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

Zijia Tang1, Qianqian Song2,\*

1Trinity College, Duke University, Durham, NC, USA; 2Department of Health Outcomes and Biomedical Informatics, College of Medicine, University of Florida, Gainesville, FL, USA

qsong1@ufl.edu

[zijia.tang@duke.edu](mailto:zijia.tang@duke.edu)

**ABSTRACT**

Traditional methods for obtaining cellular responses after perturbation are usually labor-intensive and costly, especially when working with multiple different experimental conditions. Therefore, accurate prediction of cellular responses to perturbations is of great importance in computational biology. To address this problem, 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 black-box models. In this work, we introduce a novel framework, scPerb, to explicitly extract the perturbation-related variances and transfer them from unperturbed cells to perturbed cells. scPerb adopts the style transfer strategy by incorporating a style encoder into the architecture of a variational autoencoder. Such style encoder accounts for the differences in the latent representations between unperturbed cells and perturbed cells, which allows scPerb to accurately predict the gene expression data of cells after perturbation. Through comprehensive 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, and 0.96 on three benchmarking datasets.

**KEYWORDS**

Single-cell RNA sequencing, Perturbation, Style transfer, Variational auto-encoder

# INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) is a revolutionary technology to profile gene expression of cells in heterogeneous tissue samples{Baron, 2016 #56;Puram, 2017 #65;Athanasiadis, 2017 #54}. This technology can measure transcripts in thousands of single cells from multiple biological samples under different conditions{Azizi, 2018 #55;Cusanovich, 2018 #60;Muraro, 2016 #64;Iram, 2018 #62;Buenrostro, 2018 #58}. Such breakthrough technology has inspired the development of tailored computational tools such as cell type annotations{Jagadeesh, 2022 #74;Shao, 2020 #75;Crow, 2018 #76;Wei, 2022 #77}, identification of pseudo-time trajectories{Tasaki, 2022 #84;Denyer, 2019 #87}, and rare cell type detection{Torre, 2018 #89;Wu, 2019 #90}, facilitating the biological insights into single-cell data{Andrews, 2021 #1;Chen, 2019 #2}.

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

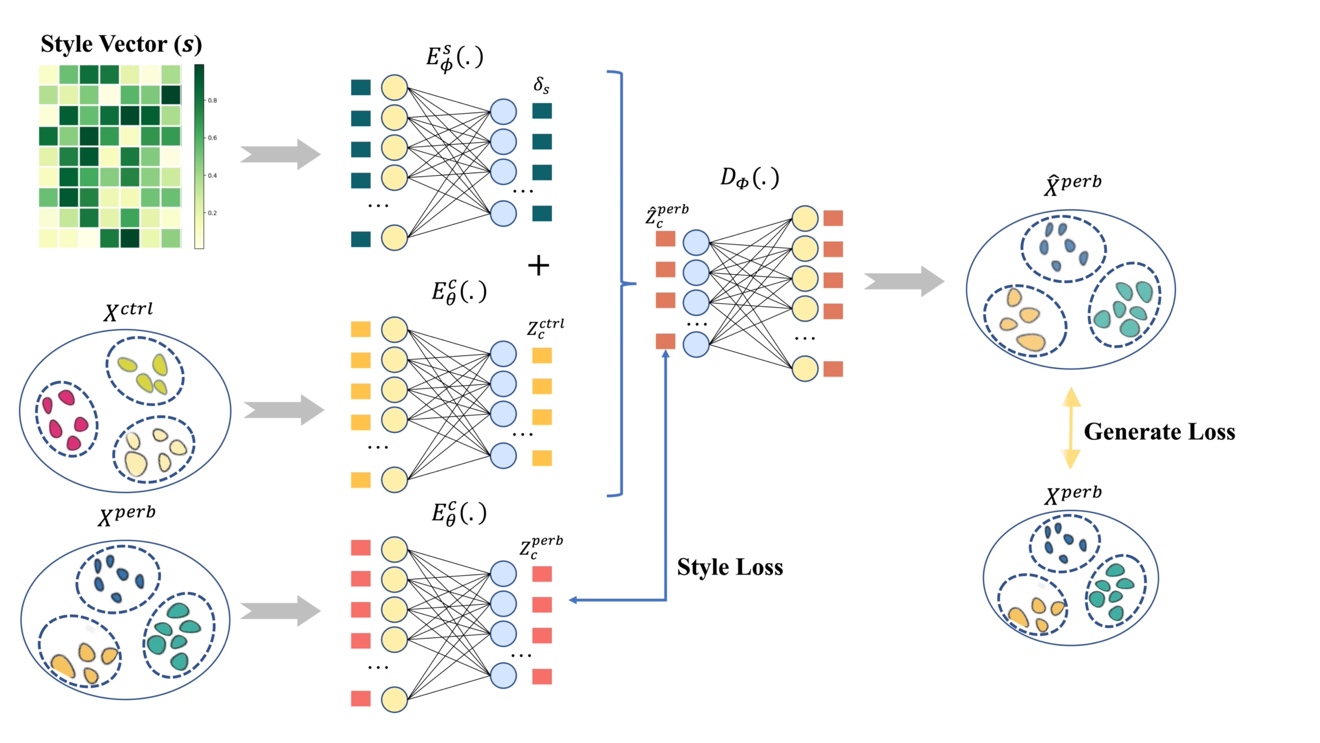
In recent studies, the gaps in perturbation tasks were addressed using generative models like Generative Adversarial Networks (GAN){Goodfellow, 2014 #20} and Variational Auto-Encoders (VAE){Kingma, 2013 #6}, to fill the missing pieces in perturbation tasks. Specifically, GAN-based models introduced a generator to learn perturbed data and trained an adversarial discriminator to determine whether the predicted data was close to ground truth or not. Such adversarial battle aimed to train a robust generator to infer accurate gene expressions. However, the major drawback of GAN lay in the difficulty in balancing the adversarial training, leading to a useless collapsed generator that was very sensitive to the input data noise. sc-WGAN{Ghahramani, 2018 #13} transferred a more stable WGAN to the single-cell perturbation and style-transfer GAN (stGAN){Karras, 2019 #21} introduced the idea of style transfer learning with the inclusion of multiple styles to the generator. On the other hand, VAE-based models generated gene expressions by sampling from a multivariate Gaussian distribution using variational inference. For example, scGen{Lotfollahi, 2019 #12} assumed a fixed linear gap between unperturbed cells and perturbed cells, calculated the latent differences between two conditions, and predicted the perturbed gene expressions using latent representation of cells from the two conditions.

