Evaluating the Utilities of Foundation Models in Single-cell Data Analysis

Tianyu Liu[1,2], Kexing Li[2,3], Yuge Wang[2], Hongyu Li[2], Hongyu Zhao[1,2*]

[1]Interdepartmental Program in Computational Biology & Bioinformatics, Yale University, CT, USA.
[2]Department of Biostatistics, Yale University, CT, USA.
[3]School of Life Sciences, Tsinghua University, Beijing, China.

*Corresponding author(s). E-mail(s): hongyu.zhao@yale.edu;
Contributing authors: tianyu.liu@yale.edu; kexing.li@yale.edu; yuge.wang@yale.edu; hongyu.li@yale.edu;

Abstract

Foundation Models (FMs) have made significant strides in both industrial and scientific domains. In this paper, we evaluate the performance of FMs in single-cell sequencing data analysis through comprehensive experiments across eight downstream tasks pertinent to single-cell data. By comparing ten different single-cell FMs with task-specific methods, we found that single-cell FMs may not consistently excel in all tasks than task-specific methods. However, the emergent abilities and the successful applications of cross-species/cross-modality transfer learning of FMs are promising. In addition, we present a systematic evaluation of the effects of hyper-parameters, initial settings, and stability for training single-cell FMs based on a proposed **scEval** framework, and provide guidelines for pre-training and fine-tuning. Our work summarizes the current state of single-cell FMs and points to their constraints and avenues for future development.

Keywords: single-cell data, large language model, foundation model, deep learning, benchmark

1 Introduction

Single-cell sequencing technologies offer high-throughout observations into complex biological systems at the cell level with multimodal data [1, 2]. They help elucidate disease mechanisms and potential treatments [3–5]. In line with the central dogma of molecular biology, these technologies enable the characterization of various molecules, such as DNA (e.g. scDNA-seq) [6], RNA (e.g. scRNA-seq) [7, 8], and proteins (e.g. Cite-seq) [9]. Furthermore, single-cell sequencing can facilitate epigenetic studies, including chromatin accessibility (e.g. scATAC-seq) [10, 11] and methylation [12, 13]. These technologies have been rated as among the most impactful ones in recent years [14, 15].

With the development of single-cell technologies, expansive single-cell datasets have been collected, and they present challenges in analysis, integration, interpretation, and downstream utilization. Therefore, we have the need to build a FM for single-cell data analysis. Similar to single-cell data, natural language processing (NLP) also boasts extensive datasets, where FMs like pre-trained Large Language Models (LLMs) have shown great success in addressing NLP tasks or multimodal tasks [16]. Numerous LLMs, including GPT-4 [17] and LLaMA [18], excel at diverse language-related tasks such as question answering and sentence generation, which has received widespread attention from both the AI community and society [19]. Moreover, these LLMs showcase impressive performance in zero-shot learning, thereby enabling them to address tasks beyond their original training scope, such as solving mathematical problems [20].

Indeed, there are notable parallels between studies based on single-cell sequencing data and those in NLP. Firstly, both have numerous downstream tasks that are intimately connected with the multimodal data [21, 22]. Secondly, both require contemplation of intra-data and inter-data relationships [23, 24]. Thirdly, both benefit from high-quality databases [16, 25].

While FMs have seen marked success in the realms of DNA analysis [26] and biomedical NLP [27, 28], their application in single-cell research remains largely uncharted. There is a limited number of robust pre-trained models (known as single-cell FMs) capable of managing multiple tasks in single-cell research. Some single-cell FMs focus on cell-type annotation or gene function prediction, including scBERT [29], CellLM [30], and Geneformer [31], while others aim to create a foundation model in this area that can handle multiple tasks, including scGPT [32], scFoundation [33], tGPT [34], GeneCompass [35], SCimilarity [36], UCE [37] and CellPLM [38]. Details of these models can be found in Appendices D and E. Furthermore, no studies to date have comprehensively evaluated the utility of these models and provided guidance for model training. Little has been done to compare NLP-focused LLMs with those used for single-cell research to gain insight into emerging abilities and zero-shot (or few-shots) learning abilities of the latter.

Here we investigate the overlap of the pre-training stages of different single-cell FMs, and present a framework for assessing various single-cell FMs and tasks (shown in Figure 1 (a)), termed as Single-cell Large Language Model Evaluation (**scEval**), shown in Figure 1 (b). Using scEval, we identify critical parameters and strategies for the fine-tuning process of single-cell FMs. We also examine the potential emergent abilities of single-cell FMs, substantiating that the latter also possess distinctive abilities. To help the audience better understand FMs, we provide a glossary summary of common

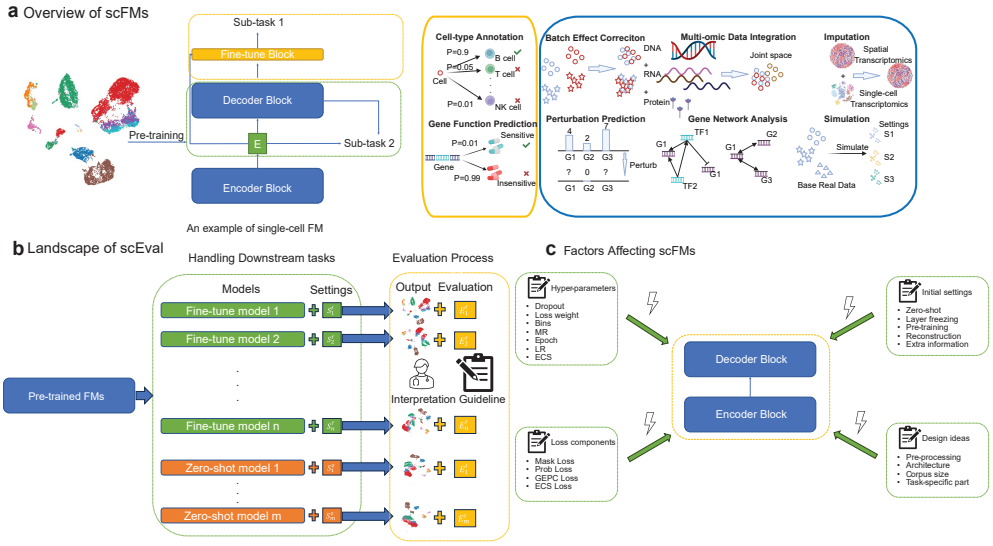
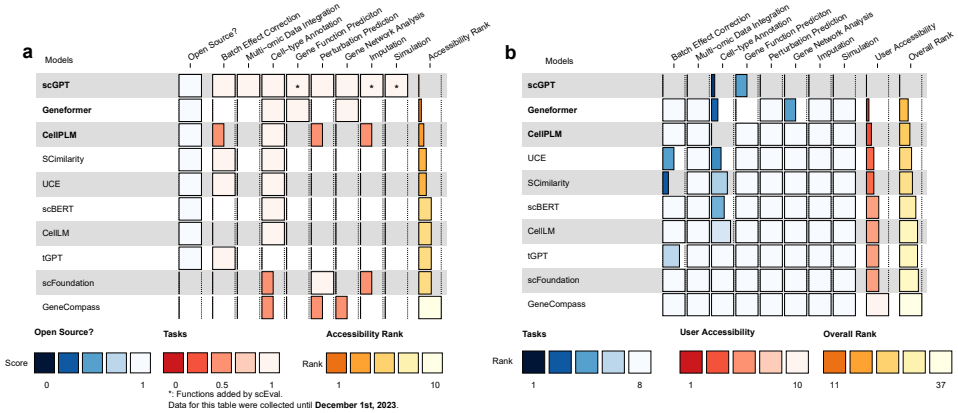


Fig. 1: Overview of single-cell FMs, landscape of scEval and factors affecting single-cell FMs. (a): Overview of single-cell FMs describing the typical structure of FMs and general tasks for single-cell data analysis. The right two blocks represent two types of downstream tasks. Yellow block: Sub-task 1, including Cell-type Annotation and Gene Function Prediction (top to bottom). Blue block: Sub-task 2, including Batch Effect Correction, Multi-omic Data Integration, Imputation (From left to right, top row), Perturbation Prediction, Gene Network Analysis, and Simulation (From left to right, bottom row). (b): The landscape of scEval shows the workflow of our systematic evaluation. Here we consider models based on fine-tuning design with settings S_1^f to S_n^f, and they have corresponding evaluation pipelines E_1^f to E_n^f. We also consider models based on zero-shot-learning design with settings S_1^z to S_m^z, and they have corresponding evaluation pipelines E_1^z to E_m^z. The reason for using the above notations to represent the process of evaluation is that different models can perform different tasks, while different tasks have different evaluation scenarios. (c): Factors which can affect the performance of single-cell FMs. These factors are summarized based on different designs of single-cell FMs. The known factors can be classified into four different types. Details of hyper-parameters can be found in Appendix A. Details of initial settings can be found in Appendix B.

terms in Artificial Intelligence (AI) and Machine Learning (ML) in Supplementary file 1.

2 Results



Tab. 1: Criteria to consider when choosing a model based on usability and performance. (a) Evaluations for the usability of different single-cell FMs. We have different scores for different conditions. We assigned **1** for open-source models and **0** for closed-source models in the evaluation of the open-source condition. For the task-level evaluations, the score **0** represents that the selected models do not meet the criterion because they do not have such design for specific tasks. The score **0.5** represents that the functions of selected models are described in the manuscripts but lack key steps for evaluation. The score **1** represents that the selected models have such functions and we evaluated their performance. We computed the final rank for user accessibility based on the sum of scores. The top three methods are boldfaced. (b) Evaluations for both performances and usability of different single-cell FMs. We record the ranks of different methods across different tasks and user accessibility evaluations and compute the final overall rank. A lower rank means a better model. The top three methods are boldfaced.

Overview of our evaluations. We evaluated the performance of eight open-source single-cell FMs (scGPT, Geneformer, scBERT, CellLM, tGPT, SCimilarity, CellPLM, and UCE) and two closed-source single-cell FMs (scFoundation and GeneCompass) by assessing their outputs on eight tasks with 25 datasets as well as their user accessibility. We did not evaluate all the models with all the datasets with the rationales of reasons provided in Supplementary files 2 and 3. The tasks that can be performed for different models as well as the overall ranks are summarized in Tables 1 (a) and (b). We also compared their performance with state-of-the-art (SOTA) task-specific methods. Our workflow can be summarized in Extend Data Figure 1. The top three methods include scGPT, Geneformer and CellPLM by considering both usability and performance. For each task, we discuss the effect of different parameter settings on model performance and investigate the contribution of different loss functions and initial settings by ablation tests. For the emergent abilities, we consider the contribution of model size to the performance of FMs. Finally, we evaluate the stability and usability of different single-cell FMs and make recommendations for preferred models. The detailed design of scEval is explained in the Methods section. To ensure fairness,

we selected metrics for different tasks based on their benchmarking analysis and we fixed all models to be from versions before December 1, 2023.

