1. **Methods**

Inspired by the transfer learning paradigm, 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 stimulation dataset , where represented the “dataset styles”.

scPerb is inspired by Variational Auto-Encoder (VAE)[1] and the style-transfer GAN (stGAN)[2]. 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 stimulation 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 stimulation 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 the pseudo-stimulus cell type based on and , a result of a neuro-network, learning the difference between and .

* 1. Encoders

We assumed the observations and from the control and stimulation 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 project the inputs into the latent space, then estimate the to represent the normal distribution of , and resample the latent variable based on the generated distribution:

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

In this manuscript, 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[2], 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:

* 1. 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:

* 1. Loss function

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

1. **Datasets and preprocess**

Mohammad et al. [3] included three groups of control and stimulated cells: two groups of PBMC cells, and a group of HPOLY cells. Mohammad et al. preprocessed the data by removing megakaryocytic cells, filtering the cells with a minimum of 500 expressing cells, extracting the top 6998 cells, and log-transforming the original data. All the data are available on <https://github.com/tongtongtot/scperb/tree/master/data>.

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.

1. **Statistics and Reproduction**

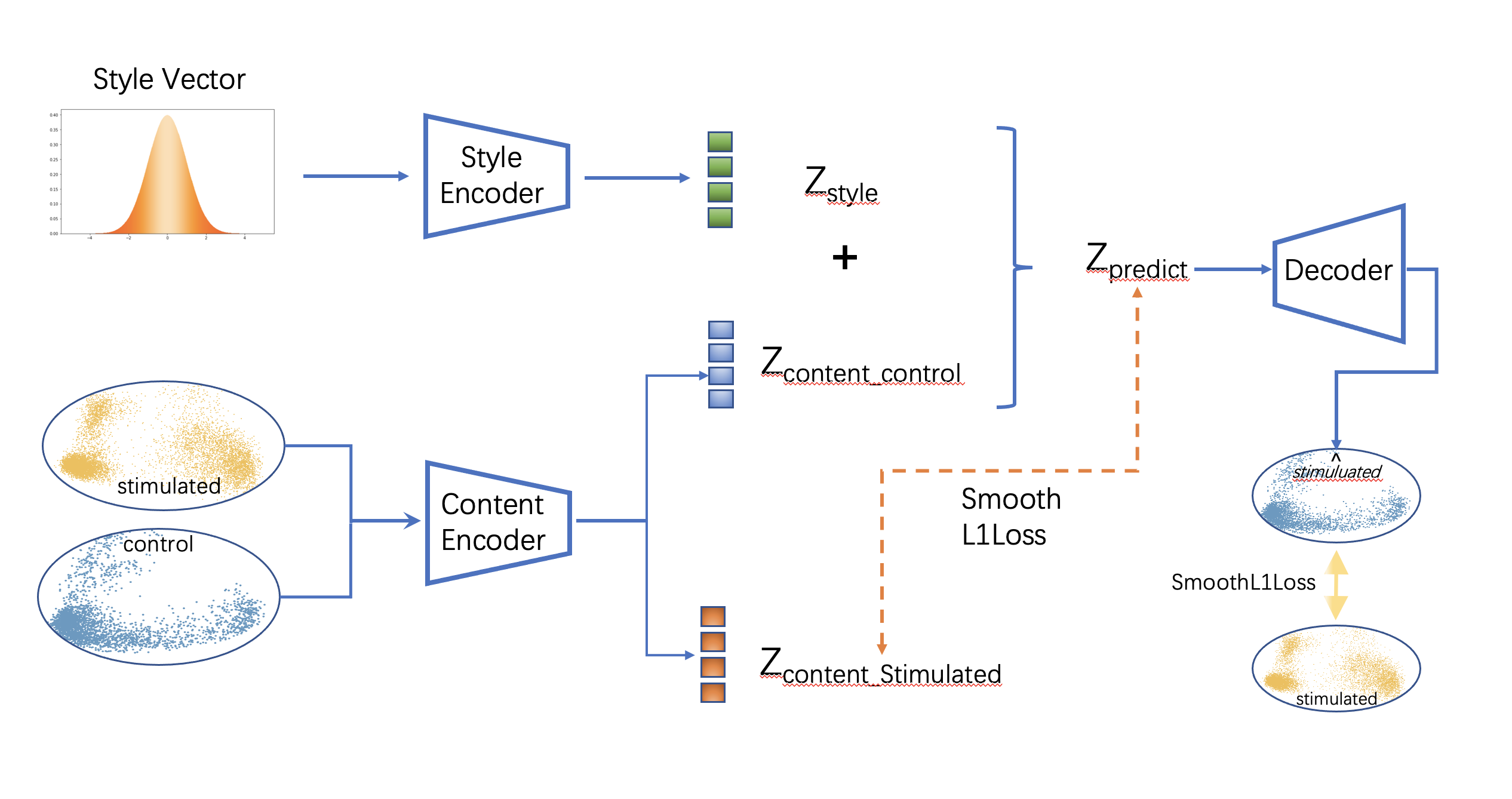
In scPerb, we evaluated the performance of our model under a fixed seed of 42 by using the square of the , which is calculated by the function[4]. This metric measures the correlation between the generated images and the ground truth data. We computed the values for all genes’ mean and variance and the 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 [5] function, employing the Wilcoxon method[6].