In this work, we presented a novel tool, i.e., scPerb, to predict single-cell gene expressions under specific conditions such as a dose{Kang, 2018 #9}, a treatment{Haber, 2017 #8;Hagai, 2018 #29}, or a modification of genes{Dixit, 2016 #34;Adamson, 2016 #35;Datlinger, 2017 #33} (**Fig.1**). Given two datasets generated under different conditions, for the same cell type, we denoted to represent the i-th cell from the control condition, and for the j-th cell from the perturbed condition. scPerb solved the perturbation task by learning the latent features of cell types and the condition-specific style vector. Specifically, scPerb estimated the multi-variance normal distribution of the cell type feature c. scPerb also used a neural network to learn the style transformation matrix from the datasets. Different with previous methods that adopt a constant vector to transfer the latent features from cells of the control condition to that of the perturbed condition, scPerb introduces learnable parameters and allows the neural network to learn both cell type and condition differences between the control and perturbed datasets. With comprehensive evaluation, scPerb performs better with more accurate prediction results when compared to other approaches.

# MATERIALS AND METHODS

Here we presented scPerb, a generative model to predict gene expression data after perturbation. We hypothesized 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” . scPerb learned the contents and of the cell types from both the control and perturbed datasets, where represented the content features of the cell types and transferred the style from the control dataset to the perturbed dataset , and represented the dataset styles (**Fig. 1**).

scPerb first translated the input data into a probability distribution in the latent space using an encoder. Specifically, it mapped the input data to a mean () and a variance () for each latent variable. We then projected the style vector into the latent space and learned the transformation from the control dataset to the perturbed dataset , and the learned difference between and would be denoted as . Furthermore, we denoted as the content encoder acquiring the cell-type awareness features, as the style encoder projecting the random style vectors to the latent space, and as the and estimation for the probability distribution generated by the encoders, and as the decoder generating the perturbed data using the latent variables and . In the inference stage, given a specific cell type from the control dataset , scPerb would extract the cell type-related features , generate the “fake” perturbed cell type based on and , and minimize the differences between and .