Comparisons of pre-training datasets for different single-cell FMs. In this section, we investigated and compared the pre-training steps of different single-cell FMs from three dimensions, including the scale of pre-training datasets, the diversity of pre-training datasets, and the major human tissues or organs overlap of pre-training datasets. Based on Extended Data Figure 2, we found that GeneCompass had the largest pre-training datasets, followed by scFoundation. However, both of them are closed-source. Interestingly, the pre-training scales of UCE, scGPT, Geneformer, SCimilarity, and tGPT are comparable, which implies that having 20-30 million may be acceptable for constructing pre-training datasets. Moreover, SCimilarity had the largest diversity in the datasets conditioned on diseases compared with other methods. Most of the single-cell FMs chose to include cancer cells in their pre-training process, which was possibly due to many single cell studies focused on cancer [39]. UCE had the largest diversity in the datasets from different species, while the rest of the single-cell FMs focused more on human. In addition, few models considered the use of multi-omic data information, except tGPT, CellPLM, and UCE. However, in the pre-training process, UCE and tGPT omitted the extra information (for example, the spatial location of spatial transcriptomic data) provided by these datasets and treated the data the same as scRNA-seq data. Using different pre-training pipelines for multi-omic datasets might improve the performance of current models on addressing related tasks. In Extended Data Figure 3, we computed the overlap of major human tissues or organs used for pre-training across different models. Geneformer had the largest diversity in this comparison because we treat it as a reference method. UCE had the largest overlap compared with Geneformer. scBERT and CellLM shared major human tissues or organs because they both used PanglaoDB [40] for pre-training. Interestingly, scGPT did not use many types of tissues for pre-training. Considering its relatively better performance, including as many types of datasets as possible may not necessarily lead to better performance, and data ablation analysis is needed for the pre-training step.

Furthermore, we analyzed the relationship between these statistics and model performance. Here, we considered two major tasks, batch effect correction, and cell-type annotation, for our comparison. We compared four FMs for batch effect correction and seven FMs for cell-type annotation, thus offering statistical meaning to perform further investigation. For each task, we computed the Spearman correlation under three settings, including performance vs. pre-training data scale, performance vs. the number of major tissues for pre-training, and performance vs. the number of model parameters. However, all the six correlations computed based on the comparisons did not show statistical significance (p-value ¿ 0.05). Therefore, the relationship between the performance of single-cell FMs on downstream tasks and their pre-training settings might be affected by many factors, including pre-training strategies, data-cleaning pipelines, rules of training-testing datasets splitting, and others. Therefore, we focus on task-driven analysis and offer guidelines for model development based on their performances on different tasks.

Evaluation based on cell-perspective tasks shows that single-cell FMs can reduce batch effect and annotate cell types.

Fig. 2: Experimental results of single-cell FMs and benchmarking methods for cell-level tasks. (a): An overall assessment of raw data and data after batch effect correction based on different methods. scGPT v1 represents scGPT model with smaller pre-training datasets compared to scGPT. (b): An overall assessment of raw data and data after multi-omic data integration based on different methods. (c): Comparison among models in the Cell-type Annotation task. The scores on the left represent the average accuracy of different models across different datasets. (d): Ablation tests of the loss function components for cell-type annotation. The red components are significant ($p - value < 0.05$).

Batch Effect Correction. We considered scGPT, tGPT, UCE, SCimilarity, Harmony [41] and ResPAN [42] for this task. We also provided a detailed analysis of the influence of various hyper-parameters on the performance of scGPT on batch effect correction. The description of the evaluation metric S_{final} can be found in Appendix C. We computed S_{final} based on the weighted average of metrics for evaluating the level of batch effect removal and the metrics for evaluating the level of biological variance conservation. As shown in Figure 2 (a), scGPT outperformed ResPAN in one of the 11 datasets and outperformed Harmony in two of the 11 datasets, while Harmony had an overall best performance. In the comparison of FMs, scGPT v1 had the best performance, and scGPT v1 outperformed the scGPT model overall, raising the issue of the need for increasing the size of pre-training datasets for this task. Moreover, only Harmony and ResPAN achieved better S_{final} compared with the raw datasets on average. Therefore, the ability of FMs (including scGPT, tGPT, UCE, and SCimilarity) to remove the batch effect is limited. Moreover, these FMs had worse performance in reducing the batch effect for large-scale datasets. One reason is that their biology conservation scores were lower than those of raw datasets (shown in Extended Data Figure 4). UCE could not handle large-scale datasets due to the out-of-memory (OOM) error. Extended Data Figures 5 and 6 show the Uniform Manifold Approximation and Projection (UMAP) [43] plots for the raw data and the scGPT results. We could still observe the batch effect in the output of scGPT for Pancrm, HumanPBMC, MCA, MHSP, DC, Lung atlas, Immune atlas, and Heart atlas datasets. Therefore, single-cell FMs also do not show advantages in handling atlas-level datasets for the batch effect correction task.

We provide a detailed analysis of the impact of various hyper-parameters on the performance of scGPT in batch effect correction based on Extended Data Figure 7. A smaller learning rate tended to lead to better performance across all datasets. The optimal number of training epochs varied across datasets, with a larger number of epochs being beneficial for most datasets. This result contradicts recent research advocating for a single-epoch training approach [44], suggesting that the optimal number of epochs might be context-dependent. Increasing the number of bins is generally associated with an increase in the final score. The impact of the mask ratio and dropout rate on model performance is unclear, suggesting that further investigation is needed to understand their influence. These observations may improve the application of scGPT for batch effect correction in single-cell data analysis and may also inform fine-tuning of other similar models.

Extended Data Figure 8 (a) presents the comparison of scores across different initial settings for the batch effect correction task using scGPT. We can see that scGPT is capable of performing zero-shot learning in batch effect correction. For the Cell Lines dataset, the zero-shot learning approach even achieved the highest score, indicating that it can be an effective method for certain datasets. Moreover, pre-training significantly contributes to the performance of scGPT in the batch effect correction task. Without pre-training, the model's performance notably decreased. Using cross-entropy as the loss function for gene expression reconstruction yielded better results than the mean square error (MSE) loss for most datasets. Freezing weights is not crucial for batch effect correction. Interestingly, the encoder structure

appears to play a more important role in the training process, as freezing the encoder layer led to a larger decrease in the score. Incorporating the cell type information as a human label into the training process enhanced performance for most datasets. These observations suggest that each dataset may require unique tuning, underscoring the importance of adaptable methodologies in single-cell RNA-seq data analysis.

Extended Data Figure 8 (b) shows the performance metrics versus the choices of different optimizers for the batch effect correction. Adam and AdamW [45] had comparable performance in this task, while SGD [46], Sophia-G [47] (a novel optimizer which is designed for training FMs) and Lion [48] (an optimized version of Adam) were worse. Therefore, the optimizers in the Adam family are preferred.

Extended Data Figure 8 (c) illustrates the impact of different loss function components on the performance of batch effect correction using scGPT. Here we consider Mask Loss (It means we mask the gene expression levels of some genes and reconstruct them to compute the loss), Prob Loss (i.e. we predict whether some genes are expressed in some cells to compute the loss), GEPC Loss (i.e. we predict the gene expression levels based on the cell embeddings to compute the loss), ECS Loss (i.e. we bring similar cells closer together and push different cells farther apart in the embedding space to compute the loss), and one task-specific loss function, known as the gradient reverse loss (i.e. we reverse the gradient of batch label classifier to perform adversarial training). We compared the scGPT without certain loss functions with its default mode to perform ablation tests. Using all components of the loss function did not always yield the best results, with the exceptions of the Pancrm, MCA, and MHSP datasets. Using only the gradient reverse loss function resulted in the worst performance. The GEPC Loss seemed to play a crucial role in the performance of the batch effect correction task. These results suggest the need for a careful composition of the loss function when training single-cell FMs for batch effect correction, with each loss function component contributing differently to model performance.

Multi-omic Data Integration. For this task, we seek to integrate scRNA-seq datasets with scATAC-seq datasets. We considered scGPT, scJoint [49], and scGLUE [50] for evaluation. We assessed the integration quality through the same score as Batch Effect Correction because these two tasks have similar targets, that is, integrating datasets from different domains while preserving biological variation. The results presented in Figure 2 (b) summarize the evaluation results for the multi-omics integration task. Overall, scGLUE surpassed scGPT and achieved the best performance. In addition, the Extended Data Figure 9 shows that the results of scGPT did not have better biology conservation scores than the raw condition across all the datasets. Therefore, there exists a performance gap between scGPT and task-specific models. Based on our analysis, scGPT performed better than scGPT v1, which implied that larger pre-training datasets might help integrate multi-omic datasets. We also observed batch effect for the results of scGPT based on UMAP plots shown in Extended Data Figure 10. In conclusion, single-cell FMs did not show an advantage in handling datasets for multi-omic data integration.

We illustrate the effect of initial settings in Extended Data Figure 8 (d). Different from the case with batch effect correction, the cross-entropy loss function led to worse performance compared to the MSE loss for this task. Interestingly, pre-training did not

significantly affect the performance for this task since training from scratch also had similar performance. The encoder part of the single-cell FM played a more important role than the decoder since we observed a larger performance drop by only freezing the encoder part. Including cell types or human labels in the training process proved beneficial, likely providing the model with more precise and useful information for the task. The zero-shot learning approach did not perform as well for this task as it did for batch effect correction. Therefore, we need more consideration for the design of single-cell FMs for this task.

We illustrate the evaluation metrics versus different parameter settings in Extended Data Figure 11. scGPT did not perform well on this task as shown by the low score (below 0.5) even if we tried to search for better hyper-parameters. Certain hyper-parameters affected the training process. A smaller weight for the loss function, a larger dropout rate, and more epochs improved the model's performance. Number of bins and mask ratio did not exhibit monotonicity, making it difficult to draw conclusions. Setting a high learning rate and ECS weight decreased the performance of scGPT. Moreover, based on Extended Data Figure 8 (e), the optimizers in the Adam family are still preferred because of better performance, which is similar to what we found for the batch effect correction task. Since the patterns discovered in the multi-omic data integration task are not exactly the same as what we discovered in the batch effect correction task, we believed specific design focusing on scATAC-seq data is required.