Through these analyses, we aimed to assess the accuracy and performance of our ScPerb model in generating realistic 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.

1. **Results**

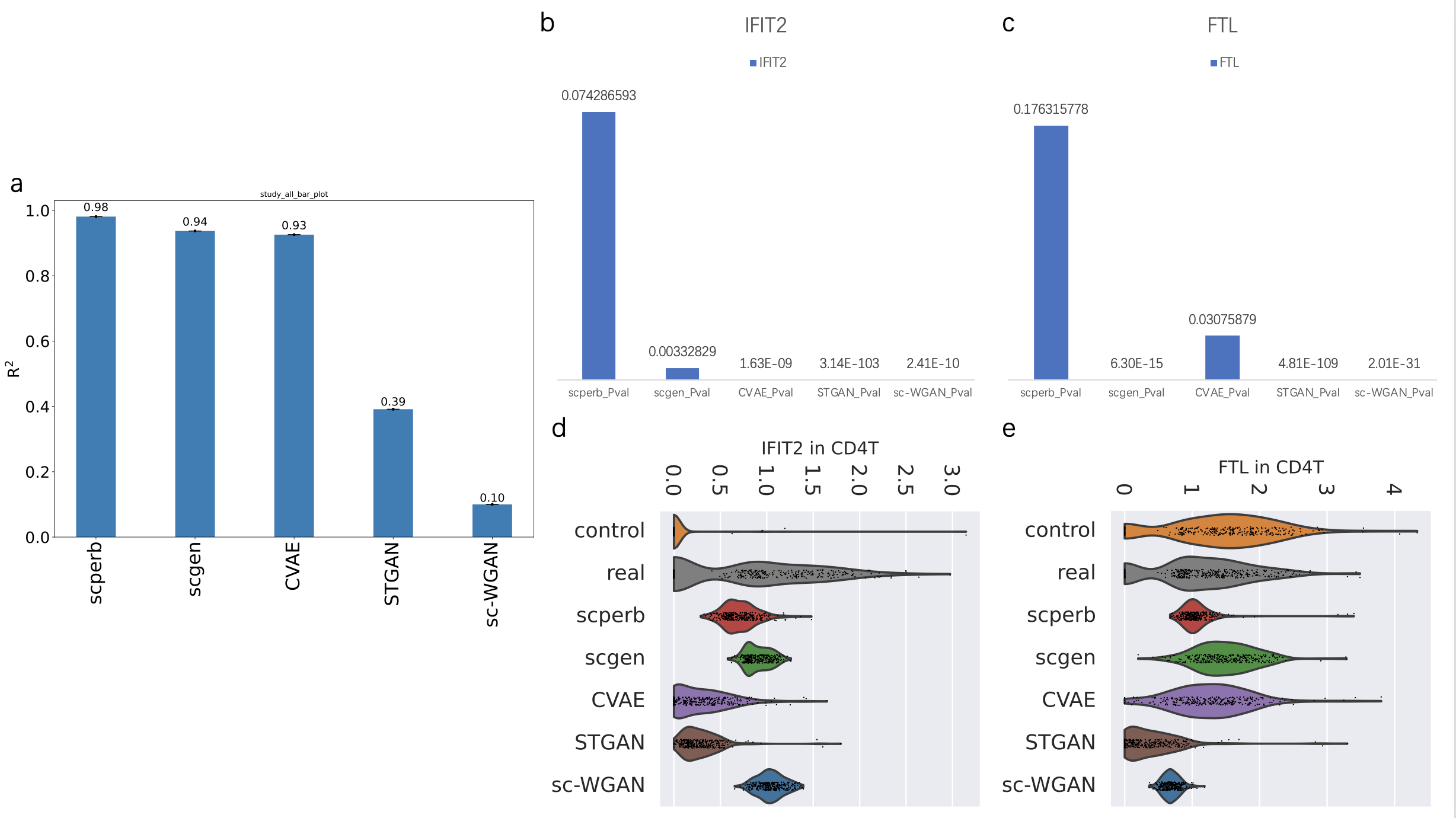


4.1 scPerb is an innovative generative model that can accurately predict single-cell perturbation responses.

The gene expressions from High-dimensional scRNA-seq are highly correlated with the single-cell responses. Traditionally, the algorithms focused on extracting the principle components from the gene expressions, along with the graph-based constraints[7, 8]. Such methods were limited by the poor representation of the limited principle components, which only explore a limited portion of the raw data. More recently, neural networks have shown their potential in projecting the observations from raw data into the high-dimensional manifold, where the latent representations can approximate the real distribution beyond the raw data observations. The neural networks were data-driven methods, and can dynamically assign weights to the dominating features from the raw data. In particular, Variational Auto-Encoder (VAE)[1] assumed the raw observation can be sampled from a multivariate normal distribution, and used an encoder to estimate the mean and variance of the distribution, resample the latent variables and used the variational inference to generate new data observations from the estimated distribution. Conditional Variational Auto-Encoder (CVAE)[9] introduced more constraints to the neural network, allowing the end-users to generate more desired reconstructed samples for customized demands. In particular, scGEN[3] assumed a fixed linear gap between the control cells and the stimulation cells, calculated the latent difference from both datasets, and predicted the stimulation cell response using latent representation from control cells and the calculated. Meanwhile, the Generative adversarial network (GAN)[10] introduced a