**Fig. 1 | scPerb predicts gene expressions of perturbed cells.** scPerb was designed to predict gene expressions in perturbed cells and combines the principles of both style transfer and VAE. With the perturbed and control dataset as inputs, the content encoder projected the data into latent space. Differences between the latent representations of the perturbed dataset and the control dataset were captured by a style vector (s), which enabled transferring from the perturbed style to the control style. Such style vector was initiated with a random vector and updated via a style encoder, which learned the style of the perturbed dataset and transferred it to the control dataset by adding it to the latent representation of the control dataset. By minimizing the differences between both latent representations and gene expressions between predicted perturbed data and real perturbed data, scPerb transferred the control style to the perturbed style and predicted the gene expression of perturbed cells**.**

## Encoders

To extract common cell type content features, we projected both inputs into the latent space. Followed by the setting of VAE, we assumed the content features were multivariate normal distributions, , where and represented the mean and variance of multivariate normal distribution). The latent representation of input data was obtained from the learned distribution

where and .

Since the projection weights were shared between the two input datasets and , the latent representation of input data was obtained from , where and . Followed by VAE settings, we used KL loss to estimate , , , and :

where KL divergence was calculated by:

In this work, our task was to generate the “fake” perturbed 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 {Karras, 2019 #21}, where randomly sampled style vector () and projected 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 :

Therefore, we had the following :

While the was defined below:

## Decoder

In the decoder part, scPerb reparametrized the latent variable from the estimated posterior distribution and . Unlike the standard VAE, which directly reconstructed the output from the latent variable and , scPerb converted the representation of the control data to the latent representation , and generated the predicted perturbed data from decoder :

Note that our task was to predict the perturbation of the cell types using the control dataset, instead of generating the samples from and as the original VAE, we only used to generate . Therefore, our was:

## Loss function

The final objective function consisted of the , , and the regulation terms.

# DATASETS AND PREPROCESS

We obtained the PBMC-Zheng dataset from Zheng et al. {Zheng, 2017 #14}. After removing the megakaryocyte cells that had uncertainly assigned labels, we log-transformed and normalized the data and selected the top 7,000 highly variable genes. The resulting dataset contains 18,868 PBMC cells, including 9,925 perturbed cells infected by and 8,943 control cells.

Kang et al. published a dataset of PBMCs including both control and perturbed cells (also infected by ) {Kang, 2018 #9}. Among these data, we extracted the average of the top 20 cluster genes, which has 6,998 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* {Haber, 2017 #8}. In this dataset, there were 3,240 control cells, 2,711 *H.poly*-infected cells, and the rest 1,770 Salmonella-infected cells. The data were also normalized and log-transformed, and 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 perturbed cells within each cell type. Specifically, we randomly selected an equal number of control cells and perturbed cells for each cell type in order to balance the dataset. This data preprocessing step helped us create a more robust and unbiased dataset, enabling accurate comparisons in each cell type. By doing such data processing, we guaranteed 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 R value (), calculated through function {Virtanen, 2020 #27}. This metric evaluated the degree to which the predicted perturbed data and the real perturbed data were correlated. We computed the values for all genes’ mean and variance and the top 100 Differential Expressed Genes (DEGs). To understand the model’s results visually, we created scatter plots comparing the predicted perturbed data to the corresponding ground truth data. This graph allowed us to observe how well the model’s predictions aligned with the actual values.