Cell-type Annotation. We considered CellPLM, Geneformer, scGPT, scBERT, CellLM, SCimilarity, UCE, and TOSICA [24] for this task. We assessed the performance of different single-cell FMs in assigning cell types based on the four metrics (Accuracy, Precision, Recall, and F1 score) discussed in Appendix C.2. These metrics are widely used in the evaluation for cell-type annotation. The UMAPs for the raw data and scGPT are shown in Extended Data Figures 12 and 13. We observed accurate annotation results on these UMAPs. Figure 2 (c) displays the Accuracy for different datasets for these models. On average, models with pre-training performed better than those without pre-training. However, CellLM did not perform well across all the datasets. CellLM had running errors for Pancrm, HumanPBMC, PBMC368k, and Liver cross datasets, which raises the issue of the reliability and usability of this model, possibly due to their model architecture or training setting. SCimilarity also did not perform well in this task, which might be due to their pre-training process which contained too many sub-cell types [36]. Moreover, for the intra-dataset prediction task, CellPLM, scGPT, Geneformer, and UCE were comparable although they had different pre-training settings. For the inter-dataset prediction task, CellPLM, scGPT, Geneformer, and UCE were better than scBERT. For cross-species cell-type annotation, the performance of CellPLM, scGPT, Geneformer, UCE, and scBERT was comparable for the MCA dataset, but scGPT did not perform well for the MHSP and MB spatial datasets. Therefore, different single-cell FMs also had large divergences in performance, and CellPLM, scGPT, and Geneformer had good performance across different datasets.

In Extended Data Figure 14, we compared the performance of models with different hyper-parameter settings. Higher loss weight, learning rate, ECS threshold, mask ratio, and smaller epochs tended to lead to worse performance of scGPT. There was little

correlation between the number of bins and the performance of scGPT. We observed the consistency in the performance of different single-cell FMs under the condition of altering their shared hyper-parameters. For Geneformer and scBERT, a lower learning rate and higher epochs also tended to lead to better performance.

We also considered different initial settings for model training. The first setting is *Freeze all*, where we froze all the weights of pre-trained layers. The second setting is *Default*, where we used the default fine-tuning settings. The third setting is *From scratch*, where we did not use the pre-trained weights. Extended Data Figure 15 (a) shows the score versus initial settings across different datasets. Here we considered scGPT and scBERT. We omitted Geneformer because it requires pre-training weights as input. It can be seen that pre-training always improved results for scGPT, especially for the cross-dataset conditions. However, there was a little benefit of pre-training for scBERT. For both cases, freezing the pre-training layers and letting them not be involved in the fine-tuning process was not recommended. In some cases, the fine-tuning performance of such freezing was worse than training from scratch. Transfer learning for different species is possible because, for the MCA dataset, pre-training based on human data can help predict cell types for the mouse. For the same type of GPU, the training process of scGPT was faster than scBERT and Geneformer, with more GPU memory usage, according to Extended Data Figure 15 (b).

In Extended Data Figure 16, we froze the front layers or scGPT (from 0 to 11) to investigate the relation between the number of freezing layers and annotation accuracy. We found that freezing layers had comparable or even slightly better performance than full-parameter fine-tuning. Although the metrics under different numbers of freezing layers are close and high, we still found that freezing five layers can lead the best performance. Therefore, we can freeze parts of layers for cell-type annotation based on single-cell FMs.

In Extended Data Figure 17, we show the performance of scGPT based on different optimizers across four datasets. Adam, AdamW, and Lion were comparable, while SGD was worse than them but better than Sophia-G, which was not stable.

Moreover, we explored the contribution of different loss function components towards the cell-type Annotation task based on ablation tests, and the results are shown in Figure 2 (d). Here we included three extra metrics, and details can be found in Appendix C. Based on the Accuracy of Figure 2 (d), the inclusion of mask loss is important. Moreover, the default setting is generally good across different tasks. Based on precision and recall of Figure 2 (d), the effect of different loss function terms had less effect on precision and more effect on recall. Such a difference could affect the final F1 score. Removing the GEPC loss function terms improved the cell type prediction for the DC, MHSP, and MB Spatial datasets, and did not affect the prediction performance for the other datasets.

Therefore, most single-cell FMs can handle the cell-type annotation task with suitable pre-training data and model structure, but investigation for the pre-training framework is needed to understand their specific performance difference across all the datasets.

Evaluation based on gene-perspective tasks shows that single-cell FMs can handle tasks related to functions of genes.

Gene Function Prediction. We considered Geneformer, scGPT, *vanilla* NN based on raw data and *vanilla* NN based on Gene2vec [51] for this task. We evaluated the results with the same four metrics as the Cell-type Annotation task because they are both classification problems. The results are shown in Figure 3 (a). On average, Geneformer and scGPT performed well in this task. Moreover, the accuracy scores of scGPT and *vanilla* NN based on Gene2vec were comparable, while there was a performance gap between single-cell FMs and *Vanilla* NN based on raw data. Therefore, using initial gene embeddings with prior information is meaningful for single-cell FMs.

Figure 3 (b) and Extended Data Figure 18 show the accuracy of different hyper-parameter settings. Smaller learning rate and loss weight tended to lead more accurate results. Geneformer was more sensitive to Epoch compared to scGPT. For scGPT, pre-training contributed more than fine-tuning in this task as increasing epochs did not affect the model performance. Only tuning the number of bins, mask ratio, dropout rate, and ECS threshold did not affect the prediction results.

In Extended Data Figure 19 (a), we considered different initial settings for model training. It can be seen that pre-training always improved results for scGPT. Moreover, freezing the whole model did not affect the performance of scGPT.

Extended Data Figure 19 (b) shows the performance of scGPT based on different optimizers. Adam and AdamW were comparable, while Lion was worse than them but better than SGD and Sophia-G.

Extended Data Figure 19 (c) shows the ablation test results of scGPT for this task. There was no significant difference by comparing the default setting and those without certain components. Therefore, the task-specific loss function is the most important design for this task.

Perturbation Prediction. We considered scGPT, scFoundation and GEARS [52] for this task. In the training process, we tried to reconstruct the expression levels of all the genes of the input cells rather than only the masked genes, which was also the only loss function used by scGPT. We used Mean Pearson Correlation (MPC) as the metric to evaluate the performance of scGPT under different hyper-parameters or initial settings, and details can be found in Appendix C. MPC, proposed by GEARS, can assess the prediction performance at the gene level for both perturbed genes and unperturbed genes. The datasets include two perturbation conditions: a single-gene perturbation and a double-gene perturbation. Based on our experiments shown in Figure 3 (c), GEARS and scGPT were comparable in this task. Moreover, the running time of scFoundation for this task was too long to finish, which suggests the challenges of deploying FMs in solving this task. Moreover, since GEARS is already a well-performed method, the requirement of developing FMs for handling such task is not clear.

Figure 3 (c) also summarizes results for different initial settings of scGPT for the Norman, Adamson, and Dixit datasets. The default setting performed best for these datasets across different settings. This indicates that the initial configuration of scGPT works well for this task. The performance was comparable between training from scratch and training from pre-trained weights. Freezing the decoder part of the model performed better than freezing the encoder part, which implied that the encoder part is important for perturbation prediction. Interestingly, the ability of zero-shot

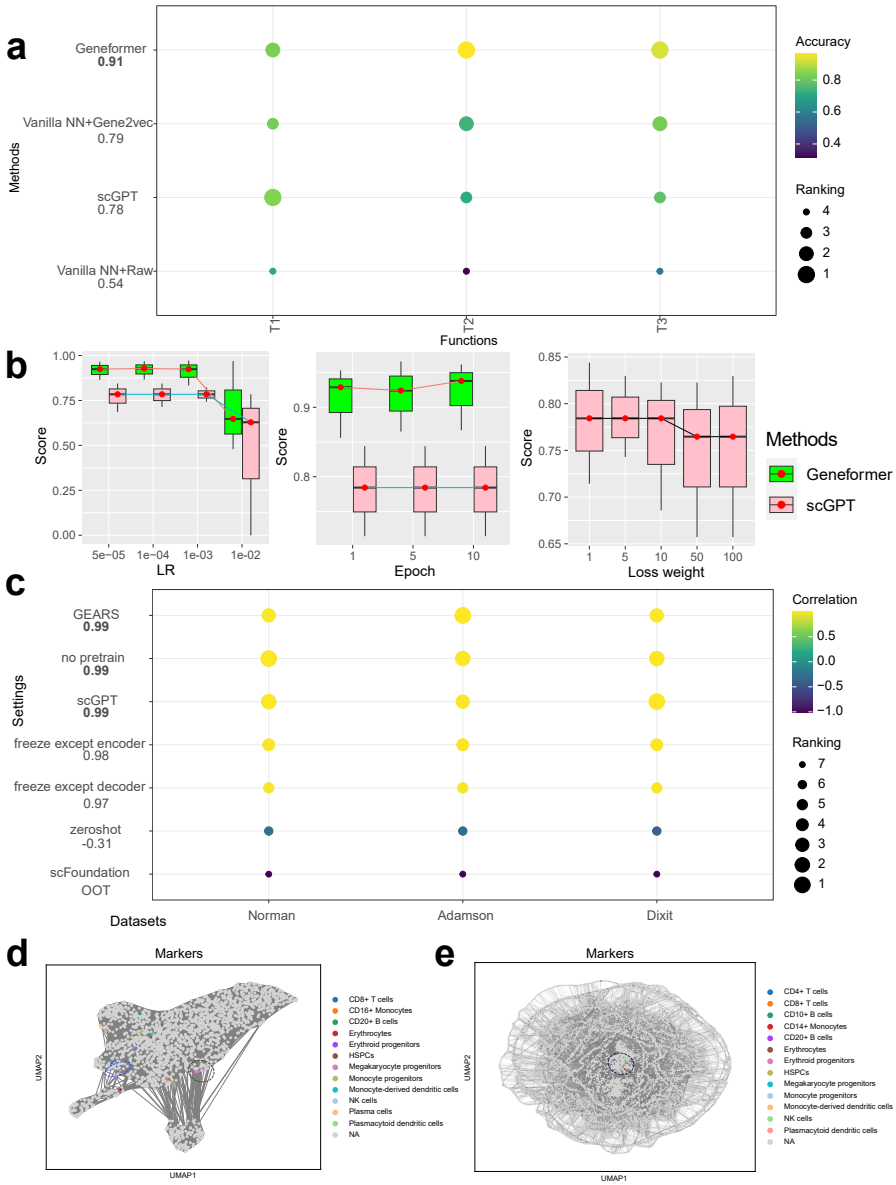


Fig. 3: Experimental results of single-cell FMs and benchmarking methods for gene-level tasks. (a): Comparisons among Geneformer, scGPT, and *vanilla* NN in the Gene Function Prediction task. (b): The effect of hyper-parameters including Loss weight, Bins, and Learning rate for scGPT and Geneformer in the Gene Function Prediction task. (c): Correlation of GEARs and scGPT under different settings across different datasets. A higher correlation means lower rank and better performance. The numbers corresponding to the settings represent the average value across two datasets. (d): Dataset-level gene embeddings from scGPT colored by the marker genes of different cell types. (e): Dataset-level gene embeddings from Geneformer colored by the marker genes of different cell types.

learning of scGPT towards this task was not suitable. Therefore, the contribution of using non-perturbed data as pre-training datasets for perturbation prediction related to perturb-seq datasets is not significant. We may need to investigate more types of perturbation, including drug-level conditions or disease-level conditions, to check the importance of introducing prior information for perturbation analysis.

