scPerb: single-cell perturbation via style transfer-based variational autoencoder

Zijia Tang  
 Guangdong Experimental High School  
 Guangzhou, Guangdong, China  
 zijiatang2006@gmail.com

Qianqian Song\*  
 Department of Cancer Biology, Wake Forest University School of Medicine  
 Winston-Salem, NC, USA  
 qsong@wakehealth.edu

**ABSTRACT**

The accurate prediction of cellular responses to perturbations faces great challenges in computational biology. To address this challenge, some methodologies have been previously developed, including graph-based approaches, vector arithmetic, and neural networks. However, these methods either mix the perturbation-related variances with the cell-type-specific patterns or implicitly distinguish them within the black-box models. In this work, we introduce a novel framework, scPerb, to explicitly extract the perturbation-related variances and transfer them from control data to perturbed data. scPerb adopts the style transfer strategy by incorporating a style encoder into the architecture of variational autoencoder. Such style encoder accounts for the differences in the latent representations between control cells and perturbed cells, which allows scPerb to accurately predicting the gene expression data in perturbed cells. Through the comparisons with existing methods, scPerb presents improved performance and higher accuracy in predicting cellular responses to perturbations. Specifically, scPerb not only outperforms other methods across multiple datasets but also achieves superior values of 0.98, 0.98, and 0.96 on three benchmarking datasets.

**INTRODUCTION**

Single-cell RNA sequencing (scRNA-seq) is a revolutionary technology to profile heterogeneous gene expression in tissue samples [1-3]. This technology can measure transcripts in thousands of single cells from multiple biological samples under different conditions [4-8]. Such breakthrough technology has inspired the development of tailored computational tools such as cell type annotations [9-12], identification of pseudotime trajectories [13, 14], and rare cell type detection [15, 16], facilitating the biological insights into single-cell data [ref].

Although scRNA-seq technologies have led to a remarkable growth of single-cell data, it was still challenging to collect the matched pairs of control and stimulated samples for a particular stimulus or perturbation. As current databases comprise a wide variety of single-cell data collected from samples at normal conditions, there is critical need to leverage the existing data at normal conditions to generate and predict the single-cell data after certain perturbation. To achieve this, an accurate and robust approach needs to demonstrate its capabilities in revealing unique gene expression patterns across different tissues, different platforms, and limited data size.

Recent studies have used generative models, such as Generative Adversarial networks (GAN) [17] and Variational Auto-Encoders (VAE) [18], to fill up the missing pieces in perturbation tasks. Specifically, GAN introduced a generator to construct the “fake” but realistic data and trained a discriminator to adversarial determine whether the “fake” data was good enough. Such adversarial battle aimed to train a robust generator to infer high-quality data samples. However, the major drawback of GAN lies in the difficulty in balancing the adversarial training, leading to a useless collapsed generator that is very sensitive to the input data noise. sc-WGAN [19] transferred the more stable WGAN to the single-cell perturbation and st-GAN [20] introduced the idea of style transferring that transfer multiple styles determined by the users to the generator. On the other hand, VAE generates data by sampling from a multivariate Gaussian distribution and used an encoder to estimate the mean and variance of the Gaussian distribution components of the original distributions, then generate new data observations from the estimated distribution using variational inference. For example, scGEN [21] assumed a fixed linear gap between the control cells and the perturbed cells, calculated the latent difference from both datasets and predicted the perturbed cell response using latent representation from control cells and the perturbed cells. Conditional Variational Auto-Encoder (CVAE) [22] introduced more constraints to the neural network, allowing the end-users to generate desired reconstructed samples for customized demands.

In this work, we present a novel tool, i.e., scPerb, to predict single-cell gene expressions under specific conditions such as a dose [23], a treatment [24, 25], or a modification of genes [26-28] (Fig.1). Given two datasets generated under different conditions, for the same cell type, we denote to represent the i-th cell from the control condition, and for the j-th cell from the perturbed dataset. scPerb solves the perturbation task through learning the latent features of cell types and the dataset-specific style transformation matrix. Inspired by the VAE architectures, scPerb first estimates the multi-variance normal distribution of the latent cell type feature . Inspired by the style-transfer GAN [20], scPerb uses a neural network to learn the style transformation matrix from the dataset. Compared with scGen, which adopts a constant vector to transfer the latent features from cells of control dataset to perturbed cells, scPerb introduces more learnable parameters and allows the neural network to learn both the style and content difference between the control and perturbed datasets. scPerb demonstrates superior performance and achieves better prediction results compared to existing methods.

**MATERIALS AND METHODS**

Inspired by the style-transfer GAN [20], we presented scPerb, a generative model that can learn the “content” and of the cell types from both the control and stimulus datasets, where represented the “content features” of the cell types, and transfer the style from the control dataset to the perturbed dataset , where represented the “dataset styles”.

scPerb is inspired by Variational Auto-Encoder (VAE) [18] and the style-transfer GAN (stGAN) [20]. We used the variational inference to estimate the distribution and of the “content features” in the latent space, and project style input vector into the latent space and learn the transformation from the control dataset to the perturbed dataset . For the rest of the descriptions, we denote as a content encoder to learn the cell-type awareness features, to project the random-style input vectors to the latent space, and represented to the and estimation for the distribution of , and for the decoder to generate the perturbed data from the latent variables and .

In the inference stage, given a specific cell type from the control dataset , scPerb will extract the cell type-related features , and get the generated pseudo-stimulus cell type based on and , a result of a neuro-network, learning the difference between and .

