments.

In the PBMC68k dataset, scANVI again demonstrates best outcomes, achieving an ARI of 0.4655, NMI of 0.5436, and HMG of 0.5106, which are, as anticipated in section 4, worse than the Tabula Muris results. The AE is again comparable to the HVG method and scTAG is good in forming clusters but these are mostly unrelated to the actual cell types. The low ARI, NMI, and HMG values indicate poor alignment with the ground truth labels, while the high SS reflects well-formed clusters.

5.2 Unconditional Sampling

To evaluate the quality of generated scRNA-seq data based on different input embeddings to a diffusion model, we employ a set of complementary metrics. We

Table 2: Comparison of generation methods across multiple evaluation metrics.

Embedding	Tabula Muris						
	PCC ↑	SCC ↑	MMD ↓	iLISI ↑	RF AUC ↓	KL-Div ↓	C-Dist ↓
AE	0.9993	0.9916	**0.1214**	0.8551	**0.6900**	0.6703	8.785
HVG	0.9990	0.9897	0.8165	0.0000	0.9961	2.3789	15.06
scANVI	**0.9994**	**0.9925**	0.1231	**0.8703**	0.7413	**0.1898**	**8.317**
scTAG	0.9978	0.9898	0.1427	0.1956	0.9499	0.5312	9.740

Embedding	PBMC68k						
	PCC ↑	SCC ↑	MMD ↓	iLISI ↑	RF AUC ↓	KL-Div ↓	C-Dist ↓
AE	0.9996	0.9732	**0.0314**	**0.8853**	**0.9877**	1.722	**6.159**
HVG	0.9996	0.9637	0.4637	0.0000	0.9999	1.098	10.65
scANVI	**0.9997**	**0.9737**	0.1792	0.6817	0.9999	**0.001**	6.186
scTAG	0.9687	0.8564	1.1343	0.0000	1.0000	1.177	12.94

use the scikit-learn implementation [13] of the following metrics, except for the C-Dist metric, which we implemented by ourselves. Pearson (PCC) and Spearman correlation coefficients (SCC) measure the linear and rank-based association, respectively, between gene expression profiles in the original and generated data, capturing similarity in both magnitude and ordering. Therefore, we first calculated the mean over all cells for each gene and afterwards calculated the PCC and SCC, resulting in a bulk-level gene profile comparison. This approach captures population-level similarity but does not assess cell-level matching or heterogeneity. Maximum Mean Discrepancy (MMD) quantifies the distributional similarity between datasets in a reproducing kernel Hilbert space, assessing how well the global structure is preserved. iLISI (integration Local Inverse Simpson's Index) evaluates the degree of mixing between the real and a generated dataset, with higher scores indicating better integration. The Random Forest AUC (RF AUC) assesses how distinguishable real and generated cells are by training a classifier to discriminate them, which means that a lower AUC suggests higher realism. The Kullback–Leibler Divergence (KL-Div) measures how much the distribution of generated data diverges from the real data distribution, with lower values indicating better fidelity. For this we first calculated the class distributions, see figs. 2 and 3, and then calculate the KL-divergence. Finally, the C-Dist metric computes the Euclidean distance between the centroids of each cell type in the real and generated datasets, averaged across all classes, to assess how well the generative model preserves the spatial structure of cell identities in the embedding space, where a lower value indicates better performance.

Initially, we validated our scDiffusion reimplementation by aligning our findings with the original results. Our SCC, iLISI, and RF AUC outcomes were comparable, yet our MMD score was inferior. This could be due to the exclusion of the pre-trained SCimilarity model.