creative but more black-box style in generating the data. GAN trained a “generator” to produce realistic “fake” data samples, while adversarially trained a discriminator to differentiate the “fake” data samples from the “real” data observations. The ultimate goal of GAN was to have a “generator” which can “fool” the discriminators when the loss converged. sc-WGAN[11] introduced GAN to the scRNA-seq data and obtain a universal representation manifold for different cell types. sc-WGAN[11] can predict the single-cell perturbation response from the well-trained latent representations. Other researchers also introduced the style-transfer GAN (st-GAN)[2], which aimed to transfer the dataset-related styles from the control dataset to the stimulation dataset.

In our manuscript, we presented a novel tool named “scPerb” (shown Fig.1), which was inspired by the VAE[1] and style-transfer GAN[2]. Suppose we have two datasets with different “styles” but the same cell types. We denoted to represent the ith cell from the control dataset, and for the jth cell from the stimulation dataset. scPerb decouples the perturbation task mentioned above into two steps: estimate the latent features of the cell types and a learnable dataset-related style transformation matrix. Inspired by the VAE architectures, scPerb first estimates the multi-variance normal distribution of the latent cell type feature c. Inspired by the style-transfer GAN[2], scPerb uses a neural network to learn the style transformation matrix from the dataset. Compared with scGEN, which used a fixed vector to transfer the latent features from the control cells to the stimulation cells, scPerb introduced more learnable parameters and allowed the neural network dynamically assign the weights of the “style-transfer” vector based on the data. Therefore, scPerb can better learn both the style and content difference between the control and stimulation datasets, and output a better prediction compared to scGEN.