**Encoders**

We assumed the observations and from the control and perturbed datasets had two independent latent features: a cell type-related latent feature, denoted as “content” , and a dataset-specific feature, denoted as “style” . To extract the common cell type content feature, we first projected the inputs into the latent space, then estimated the to represent the normal distribution of , and resampled the latent variable based on the generated distribution:

We shared the projection weights between the two datasets and , and therefore we could have the latent representation of as:

In this work, our task is to generate the pseudo-stimulus cell types from the same cell types in the control dataset. Therefore, instead of learning the dataset styles explicitly, we applied a light-wise network to learn the transformation in the latent space. Our idea was inspired by the style transfer learnings [20], where randomly sampled a noise and project the latent space as the styles. In ScPerb, we applied a style encoder , which can project the into the latent space as the transformation variable to convert to :

And therefore, we have the following style loss and the KL regulations:

Where SmoothL1Loss and KL divergence are:

**Decoder**

We applied a decoder to generate the observations from the latent variables . Accordingly, the generated samples were denoted as:

Note that our task was to perturb the cell types from the control dataset to the stimulus dataset, instead of generating the samples from and , we use . Therefore, our Generated Loss is:

**Loss function**

The objective functions will be combined with the Generated loss, Style Loss, and the KL regulation terms.

**DATASETS AND PREPROCESS**

We obtained the PBMC-Zheng dataset from Zheng et al. [29]. After removing the megakaryocyte cells, which have an uncertainty of assigned labels, we log-transformed and normalized the data and selected the top 7,000 highly variable genes.

Kang et al. published a dataset from PBMCs including both control and stimulated cell types [23]. Among these data, we extracted the average of the top 20 cluster genes, which has 6998 genes in total, from seven cell types, respectively: B cells, CD4-T cells, CD8-T cells, CD14 Mono cells, Dendritic cells, FCGR3A Mono cells, and NK cells, the same cell types as the PBMC-Zheng dataset.

Harber et al. presented a dataset using the responses of epithelial cells infected by *Salmonella* and *H.poly* [24]. In this dataset, there are 1,770 *Salmonella*-infected cells, 2,711 *H.poly*-infected cells, and the rest 3,240 control cells. The data were also normalized and log-transformed. The top 7,000 highly variable genes were selected in this dataset.

In our model, we performed further data preprocessing to ensure consistency between control and stimulus cells within each cell type. Specifically, for each cell type, we randomly selected an equal number of cells from both the control and stimulated groups and used them to balance the dataset. This data preprocessing step helped us create a more robust and unbiased dataset, enabling accurate and fair comparisons between each cell type’s control and stimulus conditions during subsequent analyses. By doing such data processing, we guarantee that each pair of and have the same cell type, so the following style transfer process would be valid.

**STATISTICS AND REPRODUCIBILITY**

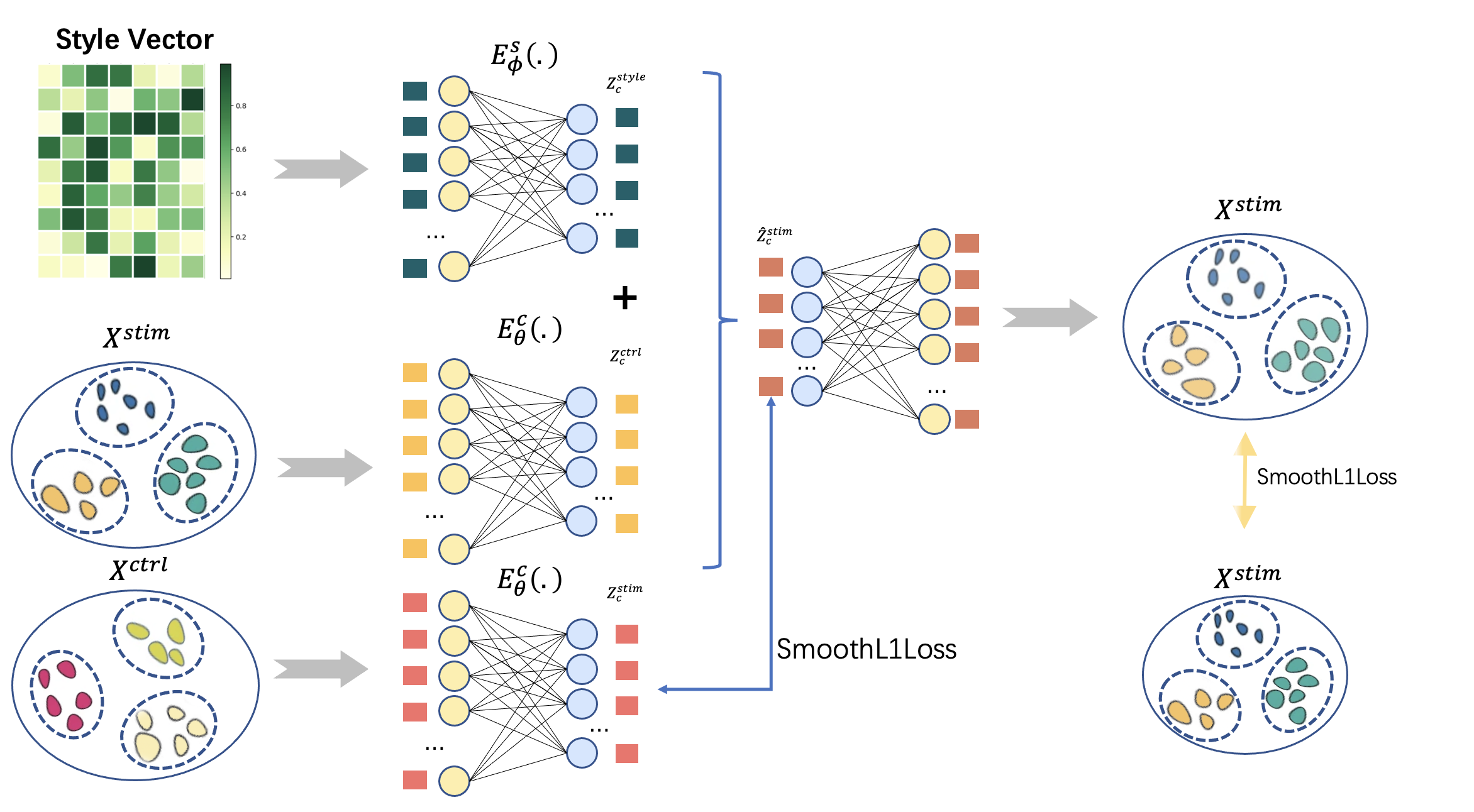
In scPerb, we evaluated the performance of our model under a fixed seed of 42 by using the square of the , which is calculated by the function [30]. This metric measures the correlation between the generated images and the ground truth data. We computed the values for all genes’ mean and variance and the top 100 Differentially Expressed Genes (DEGs).