Regarding the effect of hyper-parameters, Extended Data Figure 20 shows that scGPT is sensitive to adjusting the learning rate and epochs. Decreasing learning rate and increasing the number of epochs improved MPC. Higher learning rate caused running error for scGPT. We did not identify patterns for other hyper-parameters. Extended Data Figure 21 (a) shows that AdamW, Sophia-G, Adam, and Lion had comparable performance for scGPT in perturbation prediction. SGD could significantly reduce the performance of scGPT. Therefore, the use of single-cell FMs for perturbation prediction does not require a complicated design.

Gene Network Analysis. We considered scGPT and Geneformer for this task. Firstly, we clarified the difference between Gene Regulatory Networks (GRNs) and Gene Co-expression Networks (GCNs), and the networks we evaluated here were GCNs. In general, GRNs characterize causal relationships among genes. The GRNs defined in scGPT are in fact GCNs, because the construction process is based on embeddings similarity. For the inference of GCNs using scGPT and Geneformer, two types of GCNs can be defined: 1. Type 1 GCN (Tissue-specific GCN): This type of GCN is generated by applying the single-cell FM to the entire dataset under a zero-shot learning framework to create gene embeddings. The similarity is then computed to infer gene-gene relationships based on these embeddings. The quality of the GCN is evaluated based on the relationships between marker genes for different cell types. 2. Type 2 GCN (Cell-type specific GCN): This GCN is created by applying the single-cell FM to generate cell-type-specific gene embeddings under the zero-shot learning framework, and the same type of similarity is used to infer gene-gene relationships based on these embeddings [32]. The quality of this GCN is evaluated based on the Gene Ontology Enrichment Analysis (GOEA) [53] for specific gene sets. These GCNs can provide valuable insights into the understanding of gene interactions and regulation in specific tissues or cell types, which could have broad applications in biology and medicine [54, 55].

We used the Immune Human Atlas dataset to evaluate the performance of inferring these two types of GCNs. The known information including marker genes [56], cell types [56], and Reactome pathways [57] was utilized to evaluate the performance of scGPT on the GCN inferences.

In the analysis of the Immune Human Atlas dataset, we initially focused on Type 1 GCN, with the results presented in Figure 3 (d) and (e). The neighboring relationships within this data are colored according to the distribution of marker genes. We collected marker genes based on the source paper of this dataset [56] and filtered them to have marker genes with similar expression patterns for each cell type. As the gene-gene relationship was determined based on k-nearest neighbors, it can be viewed as a form of gene co-expression relationship. From Figure 3 (d), only marker genes from two cell types showed the co-embedded and isolated relationship. They are Monocyte-derived dendritic cells and Megakaryocyte progenitors. Figure 3 (e) shows that the

embeddings from Geneformer for maker genes were all co-embedded. Therefore, the embeddings of scGPT are better than the embeddings of Geneformer on preserving the cell-type-specific information. Extended Data Figures 22 (a) and (b), on the other hand, represent the cluster labels based on the Leiden clustering method [58]. These clusters can be interpreted as groups of genes that share common functions, or "gene co-function clusters". We first analyzed the cluster information from the gene embeddings from scGPT. For marker genes from other cell types, some of them are in different clusters shown in these two figures, and some genes are co-embedded with other cell types' marker genes. There are two isolated groups (9 and 12), but no marker genes are identified in these two groups. For the gene embeddings from Geneformer, we found that most of the marker genes from different cell types are embedded in the same cluster, while the rest of clusters did not contain much cell-type-related information. Therefore, the clustering result of scGPT was also more biologically meaningful than Geneformer, possibly due to the number of selected genes. Geneformer generated all genes for embeddings by default, while scGPT only focused on highly variable genes.

These analyses highlight how scGPT can be used to create gene embeddings to reveal important patterns of co-expression and co-functionality between different genes, offering insights into potential gene-gene interactions and regulatory relationships.

Extended Data Figures 22 (c) and (d) focus on gene embeddings categorized by cell types from scGPT, while Extended Data Figure 22 (e) shows the cell-type-specific gene embeddings from Geneformer. Extended Data Figures 22 (c) and (d) show that gene embeddings from different cell types tended to be co-embedded and there was no apparent difference. There was specific gene enrichment in the genes from Erythroid progenitors and CD 16+ Monocytes. The distribution of the remaining genes on the UMAP results was relatively random, and we also observed random distribution of gene embeddings from Geneformer. One reason could be that the quality of gene embeddings was unsatisfactory. Since scGPT and Geneformer adopt the zero-shot learning scheme embeddings, it does not incorporate cell-type-specific information for a specific dataset. The other reason might be that the complex biological network in the human immune system makes the communication between cell-cell or gene-gene difficult to decompose [59, 60]. Additional analysis is needed for generating gene embeddings.

This analysis illustrates the complexity of cellular functionality and the difficulty of clearly defining these relationships based on gene embeddings. Despite these challenges, the scGPT model still demonstrates its potential in identifying functional similarities between different cell types.

This study also involved evaluating and exploring GCNs based on the human immunology system quantitatively, which is known for its complexity due to interactions among various cell types. The original analysis of GCN from [32] focused on HLA genes and CD genes. We did not analyze HLA genes because they are highly polymorphic and thus carry a higher risk for having errors of reads[61–64], so the network result is not reliable. Therefore, we selected genes co-embedded with CD3 genes, and utilized the embeddings of these genes to compute the GCNs. The results of scGPT and Geneformer for CD3-related genes are shown in Extended Data Figures 22 (g) and

(h). The value above the edges represents the strength of correlation. The threshold of strong correlation here is set as 0.4. A publicly accessible database containing pathway information- the Immune System R-HSA-168256 pathway from the Reactome 2022 database [57] - was used as a reference for validation. The GCN was constructed based on the correlation between the two genes. For the CD3-related genes, there was nearly no overlap (7/1943) for scGPT as well as for Geneformer (8/1943), indicating poor inference of GCN. Moreover, for all the pathways involved in the selected gene set, we computed the ratio between the number of significant pathways and all pathways, and recorded the results in Extended Data Figure 22 (f). We found that gene embeddings from scGPT had a higher ratio, suggesting better quality of scGPT output.

Our results highlight the importance of critical evaluation and cross-referencing in the development of GCNs inference, as well as the potential and limitations of using single-cell FMs for this purpose.

Evaluation based on Imputation and Simulation Analysis shows that further improvement of single-cell FMs is needed.

Imputation. We considered scGPT and Tangram [65] for this task. The evaluation metrics for this task based on clustering and correlation are provided in Appendix C. The evaluation of clustering performance and gene expression correlation can assess the preservation of biological variation after imputing. We considered imputation for two different types of datasets, known as scRNA-seq datasets and spatial transcriptomic datasets. The imputation results for scRNA-seq are summarized in Figure 4 (a), which implies that the imputation function of scGPT for scRNA-seq data introduced more noise into the original sequencing data, suggesting the unreliability of the decoder's output.

According to Figure 4 (b), scGPT performed well in the spatial transcriptomic data imputation task compared to the SOTA spatial imputation method, Tangram [65, 66]. Based on the evaluation of correlation and significance proportion, the imputation results of scGPT are better than the results of Tangram. Moreover, the scores of these two metrics based on the zero-shot learning version were even better than the pre-training version with scRNA-seq data. However, based on the results of the average bio score evaluation, the raw data had better scores. This could be caused by the sources of the spatial clustering labels, which were generated from gene expression clusters rather than expert annotation. Such methods could introduce bias before and after imputation.

Extended Data Figure 23 shows the results of Differentially Expressed Genes (DEGs) based on pre-imputation data and post-imputation data. Results in Extended Data Figure 23 (a) showed that scRNA-seq imputation was not reliable because the expression patterns of all genes were similar after imputation based on scGPT. However, based on Extended Data Figure 23 (b), no mitochondrial (MT) genes were included in the DEGs after imputation based on scGPT. These genes were identified as DEGs in the raw dataset. A high proportion of MT genes is indicative of low-quality data, which means cells with such patterns are apoptotic or lysing [67]. Therefore, the MT genes should be omitted in the downstream analysis by filtering[68, 69]. Moreover, based on Extended Data Figure 23 (b) and (c), the DEG patterns after imputation