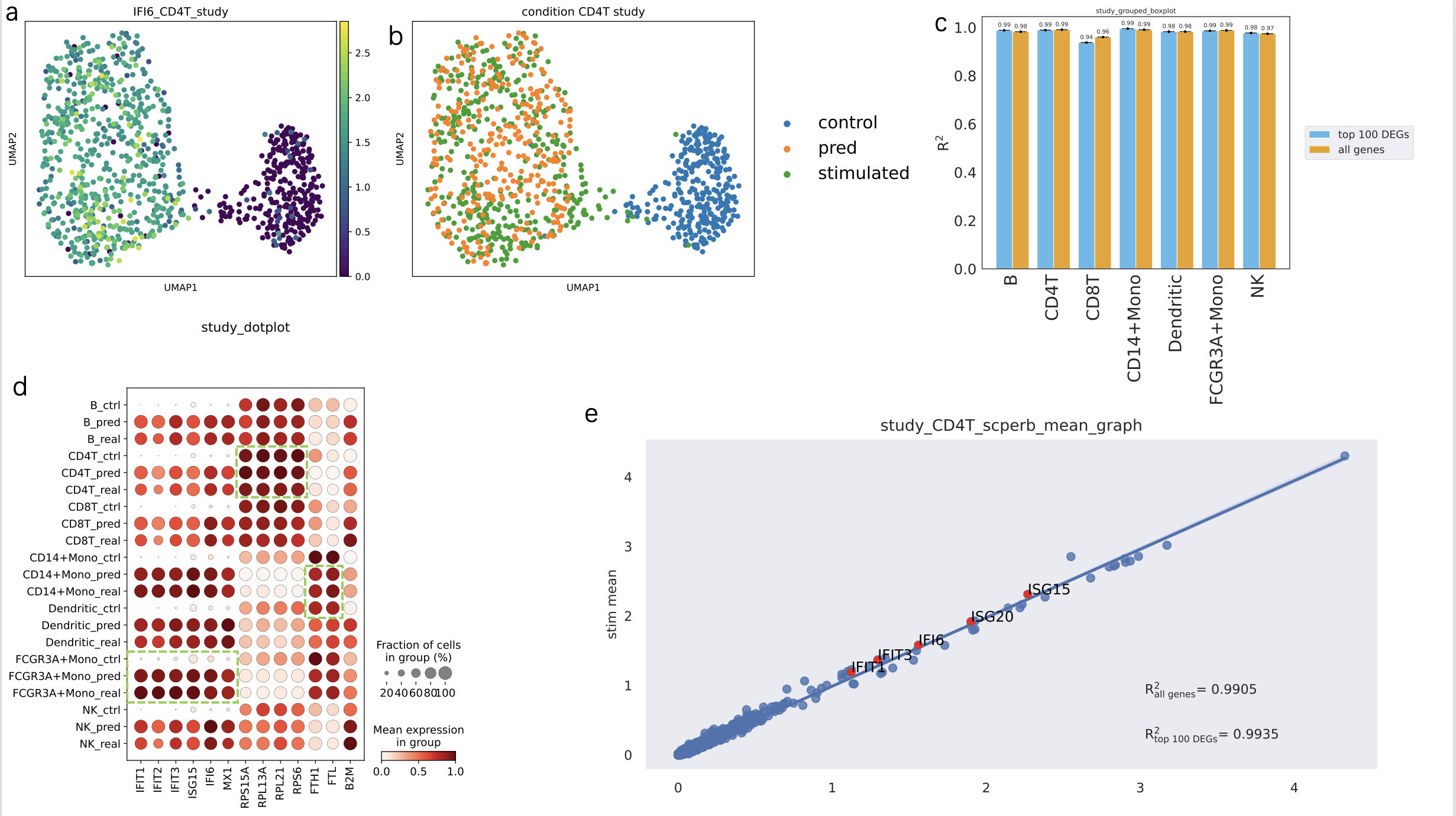
To demonstrate the performance of scPerb, we applied it to three datasets. Among these three datasets, two of them are two groups of published human peripheral blood mononuclear cells (PBMC) datasets[12, 13] stimulated with interferon () methods, and the rest is a group of intestinal epithelial cells fetched by parasitic helminth H.poly cells[14]. We fairly compared our proposed scPerb with other benchmarking papers [2, 3, 9, 11]. In this process, we first run all the cell types of one dataset in each model and combine the results for further processing. Then, we compare the prediction of all the methods with the ground truth, and the stimulation cells in the dataset, and get a final score. Compared to all the other methods including scGEN, CVAE, style-transfer GAN, and sc-WGAN, the predictions of scPerb are the most correlated with the cell types in the simulation dataset. In the published human peripheral blood mononuclear cells (PBMC) dataset, scPerb gain a mean value of 0.98, while scGEN and CVAE only achieved 0.94 and 0.93 respectively. Moreover, both GAN-based methods style-transfer GAN and sc-WGAN poorly predicted the perturbation response, resulting in values of 0.39 and 0.10 accordingly. In conclusion, scPerb best correlates the stimulation cells among all the other benchmarking methods.