To understand the model’s results visually, we created scatter plots comparing the generated images to the corresponding ground truth data. This graph allowed us to observe how well the model’s predictions aligned with the actual values.

Additionally, we investigated the differences between the generated images and the ground truth data for the top DEG using a violin plot. The DEGs were identified using the [31] function, employing the Wilcoxon method [32].

Through these analyses, we aimed to assess the accuracy and performance of our ScPerb model in ~~generating realistic images~~  generating cellular graphs that closely resemble actual images based on the input gene expression data. The evaluation of values and the visualization of the scatter and violin plots provided valuable insights into the model’s capabilities and highlighted any discrepancies between the generated and true data for further investigation.

**RESULTS**



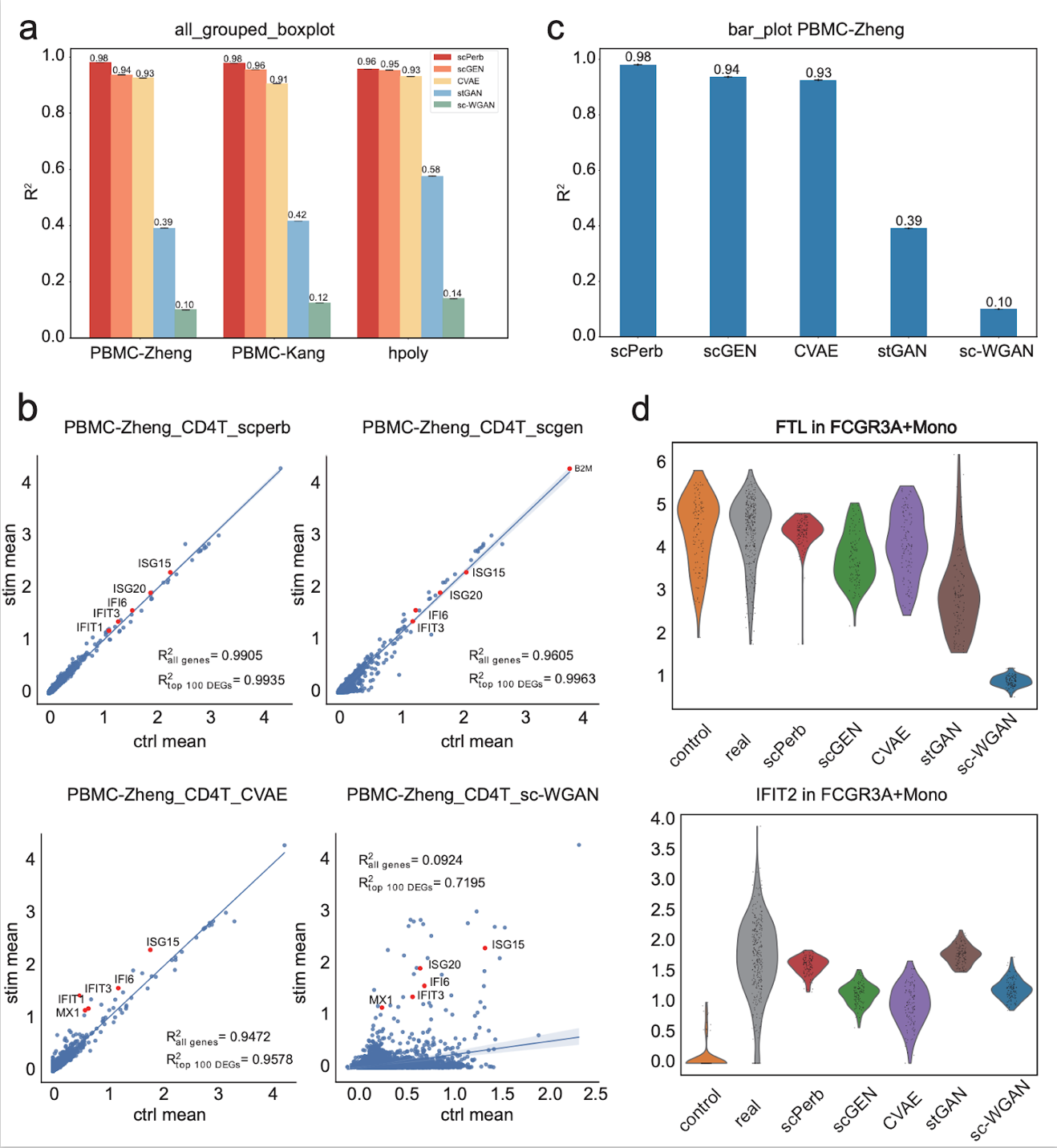
**scPerb outperforms other benchmarks**

To demonstrate the performance of scPerb, we applied it to three datasets. Among these three datasets, two of them are two groups of published human peripheral blood mononuclear cells (PBMC) datasets [23, 29] stimulated with interferon () methods, and the rest is a group of intestinal epithelial cells fetched by parasitic helminth *H.poly* cells [24]. To differentiate the two PBMC datasets, for the rest of the manuscripts, we denote them as PBMC-Kang [23] and PBMC-Zheng datasets [29], and PBMC-Zheng was a mixed published dataset from two different studies.