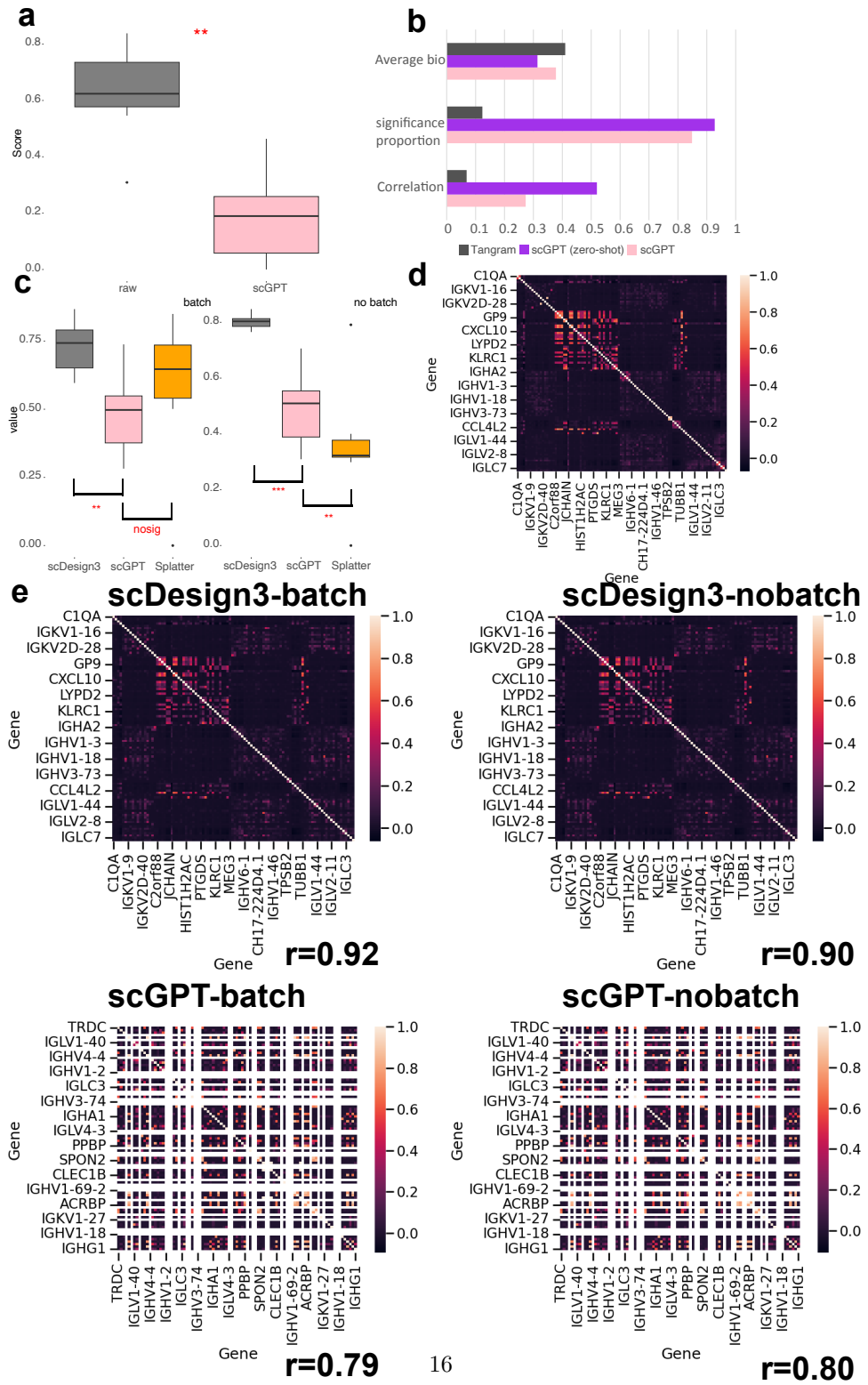


Fig. 4: Experimental results of the Imputation task and the Simulation task. The significance level was computed based on paired Student's t-test. The number of stars represents the significance level ($*** : p - value < 0.005, ** : p - value < 0.05$). (a): Comparison of the average bio score between the raw data and imputed data by scGPT in the scRNA-seq imputation task. (b): Comparison of the average bio score, average correlation score, and average significance level score among Tangram, scGPT, and scGPT (zero-shots) in spatial transcriptomics imputation task. (c): Comparison of the average bio score among scDesign3, Splatter, and scGPT for simulation. (d): Gene-gene correlation heatmap from the raw HumanPBMC dataset. We select the subset of the top 100 highly variable genes. (e): Comparison of different simulation methods by correlation. The heatmap represents the top 100 highly variable genes (for raw and scDesign3) or the subset of the top 100 highly variable genes (for scGPT) based on the HumanPBMC dataset. The correlation "r" represents the Pearson correlation between the gene correlation of raw data and the gene correlation of simulation data.

based on scGPT and Tangram are similar. Thus, scGPT has the potential to produce biologically meaningful imputation for spatial transcriptomic datasets.

Simulation Analysis. We considered scGPT, Splatter [70], and scDesign3 [71] for this task. We evaluated the output of scGPT against the output of scDesign3 and Splatter. For conditions incorporating batch effects, we employed the same metrics used in the evaluation of batch effect correction. In scenarios without batch effects, our metrics are primarily focused on assessing the preservation of biological information. As shown in Figures 4 (c)-(e), scDesign3 outperformed scGPT and Splatter across two conditions of the simulation task. However, scGPT performed better than Splatter for simulating datasets without batch effect. In particular, scDesign3 had a more pronounced superiority in generating simulation data without batch effects, in comparison to scGPT. This is consistent with the results shown in Figures 4 (d), (e) and Extended Data Figure 24. The gene-gene correlation from scDesign3 was also more similar to the gene-gene correlation of the raw data. Therefore, the simulation task needs to be improved for single-cell FMs as reference-based simulators.

In addition, we present UMAPs of the output produced by different methods in Extended Data Figures 25 and 26 that illustrate the advantage of scDesign3. The embeddings of scGPT with the no batch effect settings tended to preserve the batch effect, while the embeddings with batch effect tended to remove the batch effect. The embeddings of Splatter with no batch effect also mixed different cell-type information.

Single-cell FMs have emergent abilities but the stability of single-cell FMs should be improved.

Emergent Ability Analysis. In this section, we explored the emergent abilities of scBERT, Geneformer, and scGPT. Emergent abilities mean the rapid improvement for certain tasks under large-scale parameter size [72]. We considered three scenarios to investigate the emergent abilities: cross-data cell type prediction, cross-species analysis, and spatial transcriptomics analysis.

Inspired by [33], we compared models with large parameter size of pre-training to models of small parameter size. For all tasks except spatial data batch effect correction, we compared the performance of the three examples with that of *Vanilla* neural networks (NNs) to identify any emergent abilities. As for the task of spatial data batch effect correction, we examined the statistics derived from our correction evaluation to detect any emergent abilities.

In the first scenario, our anticipated emergent ability would be a significant improvement in prediction accuracy for single-cell FMs compared to *vanilla* NNs of a smaller size. Figure 5 (a) provides an overview of different model sizes. From Figure 5 (b), it is evident that scGPT and Geneformer outperformed *vanilla* NNs in terms of accuracy in cross-data scenarios, suggesting the emergent abilities in the cross-data cell-type annotation task occurred for models which contain parameters larger than 10 million. The low accuracy of scBERT may be caused by its default fine-tuning setting, pre-trained model weights, or model scale.

In the second scenario, the desired emergent ability mirrors that of the first task. Nonetheless, there was no enhancement in performance from Figure 5 (c). Figure 5 (c) shows that *vanilla* NN outperformed scGPT for both the MCA and MB Spatial datasets. *vanilla* NN was comparable to Geneformer for the MCA dataset and worse

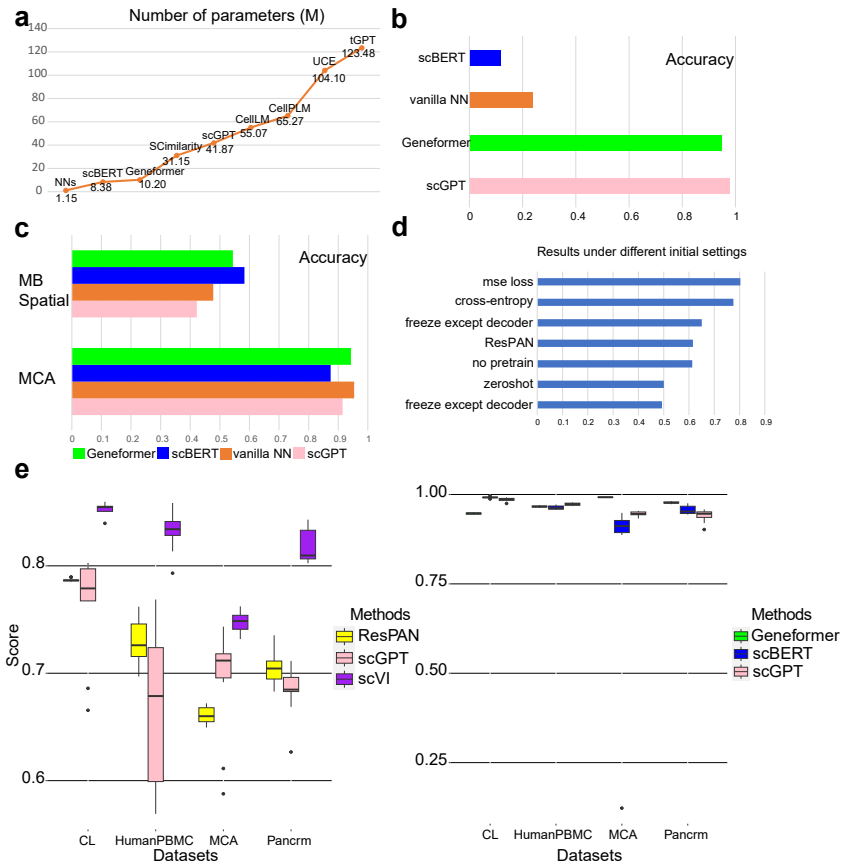


Fig. 5: Different comparison groups for emergent ability analysis and stability analysis. (a): The model scale of different methods, defined by the number of parameters (unit: Million). (b): Accuracy of FMs and *vanilla* NN in Cell-type Annotation task. The dataset here is the Pancreas cross dataset. (c): Accuracy of FMs and *vanilla* NN in Cell-type Annotation task. The datasets here are MB Spatial and MCA. (d): Overall score comparison including ResPAN and different settings of scGPT. The dataset here is the human spatial transcriptomic dataset. (e): Different batch correction scores of different models based on changing random seeds (left) and different average classification scores of different models based on changing random seeds (right). The bold black line represents the median value while the length of each box can be interpreted as the variance level.

than Geneformer for the MB spatial dataset. Moreover, *vanilla* NN was better than scBERT for the MCA dataset and worse than scBERT for the MB spatial dataset.

In the third scenario, we considered two possible emergent abilities. Firstly, in line with the batch effect correction task, we postulated that using Human scRNA-seq data for pre-training could aid in batch effect correction for human spatial transcriptomic data. Secondly, resonating with the cell-type annotation task, we hypothesized

that pre-training on Human scRNA-seq data might assist in cell-type annotation for mouse spatial transcriptomic data. Figure 5 (c) shows that we did not detect any emergent ability in the MB Spatial data annotation task. Figure 5 (d) suggests the emergent ability for batch effect correction. The fine-tuning process appeared beneficial in reducing the batch effect inherent in the spatial data, whereas scenarios with model freezing except decoder yielded subpar results. The performance of scGPT in the integration of spatial data was better than that of ResPAN. Therefore, we observed emergent abilities in the batch effect correction for spatial transcriptomic data.