4.2 scPerb outperforms other benchmarks

In the Study“命名有问题” dataset (Fig2 (a)), we measure the mean among all cell types. Each reflected the specific correlation score between the prediction and the real stimulation cell type. scPerb achieved a mean of 0.98, which is higher than the second-best benchmarking scGEN () and the third-best benchmarking CVAE (). Surprisingly, the GAN-based methods had much worse performance. The stGAN and sc-WGAN only have and accordingly. When comparing the performance with a specific cell type , scPerb achieved =0.99, followed by scGEN and CVAE with and . stGAN and sc-wGAN had poor performance, with and .

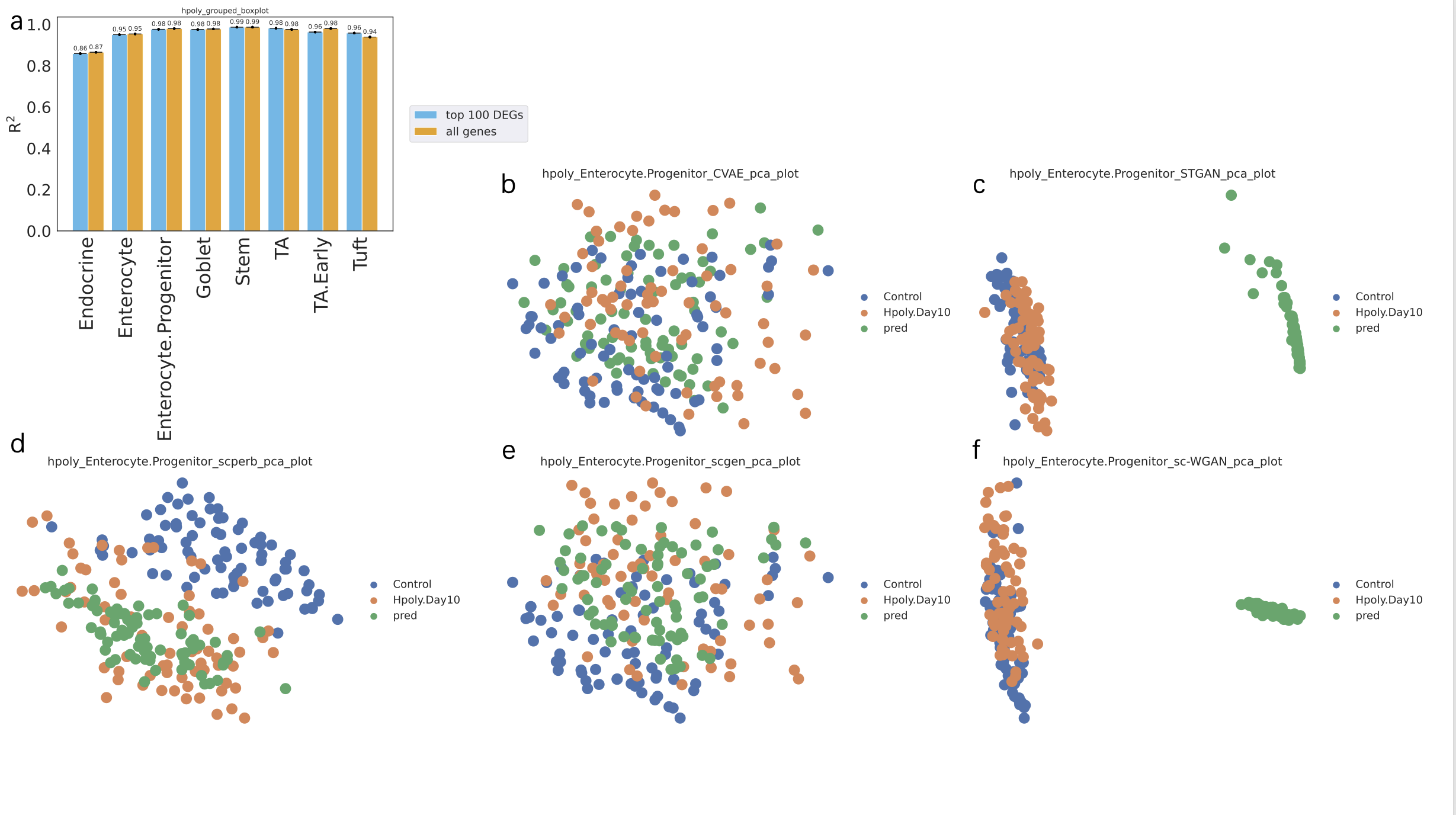
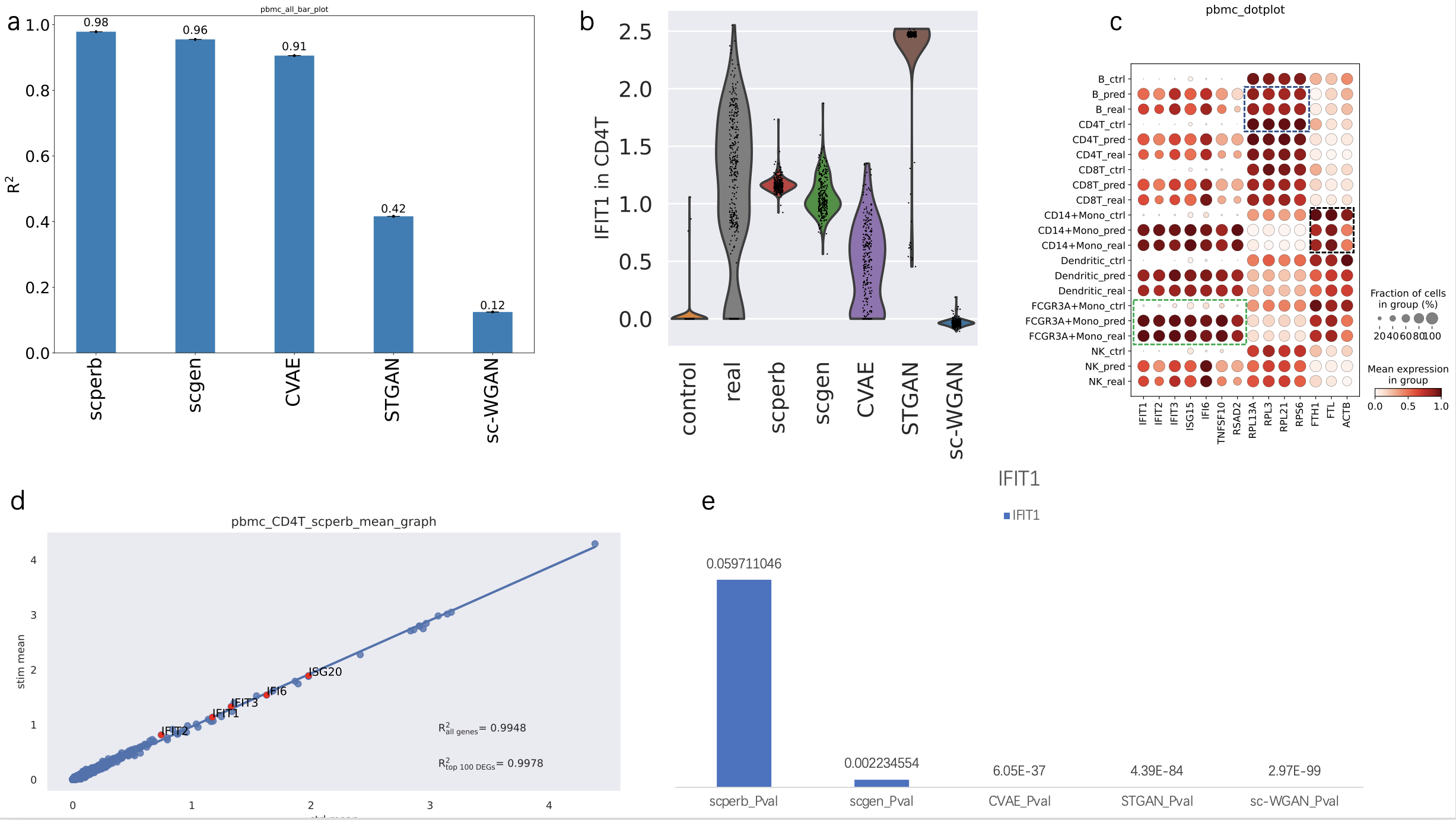