We fairly compared our proposed scPerb with other benchmarking papers [19-22]. In this process, we select a target cell type and use the rest of the cell types to train a model to transfer the gene expression from the control dataset to the perturbed dataset. We evaluate the model performance using the between the prediction and the ground truth simulation dataset. We repeated such process for all cell types and presented the average of the in Fig. 2a. In the PBMC-Zheng dataset, scPerb achieved the average score of 0.98, which was better than the performance of the competitors, including scGEN (average 0.94), CVAE (average = 0.93), st-GAN (average = 0.39) and sc-WGAN (average = 0.10). Surprisingly, the GAN-based methods had much worse performance, as both GAN-based methods cannot reach a value of exceeding 0.5. Meanwhile, in the PBMC-Kang dataset, our proposed method achieved the highest average score of 0.98, while the second-best and third-best approaches are scGEN and CVAE which had 0.96 and 0.91. Similarly, the GAN-based methods also got bad scores in this dataset: the st-GAN only got an average score of 0.42 and sc-wGAN only got 0.12. Finally, we applied scPerb to the hpoly dataset and still got a 0.96 average score, followed by the scGEN, CVAE, st-GAN, and sc-wGAN with the average score of 0.95, 0.93, 0.58, 0.14. When comparing the result in a specific cell type, scPerb consistently outperforms other benchmarking methods (Fig. 2b). For example, in CD4-T, one of the most numerous cell types in the PBMC-Zheng dataset, scPerb achieved an outstanding score of 0.99, which is much than scGEN, CVAE, stGAN, and sc-WGAN, having score 0.96, 0.95, 0.16, and 0.09 respectively.

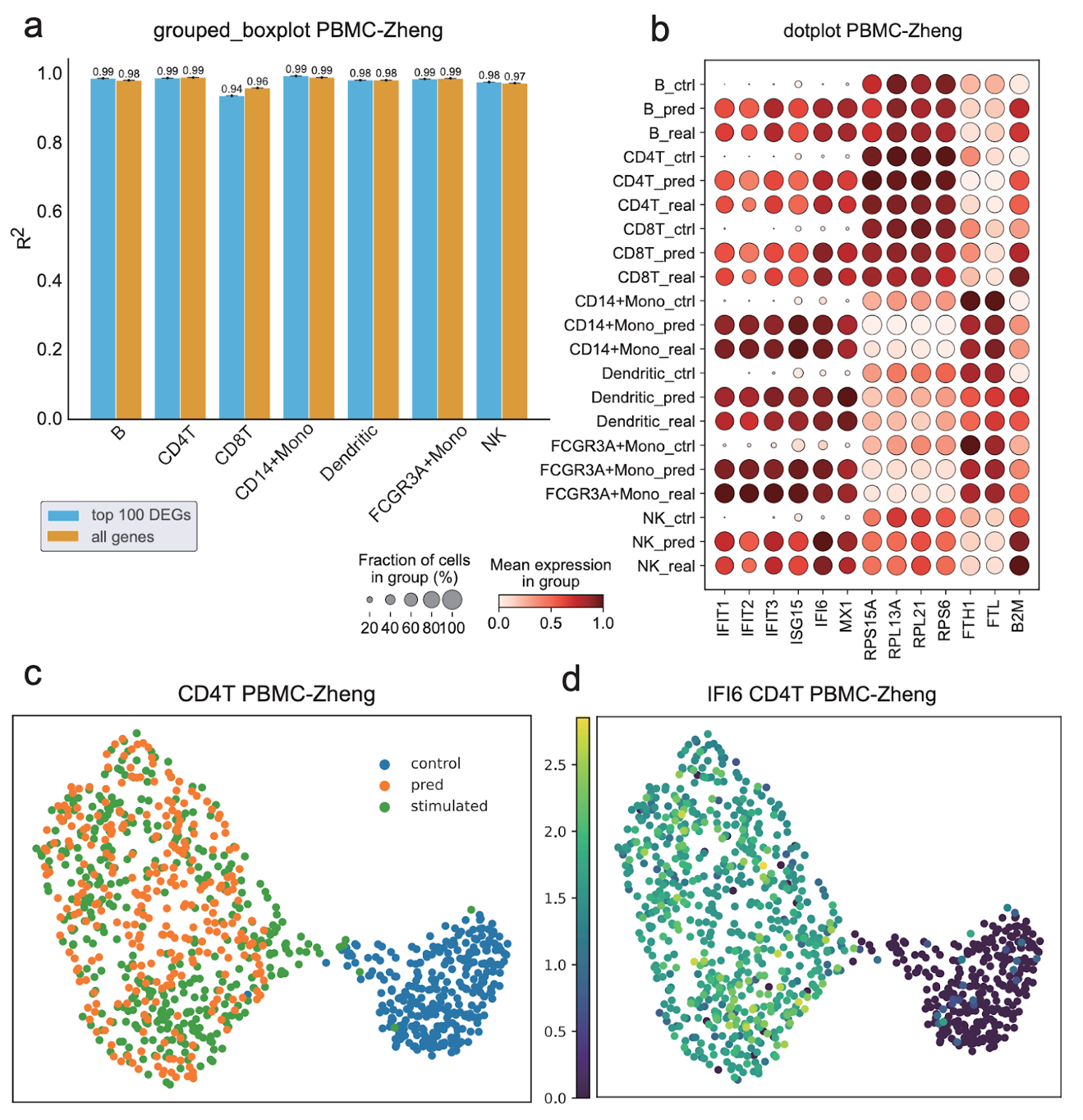
In particular, we evaluate the performance of the proposed scPerb and the benchmarking methods at the gene level. In Fig. 2c, we illustrated the prediction of our scPerb and the performance of the other three benchmarking methods in CD4-T cells from the PBMC-Zheng dataset. The scatter plot demonstrated scPerb got the average score of 0.9905 when we used all the genes in this cell type. The performance can go up to 0.9935 when we only consider the top 100 DEGs. In comparison under the same setting, scGEN achieved the average score of 0.9605 over all genes and 0.9963 on the top 100 DEGs. Our scPerb can outperform CVAE (average score of all genes=0.9472, average score of top 100 DEGs =0.9578) and sc-WGAN (average score = 0.0924, average score=0.9578) on both the evaluation metrics. Specifically, DEGs including IFIT1, IFIT3, IF16, ISG20, and ISG15, showed the best performance.