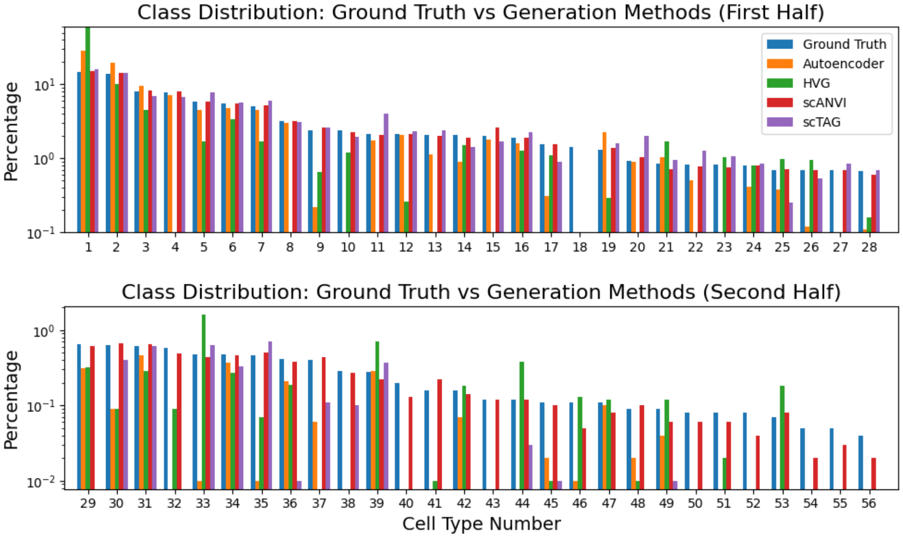

Fig. 2: Comparison of class distributions of Tabula Muris. The upper figure shows the first 28 cell types and the lower figure shows the last 28 cell types. The mapping from number to cell type can be seen in table A.1 in the appendix.

Across both datasets, scANVI performed consistently the best. The PCC and SCC are quite similar for all methods and datasets except for scTAG with PBMC68k, where SCC is considerably lower. The diffusion models inferred the overall gene expression pattern from the four embeddings. The MMD is quite small for AE, scANVI, and scTAG, indicating distributional similarity, whereas the HVG embedding performed quite poorly in comparison. The iLISI value shows that the generated samples from the scANVI diffusion model and the AE diffusion model are well integrated, meaning they exhibit less systematic deviation, in contrast to scTAG. For the PBMC68k dataset, the generated samples are even better integrated than the ones from the scANVI method. For the HVG method we received a value very close to 0, meaning there is no integration of the real and simulated data. For the Tabula Muris dataset we can further see that the samples generated from the AE and scANVI method are harder to distinguish from the real data than the ones from the other two methods. For the PBMC68k dataset all generated samples can easily be distinguished from the real ones. The KL-Div indicates very strong matching of the class distributions for the real samples and the generated samples from the scANVI method. This can also be seen in figs. 2 and 3. The plot shows that none of the methods generated samples for class 18, as the original dataset contains unlabeled samples, which are grouped in this class and labeled as Unknown. This is because we use the trained decoder to get to the gene-space, where we use a trained classifier to predict the label, but in this case, it can not predict the correct label, because it

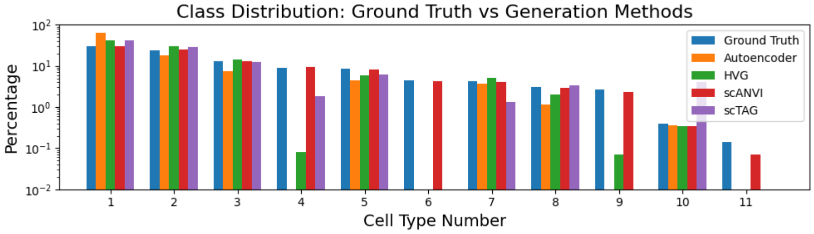

Fig. 3: Comparison of class distributions of the PBMC68k dataset. The mapping can be seen in table A.2 in the appendix.