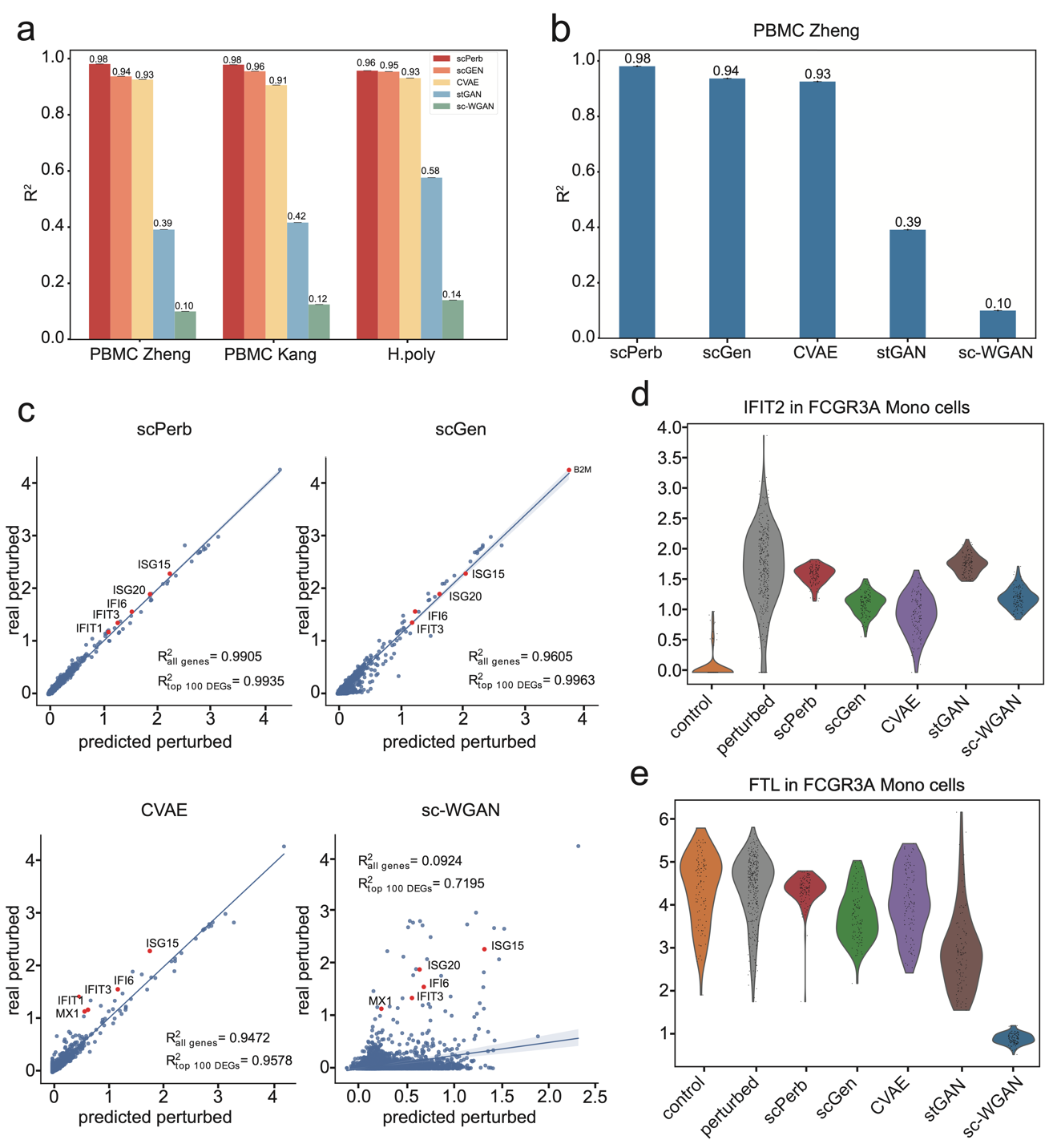
Additionally, we used a violin plot to examine the discrepancies between the predicted perturbed data and the real perturbed data for the top DEGs. The DEGs were identified using the {Wolf, 2018 #11} function, employing the Wilcoxon method {Cuzick, 1985 #16}.

Through these analyses, we aimed to assess the accuracy and performance of our scPerb model 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 predicted and real perturbated data for further investigation.

# RESULTS

## scPerb outperforms other benchmarking methods

To demonstrate the performance of scPerb, we compared scPerb with currently existing methods, including scGen {Lotfollahi, 2019 #12}, CVAE {Cortes, 2015 #7}, stGAN {Karras, 2019 #21}, and sc-WGAN {Ghahramani, 2018 #13}. Three datasets were used for benchmarking, including two published human peripheral blood mononuclear cell (PBMC) datasets, i.e., PBMC-Kang {Kang, 2018 #9} and PBMC-Zheng {Zheng, 2017 #14} datasets, which were perturbed with interferon (), and the intestinal epithelial cell dataset fetched by parasitic helminth *H.poly* {Haber, 2017 #8}, i.e., *H.poly* dataset.