In Fig. 2d, the distribution of IFIT2 in the control dataset varied from the distribution in the perturbed dataset. In the prediction results, our scPerb was closest to the mean of the perturbed dataset, with a relatively large value range between 0.5 to 3.0 after log transformation. The scGEN and sc-WGAN provided comparable predictions, but the mean of the predictions was slightly larger than the real stimulation data. In this particular gene, CVAE was more associated with the control data, and stGAN focused on the outliers with high gene expressions. In Fig. 2e, the distribution pattern of FTL in the control dataset was similar to the distribution in the real perturbed dataset. Under such a scenario, most of the predictions in scPerb were close to the mean of the simulation data, while the predictions from scGEN and CVAE were expanded to a larger range. St-GAN responded to the high gene expressions while the predictions in sc-WGAN had lower gene expressions. To further illustrate that our result is better than that of benchmarks, we applied Wilcoxon[32] test to these results. In this case, only scPerb results in a P value larger than 0.05 for both genes (0.1763, and 0.0742 respectively for FTL gene and IFIT2 gene), which shows that the prediction of scPerb does not have a significant difference from the ground truth. In contrast, all of the benchmarking methods result in a P value less than 0.05, showing a significant difference from the stimulus dataset. To be more specific, scGEN scored and 0.0033 for FTL gene and IFIT2 gene, while CVAE scored 0.0307 and , stGAN scored and , and sc-WGAN scored and . Therefore, scPerb demonstrates superior performance than the other benchmarking methods.



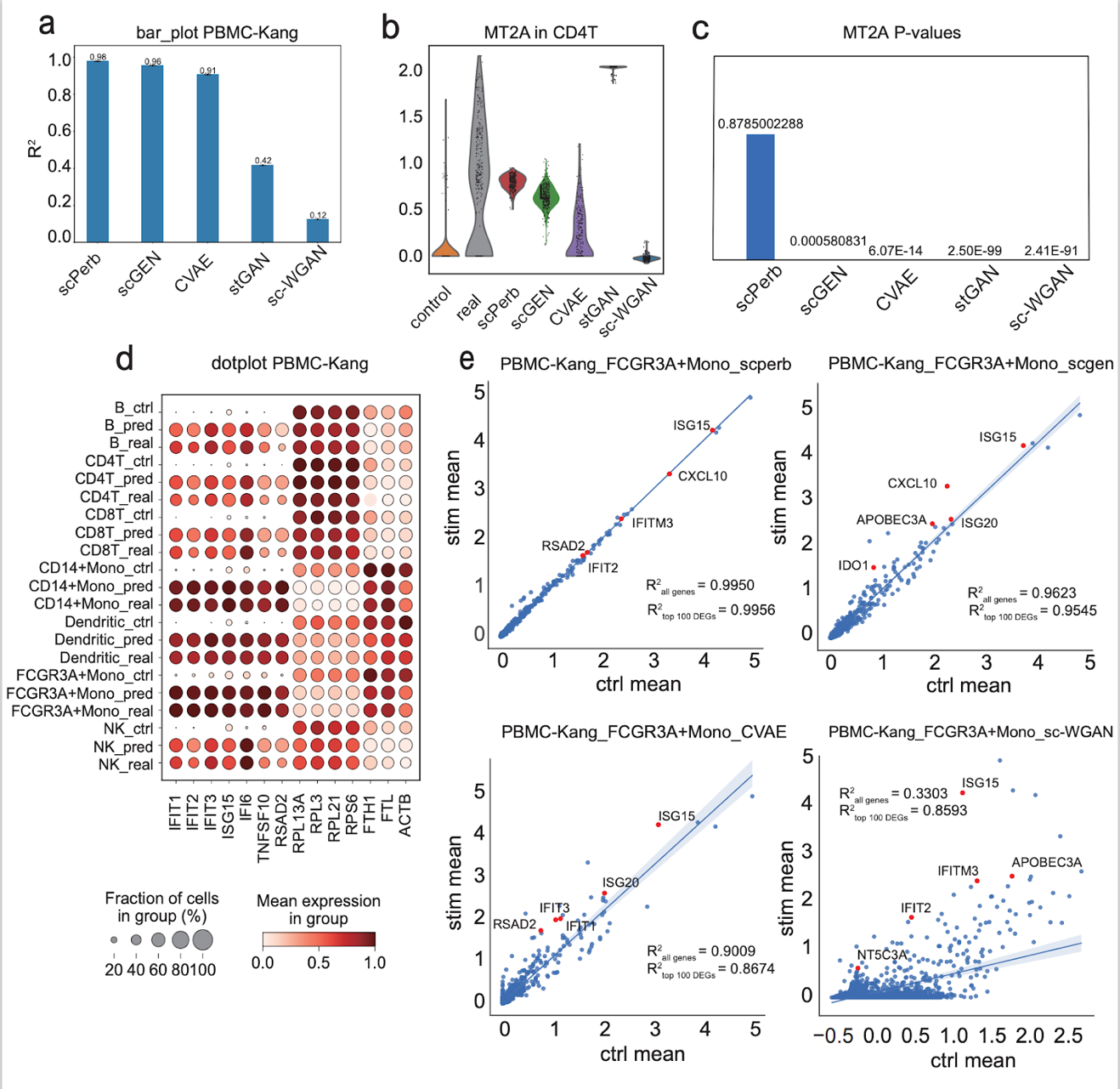
**scPerb is an innovative generative model that can accurately predict single-cell perturbation responses**