Stability Analysis. To analyze the stability of single-cell FMs, we selected Batch Effect Correction and Cell-type Annotation as two representative tasks and varied the seeds from 0 to 9 of single-cell FMs to investigate the model stability. All of the methods used in this section are fixed with their default hyper-parameters. These two tasks are the main tasks in single-cell data analysis and have solid metrics for evaluation. Ideally, the results of different single-cell FMs should not vary substantially across different datasets. We also considered stability for other benchmarking tools. Our experiment results summarized in Figure 5 (e) showed that the stability for single-cell FMs is task-specific.

Based on Figure 5 (e), the variance of scVI [73] and ResPAN was generally lower than that of scGPT, and scVI and ResPAN also had a higher score on average. Therefore, single-cell FMs were not as stable as SOTA deep-learning-based methods for the Batch Effect Correction task.

Figure 5 (f) suggests that the variance of Geneformer was generally smaller than scGPT and scBERT. All three models had high median average scores. Moreover, the variance of scBERT was relatively large in the experiments based on the MCA dataset, which implies that single-cell FMs might fail under certain random seeds.

3 Discussion

In this paper, we have evaluated the performance of ten single-cell FMs on eight different tasks for single-cell data analysis. The ranks of these single-cell FMs are shown in Tables 1 (a) and (b), where we not only considered the broader functions of the models, but also their usability. Based on our evaluation results, open-source models have higher ranks than closed-source models. Open-source models with tutorials are more friendly to researchers and these models also receive a number of stars and likes based on Extended Data Figure 27. Moreover, open-source developers can utilize the feedbacks of users to improve their models.

For the Cell-type Annotation task, we found that CellPLM, scGPT, and Geneformer had comparable performance. Moreover, these methods were better than the other FMs, including UCE, scBERT, SCimilarity, and CellLM, as well as task-specific methods, in this task. UCE also had a relatively large model scale compared with the other FMs. This suggests that continually seeking larger models, driven purely by the scaling law, may not always be the preferred approach. For the Gene Function Prediction task, Geneformer also performed well and better than a classical model for prediction. We also showed that the pre-trained model can reach better performance. Moreover, during the fine-tuning process, the optimization of the encoder-decoder

structure did not seem to improve performance. Such findings suggest that we may not need to train the whole model for this task. For the Imputation task, we showed that scGPT did not perform well in the scRNA-seq imputation task, but it outperformed Tangram in the spatial transcriptomic data imputation task both under zero-shot learning and fine-tuning frameworks. Such finding suggests that single-cell FMs can transfer knowledge across different omic data.

We also observed the emergent ability of single-cell FMs. For the cross-data Cell-type Annotation task, the performance of pre-trained scGPT is much better than the performance of neural networks with smaller parameter size. Moreover, scGPT can also be used to analyze spatial transcriptomic data for batch effect correction, which has attracted much attention in recent years [74].

Thus, the contributions of single-cell FMs are substantial, as they allow for the extraction of meaningful biological information across a variety of tasks. These models provide valuable insights and have the potential to advance our understanding of single-cell biology.

Despite the significant contributions of single-cell FMs, there are evident limitations in their construction and training steps. A more comprehensive understanding of such FMs is necessary to address these issues. We employed an ablation testing approach to discern the relative importance of different components in the loss function during the fine-tuning process. The scGPT loss function comprises, at most, five parts, with four fixed components and one task-specific term. However, our experimental results suggest that not all components are of equal importance. The GEPC loss term appears to be the least significant, and the inclusion of the ECS Loss is not consistently beneficial across all the datasets we tested. Therefore, ablation tests should be required in the development process of new single-cell FMs.

For the Batch Effect Correction and Multi-omic Data Integration tasks, single-cell FMs did not outperform task-specific SOTA methods. Moreover, the correction results of atlas-level datasets were worse than the raw datasets. One possible solution is to incorporate extra information, including cell types, in the pre-training or fine-tuning process. Moreover, for the Perturbation Prediction task, the function of model pre-training is not obvious. GEARS was comparable with scGPT in this task. For the Gene Network Analysis task, we did not figure out great contribution of gene embeddings from single-cell FMs for either tissue-specific or cell-type-specific analysis. For the Simulation task, scGPT also did not perform very well. scDesign3 is a better method for simulating reference-based scRNA-seq datasets. One possible reason is from [75], which suggests that current FMs rely on non-transferable procedures for specific tasks, and not all the abilities of FMs are transferable and general.

In our stability analysis, we found that single-cell FMs were not very stable in the Batch Effect Correction task. For some datasets, the cell-type prediction accuracy was also not reliable. Moreover, in the evaluation for the Cell-type Annotation task, we found that different single-cell FMs had different variance levels. The variance under different random seeds was driven by datasets. Considering the problems of stability, we still have difficulty refining it into a toolbox (for example, Seurat [76] or Scanpy [69]) with various functions.

Moreover, we explored the parameter-efficient fine-tuning (PEFT) framework known as LoRA [77] for scGPT in Batch Effect Correction and Cell-type Annotation. Extended Data Figure 28 shows that LoRA slowed down the fine-tuning process. One possible reason is the trade-off between accelerating the forward process and accelerating the backward process since incorporating LoRA decreases the accelerating process [77]. LoRA also could not improve the performance of scGPT.

Thus, the training of single-cell FMs is more nuanced than that of general FMs. While it is important to consider the similarities between these two model types, we must also consider the differences rooted in the datasets and domain-specific knowledge.

We also studied the impacts of the pre-training process. Different models have specific and shared components for their pre-training datasets, and we did not find a clear correlation between the settings of pre-training datasets and model performance. However, by comparing the model from scratch and with the pre-training setting, we found that the pre-training phase plays an important role across multiple tasks, particularly for Cell-type Annotation and Gene Function Prediction. We also observed a considerable performance improvement for these two tasks with pre-training. Moreover, even if the performance of single-cell FMs in batch integration tasks is not outstanding, they can still annotate cell types with high accuracy. Thus, single-cell FMs may be robust to batch effect.

Our results suggest that scGPT's performance is significantly influenced by the learning rate and the number of bins. The former plays a critical role during the training process, while the latter is set during the pre-processing stages. These stages correspond to two essential aspects of task-specific fine-tuning. Furthermore, the pre-training phase can also boost the model's performance on tasks involving cross-species datasets. As an example, the pre-training strategy using human single-cell datasets enabled scGPT to handle the Cell-type Annotation task on mouse datasets effectively.

Interestingly, we found that the optimal optimizer for single-cell FMs was still Adam. When other factors were controlled, the performance derived from the SGD optimizer significantly lagged behind Adam, highlighting the necessity of tolerating the additional memory consumption caused by Adam. Moreover, the contribution of the encoder part is more important than that of the decoder part, which may be explained by the fact that the noise in the single-cell data makes the encoder part of the model take on more tasks. We note that this result is inconsistent with current research in general FMs [78].

Although some tasks were only evaluated based on scGPT, the patterns we found in scGPT could be extended in the training of other single-cell FMs based on Figure 2 (f). We have identified several beneficial factors that can improve the performance of single-cell FMs: 1. Employing a **smaller learning rate** (e.g., 1e-4) and a **larger bin size** (e.g., 501) can improve the final score. A smaller learning rate ensures a more stable training process, while a larger bin size contributes more to reducing batch effects before the training process. 2. During the fine-tuning process, freezing parts of the model can retain performance while saving training time. There is a higher priority for freezing the **decoder** than the encoder. Given that we have numerous tasks requiring the resolution of embeddings, freezing the encoder is not advisable.

3. There is a benefit in integrating **cell-type** into the fine-tuning process in multiple tasks. 4. **Adam** is the most preferred optimizer for single-cell FMs. 5. Possessing a GPU with **larger memory** (e.g., A6000, A100 and H100) is advantageous for the fine-tuning process. 6. Cross-entropy loss is the optimal choice for masked gene expression reconstruction. We provide a table shown in Table 1 to evaluate the number of tasks that different single-cell FMs can handle. Models with the **scGPT** framework achieve the best score based on consideration of both usability and stability.

The application of FMs to single-cell data remains a promising avenue of exploration, given its impressive performance in Cell-type Annotation and Perturbation Prediction tasks. Since much has been done to optimize general FMs [79–82] (including efficient tuning, model compression, and other research directions), we focus specifically on how to better train and apply single-cell FMs. We discuss several potential research topics and future directions below:

In terms of pre-training preparation, we need high-quality data spanning different contexts, such as various cell types, disease states, genders, and even data from different species for pre-training datasets. High-quality pre-training datasets are important for general FMs otherwise the performance may be reduced [83, 84]. The qualification of pre-training data can also be verified with the online learning framework [85]. We can evaluate the performance of single-cell FMs on downstream tasks and then decide which datasets to include. With a better pre-training design, we may increase the scale of current models to a billion level. Moreover, the incorporation of biological information is crucial for the success of a FM in biology. Therefore, we can incorporate other biological factors including GRN and cell-cell interactions (CCI) [86] in the modeling process. With such information, we can also develop domain-specific or tissue-specific FMs for single-cell data analysis. Exploring the contribution of multimodal data to develop a multimodal FM is also a possible track. For example, incorporating text-based biological information [87] or multi-omic data with new tokens may help us further extend the functions of these FMs.

As for model training, both the pre-training and fine-tuning steps of the existing single-cell FMs need improvement. During model pre-training, we should focus on incorporating biological information or human feedback into the process, as opposed to just relying on the conventional masked token prediction task. For instance, integrating cell-type or disease condition information into the pre-training step is an intriguing approach, and allows experts to evaluate the quality of model output during the training process is also meaningful. Moreover, we should also consider the security or trustworthiness of FM training [88]. Using poisoned training single-cell datasets (for example, wrong data or make-up data) as an attack can test the robustness of single-cell FMs. For model fine-tuning, instruction tuning [89] is a potential direction to explore. In this context, cells could be considered as prompts, as described in scGPT. Another possible direction is to focus on generating unified embeddings for cells/genes and combining them with task-specific models for downstream applications, inspired by [33, 90].