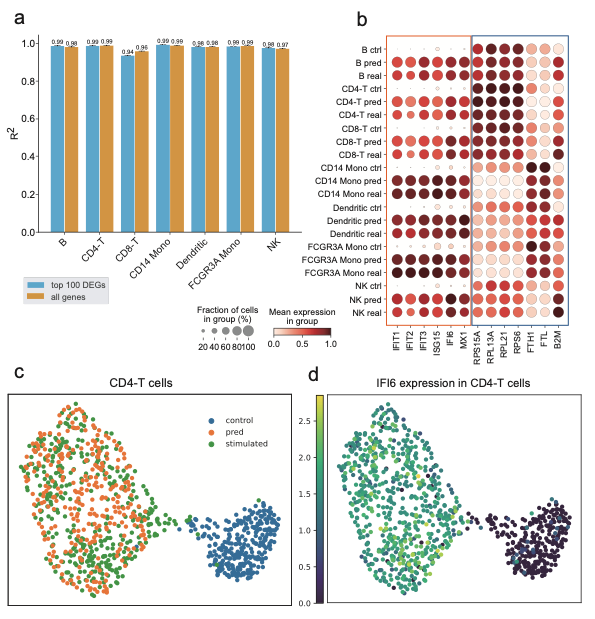
**Fig. 2 |Results of scPerb in general.** **a**: Comparison of values across all benchmarking methods; **b**: Bar plots showed the value of all methods in the PBMC-Zheng dataset {Zheng, 2017 #14}; **c**: Scatter plot showed the correlation between real and predicted gene expression of 7,000 genes by scPerb and other three benchmarking methods in CD4-T cells, and the five red dots represented the top five DEGs; **d**: The distribution of the control dataset, perturbed dataset, and the prediction of all methods in one of the least DEGs (*FTL*), and one of the top DEGs (*IFIT2*)

Based on those three datasets, each method’s performance was evaluated using the between predictions and real perturbed data. Specifically, we randomly selected a cell type to predict its gene expression data after perturbation, meanwhile using the rest of the cell types for model training. We repeated such process across all cell types and presented the average of the in **Fig. 2a**. In the PBMC-Zheng dataset {Zheng, 2017 #14}, scPerb achieved the average score of , which was better than the performance of the competitors, including scGen (average = ), CVAE (average = ), stGAN (average = ) and sc-WGAN (average = ). Surprisingly, the GAN-based methods had much worse performance, as both GAN-based methods could not reach a value exceeding . Meanwhile, in the PBMC-Kang dataset, scPerb achieved the highest average score of , while the second-best and third-best approaches were scGen and CVAE which had and . Similarly, the stGAN and sc-wGAN only had an average score of and , respectively, in this dataset. Finally, we applied scPerb to the H.poly dataset and still got a average score, followed by the scGen, CVAE, stGAN, and sc-wGAN with the average score of , , , . When comparing their results in a specific cell type, scPerb consistently outperformed other benchmarking methods (**Fig. 2b**). For example, in CD4-T cell type, one of the most numerous cell types in the PBMC-Zheng dataset, scPerb achieved a superior score of , which was much better than scGen, CVAE, stGAN, and sc-WGAN ( score: , , , and ) respectively.

In addition, we evaluated the performance of the proposed scPerb and the other benchmarking methods across genes. 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 that scPerb got the average score of when we used all the genes in this cell type. The performance could go up to when we only consider the top 100 DEGs. In comparison under the same setting, scGen achieved the average score of over all genes and on the top 100 DEGs. Our scPerb could outperform CVAE (average score of all genes = , average score of top 100 DEGs = ) and sc-WGAN (average score = , average score = ) on both the evaluation criteria. Specifically, DEGs including *IFIT1*, *IFIT3*, *IFI6*, *ISG20*, and *ISG15*, showed the best performance.

In **Fig. 2d**, the distribution of *IFIT2* in the control dataset largely differed from the distribution of its perturbed dataset. Notably, based on the predictions of perturbed gene expressions, the mean of scPerb’s prediction was close to the mean of the perturbed dataset. However, the distribution of scGen’s and st-WGAN’s prediction was comparable to the ground truth but resulted in a mean much lower than the mean of the ground truth. The predictions of CVAE resulted somewhere in between the control data and the perturbed data, meaning that it cannot clearly learn the style difference between control data and perturbed data. Though the prediction of stGAN seems to resemble the mean of the ground truth, the Wilcoxon test {Cuzick, 1985 #16} resulted in P value less than , showing the significant difference between the mean of stGAN’s prediction distribution and the ground truth. For the other gene *FTL*, as shown in **Fig. 2e**, its distribution pattern in the control dataset resembled the distribution in the real perturbed dataset. Under such scenario, most of the predictions in scPerb were close to the mean of the perturbed data, whereas the predictions from scGen and CVAE exhibited a much lower mean compared with the ground truth. Both GAN-based methods stGAN and sc-WGAN presented many outliers which were deviate from the perturbed data. To further illustrate that our result was better than that of benchmarks, we applied Wilcoxon test to these results. In this case, only scPerb resulted in an adjusted P value larger than for both genes (, and respectively for the *FTL* gene and the *IFIT2* gene), which showed that the prediction of scPerb did not have a significant difference from the ground truth. In contrast, all benchmarking methods resulted in P values less than , showing a significant difference from the ground truth. To be more specific, scGen scored and for the *FTL* gene and the *IFIT2* gene, while CVAE scored and , stGAN scored and , and sc-WGAN scored and . Therefore, scPerb demonstrated superior performance than the other benchmarking methods.