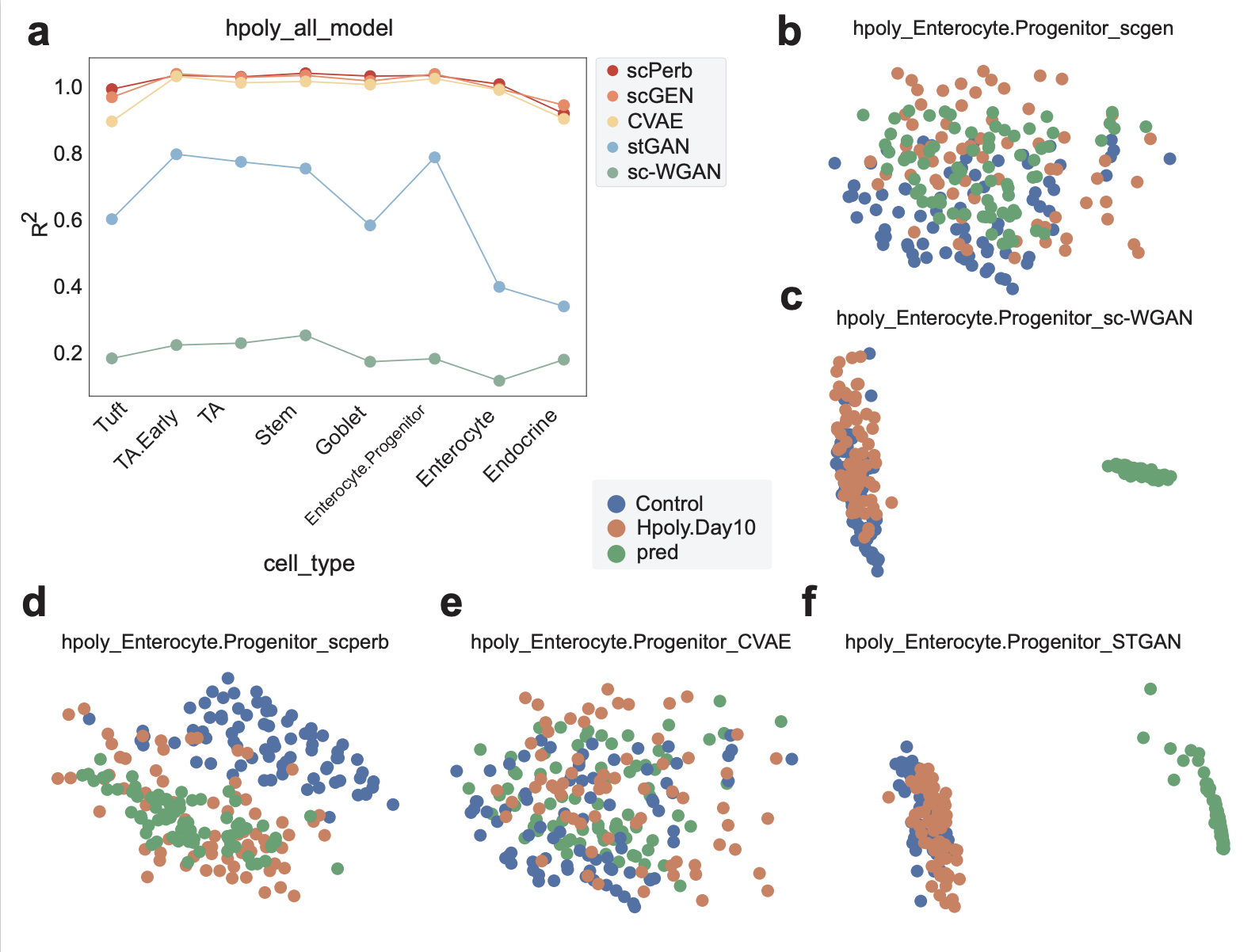
In this section, we aimed to show scPerb can accurately predict the single-cell perturbation responses for other cell types. Fig. 3a summarized the performance of scPerb over different cell types. In CD4-T, CD14+Mono, FCGR3A+Mono cells, scPerb can achieve average score=0.99 in both the top 100 DEGs and all gene expressions. In Dendritic cells, the average score was 0.98 and 0.98 accordingly. In B cells and NK cells, the performance of the top 100 DEGs was slightly better than the performance of all genes, which was 0.99 vs. 0.98 and 0.98 vs. 0.97 respectively. We also observed that in CD8-T cells, the performance of the top 100 DEGs was 0.94, which was slightly lower than the performance on all genes (average score=0.96). In Fig. 3b, the dot plot demonstrated the correlation of representative genes among different cell types. In half of the selected genes, the dot plot showed a strong difference between the gene expression and the real stimulation gene expression. On the other half of the selected genes, we presented similar gene patterns in both the control dataset and perturbed dataset. In the green dashed rectangle box, we highlighted the mean of the expression in the control, predicted, and real stimulus datasets. Fig. 3b implied that the mean gene expression of B cells, CD8-T cells, and Dendritic cells in our scPerb prediction was associated with the mean gene expression in the real perturbed dataset. The UMAP in Fig. 3c showed that the predicted gene expression from scPerb in CD4-T cells was correlated with the real stimulation gene expression in the latent space. In particular, for a specific gene IFI6, we also illustrated the consistent observation.