Moreover, we inspected the correlation of a specific gene expression among the benchmarking papers. is **“the relation between FTL and CD4T, need refers 需要FTL和CD4T对应的参考文献”**. In our dataset (Fig2 (e)), the gene expression of FTL in the control dataset is similar to the stimulation dataset. In this case, our scPerb focused on the mean of the gene expression in the real dataset and had a few high expressions. In contrast, scGEN, and CVAE were more similar to the gene expression in the control dataset, while st-GAN and sc-WGAN had relatively smaller gene expression. To support our observation, we also inspect another gene expression in the same cell type “这里需要IFIT2和CD4T对应的参考文献”. The gene expression of from the control dataset is full of zero values, while the gene expressions of in the stimulation dataset had higher values.(Fig 2(d)) In this case, scPerb, scgen, and sc-WGAN can reflect the dataset difference, while CVAE and stGAN failed to transfer the gene expression from the control dataset to the stimulation dataset. Furthermore, we applied a Wilcoxon test[6] to examine whether the prediction and the real stimulation gene expression had significant differences. The p values showed that scPerb had 0.1763 in the prediction of and 0.0742 in the prediction of , which failed to reject the hypothesis that the prediction and the real stimulation gene expression had significantly difference. Meanwhile, in the rest of the benchmarks, the P values of scGEN were and , CVAE (pvalue=0.0307 for FTL and pvalue<0.0001 for IFIT2). For the GAN-base methods, all of the P values were smaller than 0.0001. Consequently, the P values from the benchmarking papers reject the hypothesis and indicate a significant difference between the prediction and real stimulation gene expressions. (Fig 2(b) and Fig2(c))