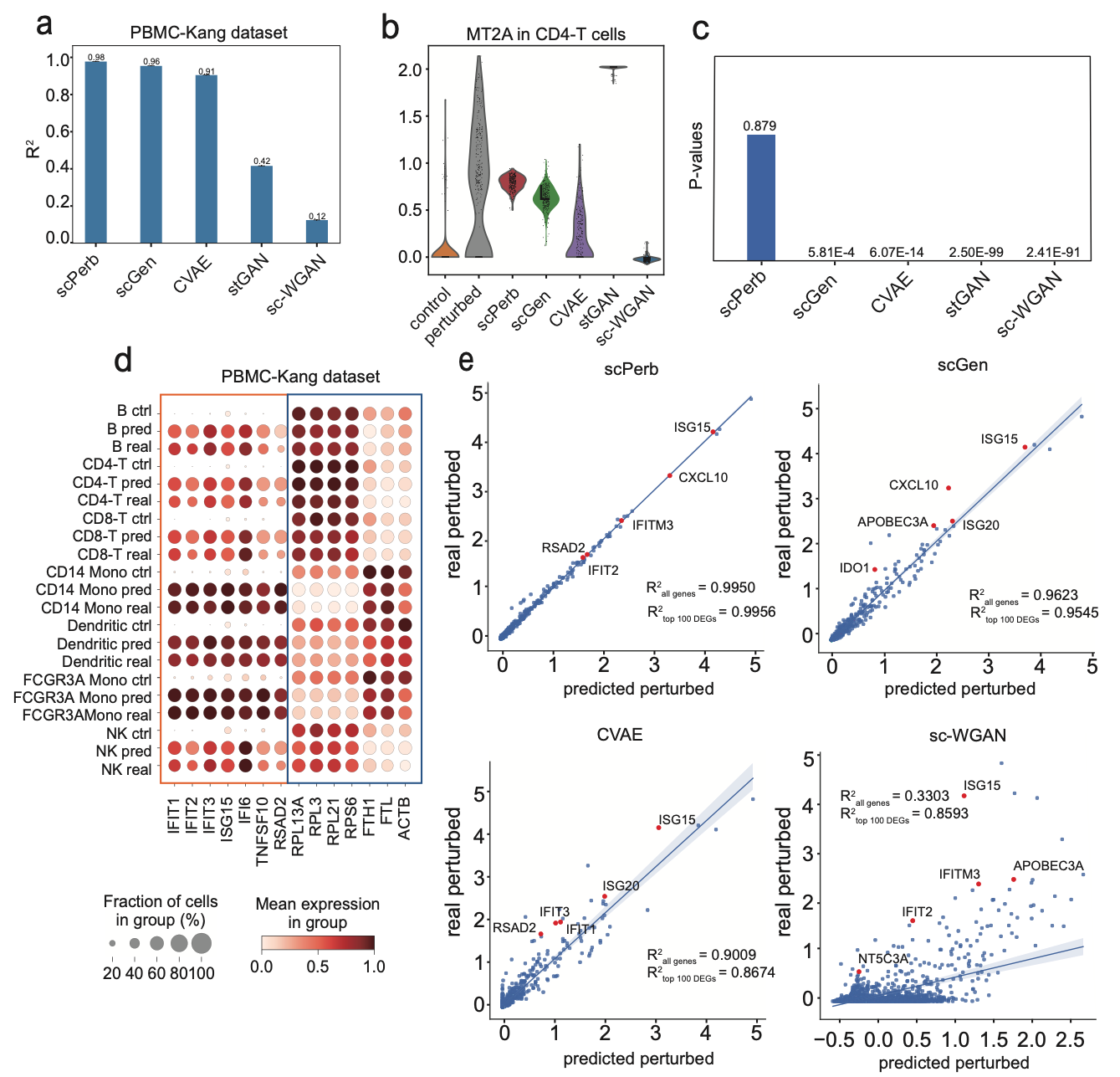
## scPerb predicts single-cell perturbation response accurately