**scPerb can accurately predict the perturbation of cells in PBMC dataset**

scPerb has robust results in multiple datasets. In PBMC-Kang dataset [23], scPerb still outperforms other methods, achieving 0.98 in the mean of all the cell types, followed by scGen with a of 0.96, CVAE with 0.91, st-GAN with 0.42 and sc-WGAN with 0.12 (Fig. 4a). Moreover, scPerb precisely predicted the result of FCGR3A+Mono cells, reaching of 0.9948 and 0.9978 respectively for all genes and its top 100 DEGs (Fig. 4b). Meanwhile, it is important to note that alternative benchmark methods, including scGEN, sc-WGAN, and style-transfer GAN, exhibited lower values in both the overall gene population and the top 100 differentially expressed genes (DEGs). To provide specific figures, scGEN yielded values of 0.9623 and 0.9545 for all genes and the top 100 DEGs, respectively, while sc-WGAN displayed R^2 values of 0.3303 and 0.8593 for the same categories, and style-transfer GAN yielded values of 0.5223 and 0.7361, respectively. This scatter plot further proved the strong prediction ability of scPerb. Moreover, in MT2A genes, one of the top DEGs in FCGR3A+Mono cells, which also has a control condition filled with zero values, scPerb made a better prediction than any other method, capturing the mean of the ground truth. In this case, the prediction of other methods barely captured the mean of the ground truth. (Fig. 4b) The Wilcoxon test can further explain the difference between the prediction and the real stimulated cells in the MT2A genes: only scPerb achieved a P-value of 0.8785, meaning that there is no statistically significant difference between the prediction of scPerb and the ground truth; however, all other methods including scGEN, CVAE, and both GAN-based methods resulted in a P value far less than 0.0001, showing a significant difference between their predictions and the ground truth (Fig. 4e). Besides, the dot plot (Fig. 4c) shows that scPerb can get robust prediction no matter whether the original control gene expression is lower (for example the IFIT1 gene), approximately the same (for example the RPL13A gene), or higher than (for example the FTH1 gene) the ground truth.





**scPerb has robust results across different datasets**

In alternative datasets, such as the Hpoly dataset[24], the robust predictive capacity of scPerb still remains. For the cell types in the Hpoly dataset, scPerb gained an average of as 0.96, which is better than the scGEN and CVAE (scGEN=0.95, CVAE=0.93), much better than stGAN and sc-WGAN (stGAN=0.38, sc-WGAN=0.14). The line plot in Fig. 5a also illustrated that scPerb maximized its difference in compared with other methods in Tuft cells, having = 0.94. While other VAE-based benchmarks had worse performance (scGEN=0.91, CVAE=0.84). In other cell types, the metrics of scPerb outperform the benchmarking in 7 out of 8 cell types. In the worst case of prediction, the Endocrine cells, scPerb had the second-best performance (), which is also comparable with the best approach scGEN (=0.89). The graph also shows that all VAE-based methods (scPerb, scGEN, CVAE) have a much better result than GAN-based methods (sc-WGAN, stGAN) (Fig. 5a).

Moreover, scPerb made better predictions in this dataset, especially in the Enterocyte.Progenitor cells. In Fig. 5b the distance between the prediction (green dot) and real stimulation data (orange dot) was closer than the distance between the perturbed dataset to the control dataset (blue dot). For the other benchmarks, the VAE-base methods, scGEN (Fig. 5c) and CVAE (Fig. 5d) cannot easily divide the control data samples from the prediction and stimulated data, so their prediction results somewhere in between the control data samples and the stimulated data samples. And the GAN-based approaches, the st-GAN in Fig. 5e and sc-wGAN in Fig. 5f indicated the prediction was far away from both the control and perturbed datasets.

**DISCUSSION**

scPerb is a generative model that dynamically transfers the gene expression in the control dataset into the reliable simulation dataset. The encoder of scPerb projected the raw control gene expression into the high-dimensional latent space and aggregated it with the dataset-specific styles to generate a high-quality representation for the perturbed dataset. Based on the representation, the decoder from scPerb can reconstruct gene expressions that are correlated with the mean of the simulation dataset. The experiments demonstrated that scPerb can capture the latent content features and generate stable dataset-specific styles across different cell types and data from multiple studies. Moreover, the quantitative evaluation indicated the performance of scPerb outperforms four representative benchmarks, having state-of-the-art results in three different datasets.

Compared with the traditional works [19-22], scPerb is a data-driven algorithm that can fully explore the gene expression in the raw dataset and didn’t rely on solid domain priors. On the opposite, the traditional works extract the principle components and build up a graph-based model in the low-dimensional manifold. Such methods relied heavily on the experienced domain knowledge, and lack of generalization abilities. Compared with other data-driven algorithms, scPerb incorporates the stableness from the VAE settings and exploits the advantage of the GAN to generate high-quality samples.

**ACKNOWLEDGMENT**

It is my privilege to have the opportunity as an intern under the supervision of Prof. Song this summer. Previously, I had a few experiences using deep learning for medical image analysis. Therefore, Prof. Song hoped to exploit my previous experience and apply it to the area of single-cell RNA-seq. Therefore, she came up with the idea of single-cell perturbation using the generative model. At the current stage, scRNA-seq can generate sufficient gene expression for the control dataset. However, it was not easy to get the data samples in response to the drugs, doses and treatments, therefore we need to use computational tools to fill up the gap and generate the stimulation samples.

The perturbation task is very challenging because I didn’t have a strong background in bioinformatics, and I spent a month reading tens of previous papers to get familiar with the topic. Thanks to scGEN, one of our benchmarking, shared their codes, so Prof. Song recommend me start from there and build up the pipeline step by step. Together, we download the dataset and rewrite the TensorFlow codes into the PyTorch-style code, and re-generate the performance as the paper described. scGEN is a top journal paper published in Nature Methods, and this paper provides solid results predicting the effect of perturbation which is hard to improve. However, when I re-wrote the codes, I found that scGEN was using a fixed linear transformation matrix to transfer the control cells into the stimulation samples. From my perspective, scGEN did not reach the full potential of the neural network. Therefore, I spent months replacing the linear transformation with the different components, such as using multi-layers, multi-encoders, different loss functions like weighted MSE loss and ZINB loss, and even other models such as GANs or VQVAE. Finally, we found that applying a style encoder on the original VAE can have the best performance.