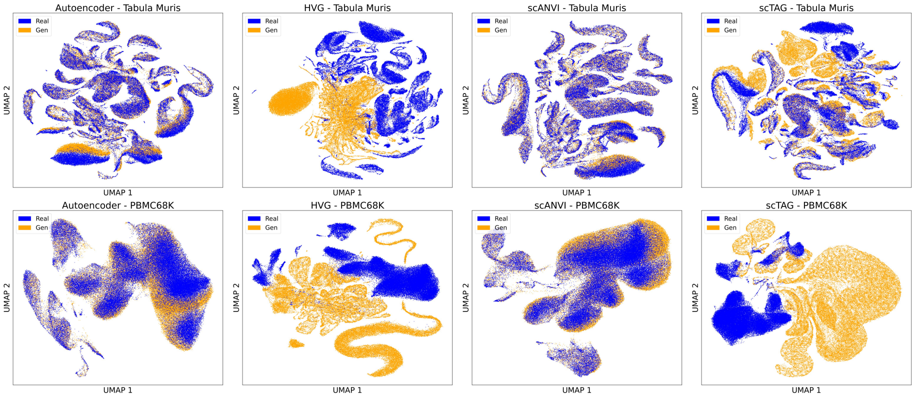

Fig. 4: UMAP plots of real (blue) and generated (orange) data for Tabula Muris and PBMC68k using different embeddings for training diffusion models. The used methods are from left to right: Base-autoencoder, HVG, scANVI, scTAG.

had no labels to train on. Lastly, the C-Dist metric tells us that the class centers of the AE and scANVI methods are closest to the centers of the real data in the latent space.

In conclusion, we can assume that the scANVI method's embedding produces the most stable and distributionally similar outcomes. This also matches with the UMAP plots in fig. 4. The plots show the results of the unconditional generation together with the real samples for the four methods for Tabula Muris and PBMC68k. On the left, the results for the baseline autoencoder method are shown. These can be compared to the UMAP plots of the scDiffusion paper, which show a very similar behaviour. Most parts are aligned quite well, but there are some areas where a shift of the generated data can be seen. The next image shows the results for the HVG method, where the generated data have hardly any overlaps with the real data. The third image shows scANVI approach results, where we can see the highest overlap of generated and real samples. The last image shows the scTAG approach, where one can see some overlap, but large parts of the generated data are not located where the real data are.

Table 3: Performance comparison of a binary kNN classifier in distinguishing real from generated data. The closer the values are to 0.5, the harder it was for the kNN classifier to distinguish real and generated data. All values are macro averages over all classes.

		Tabula Muris				PBMC68k			
		AE	HVG	scANVI	scTAG	AE	HVG	scANVI	scTAG
Base	Acc	**0.66**$_{(\pm 0.11)}$	0.95$_{(\pm 0.12)}$	0.70$_{(\pm 0.12)}$	0.75$_{(\pm 0.14)}$	0.55$_{(\pm 0.06)}$	0.88$_{(\pm 0.13)}$	**0.52**$_{(\pm 0.08)}$	0.86$_{(\pm 0.16)}$
	AUC	**0.78**$_{(\pm 0.15)}$	0.96$_{(\pm 0.11)}$	0.84$_{(\pm 0.14)}$	0.83$_{(\pm 0.16)}$	0.66$_{(\pm 0.08)}$	0.96$_{(\pm 0.08)}$	**0.59**$_{(\pm 0.09)}$	0.92$_{(\pm 0.10)}$
scANVI	Acc	0.71$_{(\pm 0.13)}$	0.95$_{(\pm 0.12)}$	**0.70**$_{(\pm 0.14)}$	0.81$_{(\pm 0.14)}$	0.57$_{(\pm 0.09)}$	0.87$_{(\pm 0.13)}$	**0.51**$_{(\pm 0.03)}$	0.90$_{(\pm 0.14)}$
	AUC	0.84$_{(\pm 0.15)}$	0.96$_{(\pm 0.13)}$	**0.83**$_{(\pm 0.15)}$	0.90$_{(\pm 0.15)}$	0.71$_{(\pm 0.12)}$	0.95$_{(\pm 0.07)}$	**0.58**$_{(\pm 0.08)}$	0.95$_{(\pm 0.07)}$