****

**Fig. 3 | Result of scPerb in PBMC-Zheng dataset.** **a**: Grouped boxplot showed the result of scPerb in values in all genes and the top 100 DEGs in every cell type in the PBMC-Zheng dataset; **b**: Dot plot illustrating the mean gene expression in each cell type and condition; **c-d**: UMAP {McInnes, 2018 #91} visualizations depicted the condition distribution of the overall CD4-T cell type in the PBMC-Zheng dataset and the expression pattern of *IFI6*, one of the top DEGs in the CD4-T cells.

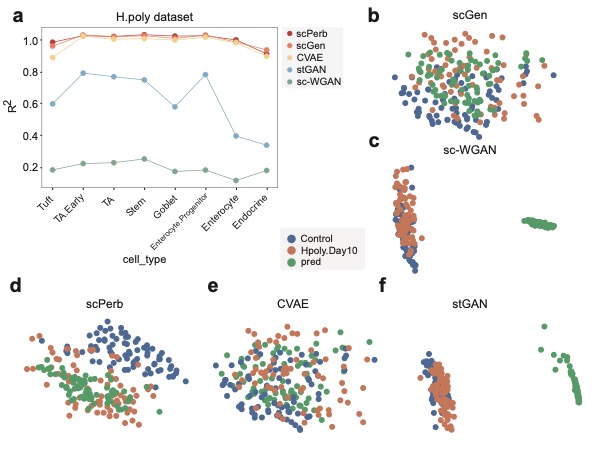
In this section, we aimed to show that scPerb could accurately predict the single-cell perturbation responses for other cell types. **Fig. 3a** summarized the performance of scPerb over different cell types. In CD4-T, CD14 Mono, and FCGR3A Mono cells, scPerb could achieve an average score = 0.99 in both the top 100 DEGs and all gene expressions. In Dendritic cells, the average score was 0.98 and 0.98 respectively. 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 perturbed gene expression. On the other half of the selected genes, we presented similar gene patterns in both the control dataset and the perturbed dataset. In the green dashed rectangle box, we highlighted the mean of the expression in the control, predicted, and real perturbed 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 perturbed gene expression in the latent space. Such consistent observation was also observed for a specific gene IFI6.

## scPerb accurately predicts the perturbation of cells in multiple PBMC datasets

****

**Fig. 4 | Result of scPerb in PBMC-Kang dataset.** **a**: This bar plot compared the values of all the methods within the PBMC-Kang dataset, while central values represented the mean values across all 7 cell types in the dataset; **b-c**: Comparing the distribution of all the methods in the *MT2A* gene in CD4-T cells in the PBMC-Kang dataset. Center values in **Fig. 4c** were the adjusted P values comparing the prediction of each method to the ground truth by using the Wilcoxon test; **d**: A dot plot comparing the mean gene expression of all 7 cell types and all 3 conditions in the PBMC-Kang dataset; **e**: The correlation of the mean expression of all 6,998 genes in FCGR3A Mono cells. It compared predictions from three of the best benchmark methods and scPerb against the ground truth, with shaded lines representing the confidence interval of the regression estimate.

scPerb had robust predictions of perturbed gene expressions in multiple datasets. For PBMC-Kang dataset {Kang, 2018 #9}, scPerb still outperformed other methods, achieving in the mean of all the cell types, followed by scGen with a of , CVAE with , stGAN with and sc-WGAN with (**Fig. 4a**). Moreover, scPerb precisely predicted the perturbed gene expressions of FCGR3A Mono cells, reaching of and respectively for all genes and its top 100 DEGs. The top 100 DEGs as well as the entire gene population showed lower values for alternative benchmark approaches including scGen, sc-WGAN, and stGAN. To be more specific, scGen produced values of and for all genes and the top 100 DEGs, respectively. For the same categories, sc-WGAN revealed values of and , and stGAN showed values of and , in the same categories. This scatter plot further reinforces scPerb’s robust predictive abilities. Moreover, for the *MT2A* gene, one of the top DEGs in FCGR3A Mono cells, scPerb presented a better prediction than the other methods, capturing the mean of the ground truth. In this case, the predictions of other methods were not close to the real perturbed data (**Fig. 4b**). The Wilcoxon test further explained the differences between the predicted gene expressions and the real perturbed expressions for the *MT2A* gene: only scPerb achieved a P value of , showing that the difference between the prediction of scPerb and the real perturbed data was not statistically different; however, all other methods including scGen, CVAE, and both GAN-based methods resulted in an adjusted P value far less than , indicating significant differences between their predictions and the real perturbed data (**Fig. 4c**). Besides, **Fig. 4d** showed that scPerb could get robust prediction no matter whether the control gene expression was lower (for example the *IFIT1* gene), approximately comparable (for example the *RPL13A* gene), or higher (for example the *FTH1* gene) than the real perturbed gene expression. Moreover, it is worth noting that the predictions of scPerb correlated better with the real perturbed data, especially the top 5 DEGs (the red dots shown in **Fig. 4e**). The values of scPerb ( and for all genes and the top 100 DEGs) were also higher than all the other benchmarks including scGen, CVAE, and sc-WGAN.