I would like to thank Prof. Song for her patience and insights during my internship at Wake Forest University. In the first few weeks, I didn’t have many ideas and I discussed them with Prof. Song every day. She gave me many useful suggestions, hands-on with my codes, and help me out in the right direction. When I figure out the methodology and was preparing for the final manuscript, Prof. Song taught me a lot about how to draw the figures, and how to write the manuscripts academically. We went over how to choose the word to clarify a sentence, how to polish the figures and manuscripts, and how to correctly insert the formats and references. It is Prof. Song that showed me a bright new world of computational biology and generative models and provided a platform where I can test all my thoughts. As a high school student, building up a paper from scratch is an unforgettable experience. I believe this experience would greatly benefit me in my future research career.

**REFERENCE**

1. Baron, M., et al., *A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure.* Cell systems, 2016. **3**(4): p. 346-360. e4.

2. Puram, S.V., et al., *Single-cell transcriptomic analysis of primary and metastatic tumor ecosystems in head and neck cancer.* Cell, 2017. **171**(7): p. 1611-1624. e24.

3. Athanasiadis, E.I., et al., *Single-cell RNA-sequencing uncovers transcriptional states and fate decisions in haematopoiesis.* Nature communications, 2017. **8**(1): p. 2045.

4. Azizi, E., et al., *Single-cell map of diverse immune phenotypes in the breast tumor microenvironment.* Cell, 2018. **174**(5): p. 1293-1308. e36.

5. Cusanovich, D.A., et al., *A single-cell atlas of in vivo mammalian chromatin accessibility.* Cell, 2018. **174**(5): p. 1309-1324. e18.

6. Muraro, M.J., et al., *A single-cell transcriptome atlas of the human pancreas.* Cell systems, 2016. **3**(4): p. 385-394. e3.

7. Iram, T. and T.M. Consortium, *Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris.* Nature, 2018. **562**(7727): p. 367-372.

8. Buenrostro, J.D., et al., *Integrated single-cell analysis maps the continuous regulatory landscape of human hematopoietic differentiation.* Cell, 2018. **173**(6): p. 1535-1548. e16.

9. Jagadeesh, K.A., et al., *Identifying disease-critical cell types and cellular processes by integrating single-cell RNA-sequencing and human genetics.* Nature genetics, 2022. **54**(10): p. 1479-1492.

10. Shao, X., et al., *scCATCH: automatic annotation on cell types of clusters from single-cell RNA sequencing data.* Iscience, 2020. **23**(3).

11. Crow, M., et al., *Characterizing the replicability of cell types defined by single cell RNA-sequencing data using MetaNeighbor.* Nature communications, 2018. **9**(1): p. 884.

12. Wei, J.-R., et al., *Identification of visual cortex cell types and species differences using single-cell RNA sequencing.* Nature Communications, 2022. **13**(1): p. 6902.

13. Tasaki, S., et al., *Inferring protein expression changes from mRNA in Alzheimer’s dementia using deep neural networks.* Nature Communications, 2022. **13**(1): p. 655.

14. Denyer, T., et al., *Spatiotemporal developmental trajectories in the Arabidopsis root revealed using high-throughput single-cell RNA sequencing.* Developmental cell, 2019. **48**(6): p. 840-852. e5.

15. Torre, E., et al., *Rare cell detection by single-cell RNA sequencing as guided by single-molecule RNA FISH.* Cell systems, 2018. **6**(2): p. 171-179. e5.

16. Wu, H., et al., *Advantages of single-nucleus over single-cell RNA sequencing of adult kidney: rare cell types and novel cell states revealed in fibrosis.* Journal of the American Society of Nephrology: JASN, 2019. **30**(1): p. 23.

17. Goodfellow, I., et al., *Generative adversarial nets.* Advances in neural information processing systems, 2014. **27**.

18. Kingma, D.P. and M. Welling, *Auto-encoding variational bayes.* arXiv preprint arXiv:1312.6114, 2013.

19. Ghahramani, A., F.M. Watt, and N.M. Luscombe, *Generative adversarial networks uncover epidermal regulators and predict single cell perturbations.* bioRxiv, 2018: p. 262501.

20. Karras, T., S. Laine, and T. Aila. *A style-based generator architecture for generative adversarial networks*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

21. Lotfollahi, M., F.A. Wolf, and F.J. Theis, *scGen predicts single-cell perturbation responses.* Nature Methods, 2019. **16**(8): p. 715-721.

22. Cortes, C., et al. *Advances in neural information processing systems 28*. in *Proceedings of the 29th Annual Conference on Neural Information Processing Systems*. 2015.

23. Kang, H.M., et al., *Multiplexed droplet single-cell RNA-sequencing using natural genetic variation.* Nature biotechnology, 2018. **36**(1): p. 89-94.

24. Haber, A.L., et al., *A single-cell survey of the small intestinal epithelium.* Nature, 2017. **551**(7680): p. 333-339.

25. Hagai, T., et al., *Gene expression variability across cells and species shapes innate immunity.* Nature, 2018. **563**(7730): p. 197-202.

26. Dixit, A., et al., *Perturb-Seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens.* cell, 2016. **167**(7): p. 1853-1866. e17.

27. Adamson, B., et al., *A multiplexed single-cell CRISPR screening platform enables systematic dissection of the unfolded protein response.* Cell, 2016. **167**(7): p. 1867-1882. e21.

28. Datlinger, P., et al., *Pooled CRISPR screening with single-cell transcriptome readout.* Nature methods, 2017. **14**(3): p. 297-301.

29. Zheng, G.X., et al., *Massively parallel digital transcriptional profiling of single cells.* Nature communications, 2017. **8**(1): p. 14049.

30. Virtanen, P., et al., *SciPy 1.0: fundamental algorithms for scientific computing in Python.* Nature methods, 2020. **17**(3): p. 261-272.

31. Wolf, F.A., P. Angerer, and F.J. Theis, *SCANPY: large-scale single-cell gene expression data analysis.* Genome biology, 2018. **19**: p. 1-5.

32. Cuzick, J., *A Wilcoxon‐type test for trend.* Statistics in medicine, 1985. **4**(1): p. 87-90.