For model evaluations, we need more effective methods to assess results for certain tasks. For instance, we may include verification from biology experiments to avoid the harm of incorrect output of single-cell FMs (also known as FM hallucinations

[91]). Moreover, analyzing the emergent abilities of single-cell FMs is also important to explore the breakthrough contribution and significance of a model, though we have not identified a task that can only be addressed by developing single-cell FMs in our evaluation. In addition, we should also account for the pre-training costs (such as training time and power consumption). We should avoid developing a model that consumes large resources but has no significant improvement for downstream tasks. This information can guide users in selecting the most suitable single-cell FM for their needs.

When it comes to task selection, we should first define the tasks rigorously. For example, in the GRN inference tasks defined by scGPT, we cannot treat a co-expression network based on gene embeddings the same as a gene regulatory network. Also from scGPT, using attention we can infer gene-gene correlation strength with direction, but the relation between the correlation of features and the value in the attention map is in debate [92–94]. We should also consider more meaningful, hard-core, and challenging tasks related to single-cell and spatial data for single-cell FMs. These tasks typically need prior information from large-scale transcriptomic data. For instance, tasks such as in-silico treatment analysis [31], complex perturbation analysis in the single-cell levels [95] and spatial data imputation [96] are difficult to handle without prior information, thus provides an opportunity for FMs to make a difference.

In summary, our goal of studying single-cell data-based FMs is to develop a large-scale model that is capable of performing multiple tasks with stable and reliable results. Such a model should also be user-friendly with detailed tutorial and well-maintained website. Our results indicate that there is much space for improvement for existing single-cell FMs. Although our evaluation is subject to the number of current single-cell FMs, the scale of current single-cell FMs, and the strategies of pre-training, we hope that our analysis can provide insights into best practice and guide the development of future FMs for single-cell data analysis.

4 Methods

Problem definition. We consider a pre-trained FM, denoted as $\mathcal{M}(x, \theta)$, which is based on the single-cell dataset $\mathcal{D}$. Here, θ embodies the set of both model parameters (e.g., network weights) and hyper-parameters (e.g., epochs and learning rate). Different FMs have used distinct pre-training datasets. The model structure for the fine-tuning phase is defined as $\mathcal{M}'(x, \theta')$. Our objective is to ascertain the optimal set of θ' for various sub-task. Formally, we denote the loss for task k as $L_k(\cdot, \cdot)$, and use the evaluation dataset $\mathcal{D}_{\text{eval}} = \{x_i, y_i\}_{i=1}^n$, to assess L_k. Our primary goal is to find:

$$\theta^* = \arg\min_{\theta'} \mathbf{E}_{x,y \in \mathcal{D}} \left[L_k(\mathcal{M}'(x, \theta'), y) \right].$$

Our second goal is to evaluate the performance of different single-cell FMs, that is, we intend to find:

$$\mathcal{M}^* = \arg\min_{\mathcal{M}'} \mathbf{E}_{x,y \in \mathcal{D}} \left[L_k(\mathcal{M}'(x, \theta^*), y) \right].$$

Our third goal is to assess other abilities of single-cell FMs, including: 1. zero-shot learning; 2. emergent ability [72]; 3. cross-species data analysis; 4. biological mechanism exploration; and 5. stability.

Parameters and tasks. Most single-cell FMs share the pre-training process. By considering the overlap among various single-cell FMs, we have selected scGPT, scBERT, and Geneformer as representative examples for our analysis. We also highlight the downstream tasks in Figure 1 (a). We focus on eight fine-tuning tasks in total: 1. Batch Effect Correction; 2. Multi-omic Data Integration; 3. Cell-type Annotation; 4. Gene Function Prediction; 5. Perturbation Prediction; 6. Gene Network Analysis; 7. Imputation; and 8. scRNA-seq Simulation. By compiling all the hyper-parameters across different models, we present a list of factors that can affect the model performance categorized by types in Figure 1 (b). The definition of different hyper-parameters is discussed in Appendix A. To analyze the effect of different hyper-parameters, initial settings, and optimizers, we selected representative datasets for different tasks. For Batch Effect Correction and Cell-type Annotation, we selected Pancrm, HumanPBMC, Cell Lines, and MCA because they cover various data types. Pancrm is from Pancreas tissue and has five batches. HumanPBMC is from PBMC and has nine cell types. Cell Lines has two cell types, as a binary label dataset. MCA is from *Mus musculus*. For Multi-omic Data Integration, we included all of the four datasets we used to analyze because their scales cover a large range. For Gene Function Prediction, we included all of the three datasets we used to analyze because they correspond to different sub-tasks. For Perturbation Prediction, we included all of the three datasets we used to analyze because their scales cover a large range and from different sources.

Explanations of scEval. Here we introduce a framework known as scEval to evaluate the performance of different single-cell FMs on various tasks. The whole pipeline is highlighted in Figure 1 (a). Since most of the single-cell FMs have an encoder-decoder structure as well as extra fine-tuning blocks, we split the eight tasks into two types. For tasks included in sub-task 1, we need an extra fine-tuning block to generate the results. Cell-type Annotation and Gene Function Prediction belong to this task type. For tasks included in sub-task 2, we rely on the outputs of the encoder part and decoder part to generate the results. The rest of the six tasks belong to this task type. The idea of scEval is inspired by benchmarking analysis in both single-cell area [56, 96, 97] and LLM area [98, 99]. According to Figure 1 (b), for each task, we collect the output of single-cell FMs under different settings. Each FM has its specific running settings, and their default modes can be classified into fine-tuning-based models and zero-shot-learning-based models, thus we propose a framework focusing on the outputs of different single-cell FMs to unify the evaluation process. To analyze the factors affecting single-cell FMs, we also adjusted the factors of different FMs and collected their results for evaluation. Moreover, scEval contains different evaluation pipelines for different tasks, and it can also be easily extended to evaluate more tasks by adding more functions.

Batch Effect Correction. Batch Effect Correction is an essential step following scRNA-seq data pre-processing. It primarily signifies the distribution disparity in

scRNA-seq datasets originating from the same tissue, which can be attributed to various factors [100]. The reduction of batch effects is critical not only to allow researchers to discern genuine biological signals but also to facilitate integrated analyses across different studies. The challenge of this task arises from the need to balance the removal of batch signals with the preservation of biological signals. We treat this task as a data integration problem.

For the Batch Effect Correction, the metrics we considered here were inspired by scIB [56], including Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Cell-type Average Silhouette Width (cell-type ASW) for the biological conservation score; and batch Average Silhouette Width (batch ASW), Principal Component Regression (PCR), Graph Connectivity (GC) and kBET [101] for the batch effect correction score. We compute the weighted average of these metrics to represent the final batch effect correction score. Details of these metrics can be found in Appendix D.1. Let S_{bio} donate the average biological conservation metric and S_{batch} donate the average batch effect correction metric as S_{batch}, the final model score is:

$$S_{\text{final}} = 0.6 \cdot S_{\text{bio}} + 0.4 \cdot S_{\text{batch}}.$$

Multi-omic Data Integration. Multi-omic Data Integration is a key step for multi-omic data analysis [22]. It is akin to an advanced form of batch effect correction. If unpaired multi-omic data are present, the objective is to map different datasets into a shared space for subsequent analysis. If paired multi-omic data are present, the goal is to assess whether the use of multi-omic data can contribute to learning a more comprehensive representation of the data. A significant challenge here is how to align omics at the feature level. For instance, the feature of the scRNA-seq data is a gene, the feature of the scATAC-seq data is a peak, and the feature of the protein data is a protein. The tokenization step can become complex given different modalities. We treat this task as a data integration problem. We used the same metrics for multi-omic data integration as those for Batch Effect Correction.

Cell-type Annotation. Cell-type Annotation is another key step following single-cell data pre-processing [102]. This step annotates each cell with its accurate cell-type label, which can be achieved through prior knowledge [103] or computational methods [104]. These annotated cell-type labels can provide essential biological information for further downstream analysis, such as cell-type specific network analysis. In addition, drug response prediction [33] or single-cell disease classification [105] can also be treated as a variation of this task. A common approach employed by single-cell FMs in dealing with the cell-type annotation task is to use single-cell datasets for model training and treat the unannotated datasets as testing datasets. The challenge lies in predicting or annotating a set of cells that originate from studies different from the training datasets. Differently, SCimilarity pre-trained the model with cell-type labels to allow query of cell types directly for testing datasets. Moreover, the existence of cells with novel cell types (which are not included in the training datasets) further complicates the problem. We treat this task as a multi-label classification problem.

In this task, we chose datasets with batch effect in two different cases. The intra-dataset case allows batch intersection, which means that the training and testing datasets can contain cells from the same batch. Here the total dataset was split into

approximately 70% as a training dataset and the rest as a testing dataset. The inter-dataset case is cross-batch (cross-data) annotation, which means that the training and testing datasets are from different sources. We consider two datasets from the same tissue in this setting. More specifically, we consider the Pancreas cross dataset from the Pancreas, and the Liver cross dataset from the Liver. The main score for evaluation here is accuracy, which is defined as:

$$S_{\text{celltype}} = \frac{\#\text{corrected classified cells}}{\#\text{Total cells}}.$$

We also consider Precision, Recall and F1 scores in the analysis for ablation test, and details can be found in Appendix D.2. Moreover, except the general comparison, we considered four datasets for the effect of different hyper-parameters, initial settings and optimizers: Pancrm, HumanPBMC, Cell Lines, and MCA, which is from *Mus musculus* rather than *Homo sapiens*. Finally, we investigated the contribution of freezing layers for cell-type annotation by freezing different numbers (from 0 to 11) of forward layers of scGPT.

Gene Function Prediction. Gene Function Prediction is important to identify the properties of genes across different conditions [31]. There are approximately 20,000 protein-encoding genes for humans [106] and only some are annotated with functions. Accurate prediction of gene function can help us understand and infer the role of genes in biological systems. Here we consider three types of functions for genes. The first one is dosage-sensitive or non-sensitive. Some genes are dosage-sensitive, which means that they are significant in the analysis of Copy Number Variants (CNVs) related to genetic diagnosis. The second one is Bivalent versus non-methylated. Bivalent chromatin structure is important to identify key developmental genes in embryonic stem cells (ESCs). Therefore, identifying bivalently marked genes versus unmethylated genes is important. The third one is Bivalent versus Lys4-only methylated. Lys4-only-methylated genes are also different from bivalently marked genes. We compare the model output with the true gene labels. We treat this task as a binary classification problem. Here we used the same metrics as the cell-type annotation task. We used a public dataset [31] considering only labeled genes in the dataset for prediction and evaluation.