****

**Fig. 5 | The result of scPerb in the H.poly dataset** **a**: Line plot using to compare the outcomes of all the methods; **b-f**: The UMAP visualization of the control, perturbed, and predicted cells.

## scPerb has robust results across different datasets

In the H.poly dataset {Haber, 2017 #8}, scPerb maintained superior performance with robust predictive capacity. For the cell types in the H.poly dataset, scPerb gained an average of as , which was better than scGen and CVAE (scGen = , CVAE = ), as well as stGAN and sc-WGAN (stGAN = , sc-WGAN = ). The line plot in **Fig. 5a** also illustrated that scPerb maximized its difference in compared with other methods in Tuft cells, with as . In contrast, other VAE-based methods had worse performance (scGen = , CVAE = ). **Fig. 5a** also showed that all VAE-based methods (scPerb, scGen, CVAE) presented much better predictions than GAN-based methods (sc-WGAN, stGAN). In most of the cell types, scPerb showed superior performance than the benchmarking methods. In addition, scPerb made better predictions in the Enterocyte Progenitor cells of this H.poly dataset. As shown in **Fig. 5b**, the predictions of scPerb (green dot) was closer to real perturbed data (orange dot), compared with the unperturbed dataset (blue dot). For the other methods **Fig. 5c-f**), their predictions could not be distinguished from the unperturbed data or the real perturbed data.

# 

# DISCUSSION

scPerb is a novel generative model that predicts gene expressions after perturbation. The encoder of scPerb projects gene expressions of both control and perturbed data into the high-dimensional latent space. scPerb aggregates 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 of perturbed data. The experiments demonstrate that scPerb can capture the latent content features and generate dataset-specific styles across different cell types and conditions. Moreover, the quantitative evaluation indicated the performance of scPerb outperforms four existing methods, presenting outperformed results in each cell types of three different datasets.

Compared with previous work{Ghahramani, 2018 #13;Karras, 2019 #21;Lotfollahi, 2019 #12;Cortes, 2015 #7}, scPerb is a data-driven algorithm that fully explores the gene expression in the raw dataset and does not rely on solid domain priors. On the opposite, previous work extract the principal components and build up a graph-based model in the low-dimensional manifold. Such methods rely heavily on the experienced domain knowledge, and lack of generalization capabilities. 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.

However, minor problems still exist. In Endocrine cells in the H.poly dataset, one of the cell types containing the fewest cells in the H.poly dataset (163 in 5,059), scPerb makes predictions slightly worse than scGen {Lotfollahi, 2019 #12}. Using values as a criterion, scGen results in while scPerb only results in . Note that scGen only calculates a fixed liner vector while scPerb uses style transfer, in this case, the problem of “overfitting” exists. However, such cases are very rare and scPerb can still outperform other methods such as scGen in other cases when the data is small. In Tuft cells, also one of the cell types containing the fewest cells in the H.poly dataset (248 in 5,059), scPerb achieves a value of while scGen only gets .

# CODE AVAILABILITY

scPerb is provided as a Python package available at <https://github.com/QSong-github/scPerb>, with detailed functions for implementation.

# AUTHOR CONTRIBUTIONS STATEMENT

Z.T. and Q.S. conceived and managed the project. Z.T. implemented the code. Z.T. and Q.S. analyzed the result and wrote the manuscript. All authors read and approved the manuscript.

# ACKNOWLEDGEMENTS

Q.S. is supported by the National Institute of General Medical Sciences of the National Institutes of Health (R35GM151089).

# REFERENCE