4.3 scPerb can accurately predict the perturbation of cells

Figures 3 (a and b) displayed the umap of IFI6, the most differently expressed gene in the Study dataset (名字有问题，而且可能需要增加DEG的作用的文献). From the umaps, it is clear that the gene expression of the control cells was low and that of the stimulated cells was high; even so, scPerb still made excellent predictions as in Figure 3 (b), the prediction cells still correlate well with the stimulated cells. The result can be enhanced by the dot plot in Figure 3 (d), as for gene IFI6 and CD4T cell type, the mean expression is low for the control cell type, and that of both the stimulated cells and predicted cells were high. Moreover, the mean expression of the stimulated cells and predicted cells were approximately the same, which proved the prediction of scPerb made a good prediction. Figure 3(e) showed the scatter plot of the mean of all genes and the top 100 DEG genes in one of the best cell types, CD4-T. The mean of the prediction gene expression achieved among all genes, and among the top 100 DEG genes. In particular, we found that the top 5 DEG genes were . (可能需要增加DEG的作用的文献)

We then explored the performance of scPerb in a larger range of genes in more cell types. Figure 3 (c) explored the among different cell types. Specifically in this study, we examined B cells, CD4-T cells, CD8-T cells, CD14 + Mono cells, dendritic cells, FCGR3A+Mono cells, and NK cells. Remarkably, when considering all genes, B cells displayed an exceptional R² of 0.99, with the top 100 differentially expressed genes (DEGs) yielding a slightly lower yet highly significant value of 0.98. Similarly, CD4-T cells showcased an impressive consistency in prediction accuracy with R² scores of 0.99 and 0.99 for all genes and top 100 DEGs, while CD8-T cells demonstrated strong predictive power with R² values of 0.96 and 0.94. Moreover, NK cells exhibited robust R² scores of 0.97 and 0.98. Notably, CD14+Mono cells, dendritic cells, and FCGR3A+Mono cells collectively exhibited outstanding R² values of 0.99 and 0.99, further underscoring the reliability of predictions. This comprehensive evaluation delineates the remarkable predictive performance of the model across various cell types, shedding light on the intricacies of gene expression patterns and offering insights of potential significance in diverse biological contexts.



4.3 scPerb has robust results across different datasets.

scPerb has robust result in multiple datasets. In PBMC(名字) dataset [12], 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, style transfer GAN with 0.42 and sc-WGAN with 0.12 (Fig 4(a)). Meanwhile, scPerb precisely predicted the result of CD4-T cells, reaching of 0.9948 and 0.9978 respectively for all genes and its top 100 DEGs.(Fig 4(d)) This scatter plot further proved the strong prediction ability of scPerb. Moreover, in IFIT1, one of the top DEGs in CD4-T, 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 4(b)) The Wilcoxon test can further explain the difference between the prediction and the real stimulated cells in the IFIT1 gene: only scPerb achieved a P value of 0.0597, 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 0.05, showing a significant difference between their predictions and the ground truth. (Fig 4 (e)) Besides, the dot plot (Fig 4(c)) shows that scPerb can get robust prediction no matter the original control gene expression is lower (for example the IFIT1 gene), approximately the same (for example the RPL13A gene), or higher (for example the FTH1 gene).

In other datasets, such as the Hpoly dataset (名字), the power of accurate prediction of scPerb still remains. For most of the cell types in Hpoly dataset, scPerb gained a higher or equal to 0.95. Notably, in Stem cells, scPerb got the most accurate prediction, resulting in a of both 0.99 in all genes and the top 100 DEGs. It is then followed by Entrerocyte.Progenitor cells, Goblet cells, and TA cells, with both 0.98 for all genes and the top 100 DEGs.(Fig 5(a)) Moreover, scPerb made better predictions in this dataset, especially in Stem cells. In Figure 5 (b-f), the umap of all five methods clearly showed this point. In these figures, the prediction of scPerb (the green dots in the graph) fitted the best to the stimulated data (the orange dots in the graph), while other VAE-based methods such as scGEN and CVAE had a prediction that has approximately the same distance between the control data and the stimulated data, resulted in a worse prediction than scPerb. Moreover, both GAN-based methods style transfer GAN and sc-WGAN totally failed to predict the stimulated data, resulted in a prediction far away from the ground truth. (Fig 5(b-f))

**Reference**

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

2. 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.

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

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

5. 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.

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

7. Bendall, S.C., et al., *Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development.* Cell, 2014. **157**(3): p. 714-725.

8. Wolf, F.A., et al., *PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells.* Genome biology, 2019. **20**: p. 1-9.

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

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

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

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

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

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