5.3 Conditional Sampling

As explained in section 4.3, we used a filtering mechanism where only if the classifier correctly labeled the generated data point with the target label, the sample was kept. Because of this, we can not simply use our classifier to evaluate the generated samples. Therefore, we used a kNN classifier with $k = 5$, which is trained to distinguish between real and generated data, meaning that accuracy values around 0.5 show that the kNN classifier can not differentiate between real and generated data. Moreover, to account for class imbalance and ensure robust evaluation, we computed the area under the curve (AUC). The reported values of table 3 are averages over all cell types. In order to get useful results and avoid the curse of dimensionality, we used a PCA before calculating the metrics. The results can be seen in table 3, where we can see that the representations from the autoencoder and scANVI performed the best. The four-layer MLP base classifier demonstrated a slightly superior performance to scANVI. For Tabula Muris and the autoencoder method, the base classifier outperformed the scANVI classifier, although their results overlap within standard deviation. Accuracy rates of 0.66% and 0.71% show the kNN classifier struggles to reliably differentiate real and generated data. The HVG results with accuracies of 0.95% and 0.96% show that the real and generated samples are easy to differentiate. The scANVI approach performed similar to the autoencoder and generated hard to distinguish samples. The results for scTag are worse than scANVI and the autoencoder results, particularly with the scANVI classifier. PBMC68k findings mirrored this trend, but the results of scTAG dropped to an accuracy of 0.86% and an AUC of 0.92%, which is a comparable result to using HVGs. As observed, while the clustering metrics in table 2 indicate the encoder has more difficulty with the PBMC68k dataset, sample generation is simpler for this dataset, possibly due to its fewer classes. In summary, our results indicate that the autoencoder and scANVI method generated the best target cell type samples, with minimal influence from the classifier. Additionally, using HVGs as embedding results in distinguishable samples, thus making HVGs an unsuitable strategy.

6 Conclusion

This paper explored the effect of different embeddings for training a diffusion model for data generation of scRNA-seq data. We investigated the quality of four different embedding strategies and evaluated their quality, where the scANVI approach yielded the best results. Then we compared the results of the unconditional generation of samples with four diffusion models, each trained on a different input embedding. While scANVI excelled in clustering metrics, surpassing the autoencoder, the performance of unconditional and conditional sample generation was similar, as can be seen in table 2. Moreover, despite the challenges in generating a good embedding with the PBMC68k dataset, the results from both unconditional and conditional sampling were better. This shows that the choice of the embedding significantly impacts the quality of the generated data, as taking only HVGs is impractical. However, initial tests like clustering metrics offer insights into the latent representation, but are not sufficient to decide which embedding will perform the best, as in our experiments with the autoencoder and scANVI approach.

This paper presented an initial exploration of how different embeddings affect the training of diffusion models for single-cell data generation. While our results demonstrate that the choice of embedding influences generation quality, our evaluation remained empirical. In particular, we observed that commonly used clustering metrics may not adequately reflect the quality of generated samples. This limitation highlights important directions for future research, including evaluating the role of embeddings in other generative architectures such as GANs. Moreover, it raises a fundamental question: can we identify metrics that reliably assess the usefulness of latent representations as input for generative models?

References

1. Baruzzo, G., Patuzzi, I., Di Camillo, B.: Sparsim single cell: a count data simulator for scrna-seq data. Bioinformatics **36**(5), 1468–1475 (10 2019), https://doi.org/10.1093/bioinformatics/btz752
2. Dhariwal, P., Nichol, A.: Diffusion Models Beat GANs on Image Synthesis (Jun 2021). https://doi.org/10.48550/arXiv.2105.05233
3. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (Oct 2020). https://doi.org/10.1145/3422622
4. Gunawan, I., Vafaee, F., Meijering, E., Lock, J.G.: An introduction to representation learning for single-cell data analysis. Cell Reports Methods **3**(8) (Aug 2023). https://doi.org/10.1016/j.crmeth.2023.100547
5. Heimberg, G., et al.: A cell atlas foundation model for scalable search of similar human cells. Nature **638**(8052), 1085–1094 (Feb 2025). https://doi.org/10.1038/s41586-024-08411-y
6. Heydari, A.A., Davalos, O.A., Zhao, L., Hoyer, K.K., Sindi, S.S.: ACTIVA: Realistic single-cell RNA-seq generation with automatic cell-type identification using introspective variational autoencoders. Bioinformatics **38**(8), 2194–2201 (Apr 2022). https://doi.org/10.1093/bioinformatics/btac095

7. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. In: Advances in Neural Information Processing Systems. vol. 33, pp. 6840–6851. Curran Associates, Inc. (2020)
8. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014), http://arxiv.org/abs/1312.6114
9. Kolodziejczyk, A.A., Kim, J.K., Svensson, V., Marioni, J.C., Teichmann, S.A.: The technology and biology of single-cell RNA sequencing. Molecular Cell **58**(4), 610–620 (May 2015). https://doi.org/10.1016/j.molcel.2015.04.005
10. Lopez, R., Regier, J., Cole, M.B., Jordan, M.I., Yosef, N.: Deep generative modeling for single-cell transcriptomics. Nature Methods **15**(12), 1053–1058 (Dec 2018). https://doi.org/10.1038/s41592-018-0229-2
11. Luecken, M.D., Theis, F.J.: Current best practices in single-cell RNA-seq analysis: A tutorial. Molecular Systems Biology **15**(6), e8746 (Jun 2019). https://doi.org/10.15252/msb.20188746
12. Luo, E., Hao, M., Wei, L., Zhang, X.: scdiffusion: conditional generation of high-quality single-cell data using diffusion model. Bioinformatics **40**(9), btae518 (08 2024). https://doi.org/10.1093/bioinformatics/btae518
13. Pedregosa, F., et al.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research **12**(85), 2825–2830 (2011), http://jmlr.org/papers/v12/pedregosa11a.html
14. Schaum, N., et al.: Single-cell transcriptomics of 20 mouse organs creates a tabula muris. Nature **562**(7727), 367–372 (Oct 2018), https://doi.org/10.1038/s41586-018-0590-4
15. Song, D., Wang, Q., Yan, G., Liu, T., Sun, T., Li, J.J.: scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics. Nature Biotechnology **42**(2), 247–252 (Feb 2024). https://doi.org/10.1038/s41587-023-01772-1
16. Wolf, F.A., Angerer, P., Theis, F.J.: SCANPY: Large-scale single-cell gene expression data analysis. Genome Biology **19**(1), 15 (Feb 2018). https://doi.org/10.1186/s13059-017-1382-0
17. Xu, C., Lopez, R., Mehlman, E., Regier, J., Jordan, M.I., Yosef, N.: Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models (2021), https://www.embopress.org/doi/abs/10.15252/msb.20209620
18. Yu, Z., Lu, Y., Wang, Y., Tang, F., Wong, K.C., Li, X.: ZINB-Based Graph Embedding Autoencoder for Single-Cell RNA-Seq Interpretations. Proceedings of the AAAI Conference on Artificial Intelligence **36**(4), 4671–4679 (Jun 2022). https://doi.org/10.1609/aaai.v36i4.20392
19. Zhang, T., Zhao, Z., Ren, J., Zhang, Z., Zhang, H., Wang, G.: cfDiffusion: Diffusion-based efficient generation of high quality scRNA-seq data with classifier-free guidance. Briefings in Bioinformatics **26**(1), bbaf071 (Jan 2025). https://doi.org/10.1093/bib/bbaf071
20. Zheng, G.X.Y., et al.: Massively parallel digital transcriptional profiling of single cells. Nature Communications **8**(1), 14049 (Jan 2017). https://doi.org/10.1038/ncomms14049
21. Zinati, Y., Takiddeen, A., Emad, A.: GRouNdGAN: GRN-guided simulation of single-cell RNA-seq data using causal generative adversarial networks. Nature Communications **15**(1), 4055 (May 2024). https://doi.org/10.1038/s41467-024-48516-6