Perturbation Prediction. Perturbation Prediction [52] is a task based on gene editing and single-cell sequencing technologies. After silencing some genes, we can obtain unperturbed and perturbed gene expression levels by sequencing, which allows us to explore the interactions between genes. A well-known technique is Perturb-seq [107]. In perturbation prediction, we intend to predict the gene expression level after gene editing. Here a model may predict seen gene perturbation in the testing datasets (an easier one) or predict unseen gene perturbation in the testing datasets (a more difficult one). We treat this task as a regression problem. The metric we used here is MPC, and the details can be found in Appendix D.4. In the perturbation prediction task, we construct the paired input-target datasets by selecting the cells with non-control guide identity and then randomly sample cells under the control condition, and then combine them as the training and testing datasets. Our Perturb-seq datasets are from GEARS [52], which contain cells with three conditions: control; one gene perturbation; and two genes perturbation. In the evaluation process, we combined case

2 and case 3. We also compared the MPC across all genes to avoid bias in the gene selection process.

Gene Network Analysis. Gene Network Analysis is a downstream task for single-cell datasets [108]. The objective is to infer specific gene networks (for example, Gene Regulatory Network (GRN) or Gene Co-expression Network (GCN)) from different datasets. A GRN can assist in understanding the regularory relationships between genes and predicted perturbation outcomes. The challenge in this task stems from the Granger causal relationship or time-dependent correlation [109]. A GCN can be used to analyze genes with similar functions or uncover the characteristics of genes in some diseases [110]. GCN and GRN are two different tasks because correlation does not imply causal relation [111]. This limitation means that we cannot determine which genes are the "causes" of expression level changes in other genes only based on embeddings similarity or correlation. We treat this task as a network inference problem. In the gene network analysis task, we considered using the overlap between ground truth genes in certain pathways and inferred genes as one metric, and using the ratio of significant pathways related to inferred genes as the other metric. Details can be found in Appendix D.4.

Imputation. Imputation is a filling task related to missing data. Generally, we have two targets: 1. Perform imputation for scRNA-seq data to reduce data noise and fill in technical zeros with biologically meaningful values [112, 113]. 2. Perform imputation for spatial transcriptomic data because of unseen or unmeasured genes [65, 66]. According to [114], current spatial imputation methods do not show strong performance across different datasets. Using single-cell FMs, we can either use zero-shot learning to impute the unseen genes, or fine-tune our model based on reference scRNA-seq with more genes to perform imputation. We treat this task as a matrix-completion problem. Details of the metrics we used here can be found in Appendix D.5. In the imputation task, we used two public datasets from the mouse tissue to analyze the performance of single-cell FMs. One dataset is a scRNA-seq dataset, and another one is a spatial transcriptomic dataset. For the imputation of the scRNA-seq dataset, we used the output of the model decoder as imputation results. To evaluate this task, we used the biology preservation score from the batch effect correction and compared it to the score from raw data. For the imputation of spatial transcriptomic data, we considered two different settings to perform imputation. The first setting uses scRNA-seq to perform fine-tuning and inference based on spatial transcriptomic data. The second setting uses a zero-shot learning framework to directly perform inference based on spatial transcriptomic data. We consider using the correlation between known raw gene expression and known imputed gene expression as a metric.

Simulation. scRNA-seq Simulation is a data generation task. Leveraging the generative pre-training process of scGPT, we can generate gene expressions based on real datasets. Since a prevalent issue with scRNA-seq data simulation is the considerable divergence between simulation datasets and real datasets [70], direct generation from real datasets is preferred.

By arranging different sequences of masking genes or altering different seeds, we can generate new simulated scRNA-seq datasets from real ones. By modifying scGPT,

we fine-tuned the baseline model based on the reference scRNA-seq dataset with reconstruction loss and utilized the outputs of the decoder part to generate the simulation datasets. Such simulation datasets do not have exactly the same gene expression profiles compared with the reference scRNA-seq dataset, but they should preserve the biological variation information from the reference dataset. We utilized scGPT to simulate datasets with batch effect or without batch effect. To simulate datasets with batch effect, we removed the loss function for removing batch effect in the original pipeline. To simulate datasets without batch effect, we kept the original pipeline for the batch effect correction task but utilized the outputs of the decoder rather than the encoder. We did not incorporate extra information for implementing this function. The quality of our simulation datasets can be evaluated by comparing them with the outputs of current simulation methods. We treat this task as a data-generation problem.

We used the same metrics as Batch Effect Correction for evaluation. It is possible to produce diverse reconstruction outcomes from a single real dataset by varying the random seeds. This feature enables us to create simulated single-cell datasets. Notably, these generated datasets retain the same gene sets as their input counterparts. We generate the simulation results by multiplying the output of the gene expression decoder and Bernoulli decoder. We have the flexibility to generate datasets either with or without batch effects. If we intend to produce datasets with batch effects, the gradient reverse loss function is omitted. Conversely, to generate datasets without batch effects, we either retain this function or utilize single-batch data.

Emergent Ability. [72, 115] argue that due to the extensive number of parameters, FMs can manage specific tasks that small-scale models cannot handle, based on few-shots learning. Such an attribute refers to Emergent Ability. For instance, GPT-4 can tackle some SAT questions even though it has not been trained on such datasets with novel scaling [17]. We hypothesize that single-cell FMs may also possess this capability. To test this, we devise different scenarios similar to instances of emergent ability to assess the performance of single-cell FMs. Such scenarios include: Cross-dataset Cell-type Annotation, Cross-species Cell-type Annotation, and Spatial transcriptomic data analysis.

Ablation tests. Given that there is no existing work investigating the significance of different loss function components of single-cell FMs, we conducted a comprehensive analysis of the impact of various loss function components. These include the Masked Gene Expression loss (Mask Loss), Zero Log Probability Loss (Prob Loss), Gene Expression Prediction for Cell Modelling Loss (GEPC Loss), and Elastic Cell Similarity Loss (ECS Loss). We retain the task-specific loss in the fine-tuning process as the baseline condition (All remove). Consequently, our null hypothesis (H_0) is that the removal of the component C will not degrade the performance, while the alternative hypothesis (H_1) is that the removal of component C will worsen the model's performance. Comparing the score after eliminating a specific component to the score from the default setting allows us to determine whether to reject the null hypothesis. The test we employed here is the paired Students' t-test. The details of the different loss function components are shown below.

1. Mask Loss: In both pre-training and fine-tuning, we mask the expression levels of some genes, denoted as $\mathcal{M}_{\text{mask}}$, for gene expression prediction. The mask loss is

motivated from this setting and works as follows,

$$\tilde{\boldsymbol{x}}^{(i)} = \mathrm{MLP}\left(\boldsymbol{h}_n^{(i)}\right),$$
$$\mathcal{L}_{\mathrm{mask}} = \frac{1}{|\mathcal{M}_{\mathrm{mask}}|} \sum_{j \in \mathcal{M}_{\mathrm{mask}}} \mathrm{ce}\left(\tilde{x}_j^{(i)}, x_j^{(i)}\right),$$

where $\tilde{\boldsymbol{x}}^{(i)}$ represents the predicted gene expression levels for cell i, while $\tilde{\boldsymbol{x}}_j^{(i)}$ represents the ground truth. $\boldsymbol{h}_n^{(i)}$ represents the embeddings for cell i with n genes. MLP means that we use linear Multi-Layer Perceptron (MLP) Neural Networks as the output of single-cell FMs under the setting of this loss function.

2. Prob Loss: Since single-cell data can be treated as count data, we can use the Bernoulli distribution to model the occurrence of the expression in masked genes and use the Maximum Likelihood Estimation (MLE) approach to estimate the parameters of Bernoulli distribution. Such loss function can be used to determine whether the given masked gene position carries zero expression levels or not. Prob Loss works as follows:

$$\mathrm{Prob}_i = \mathrm{MLP}\left(\boldsymbol{h}_n^{(i)}\right),$$
$$\mathrm{Dist}_i = \textit{Bernoulli}(\mathrm{Prob}_i),$$
$$\mathcal{L}_{\mathrm{Prob}} = -\mathrm{LogProb}_i(x^{(i)} > 0),$$

where Prob_i represents the output of the model for the parameter estimation for cell i. The estimation is based on one MLP model. Dist_i represents the Bernoulli distribution based on Prob_i. $\mathrm{LogProb}_i$ is the log probability based on the relationship between gene expression levels of cell i and zero, which can be computed based on Dist_i.

3. GEPC Loss: This loss function is similar to Mask Loss, but now we predict the gene expression levels based on cell embeddings or cell representation. For cell i with gene j, we create a query vector $\boldsymbol{q}_j$ and represent the cell based on $\boldsymbol{h}^{(i)}c$. We can use the inner product between these two terms to predict gene expression levels. That is,

$$\boldsymbol{q}_j = \mathrm{MLP}\left(\mathrm{emb_g}\left(\boldsymbol{t}_g^{(i)}\right)\right),$$
$$\tilde{x}_j^{(i)} = \boldsymbol{q}_j \cdot \boldsymbol{W}\boldsymbol{h}_c^{(i)},$$
$$\mathcal{L}_{\mathrm{GEPC}} = \frac{1}{|\mathcal{M}_{\mathrm{mask}}|} \sum_{j \in \mathcal{M}_{\mathrm{mask}}} \mathrm{ce}\left(\tilde{x}_j^{(i)}, x_j^{(i)}\right),$$

where $\mathrm{emb_g}(\boldsymbol{t}_g^{(i)})$ represents the embeddings of the gene token g in cell i. We also use one MLP to generate the query embeddings. The following process is similar to the steps for $\mathcal{L}_{\mathrm{mask}}$.

4. ECS Loss: This loss function is used to control the similarity of the embeddings of cells in the same batch, which is defined as:

$$\mathcal{L}_{\mathrm{ECS}} = -\left(\mathrm{CosSim}\left(\boldsymbol{h}_c^{(i)}, \boldsymbol{h}_c^{(i')}\right) - \beta\right)^2,$$

where CosSim represents the cosine similarity function, i and i' are the indices of the two cells. The idea of this loss function is to ensure the similarity between paired cells is higher than the predefined threshold β. Moreover, dissimilar pairs should be more dissimilar, respectively.

Task-specific fine-tuning process. For the experiments we have in the Results section, we load the pre-training weights based on the requirement of different single-cell FMs. The pre-training weights we used can be found in our GitHub folder. After the fine-tuning process, we recorded the related metrics and conducted more analysis. In the experiments for all tasks, we chose scGPT as a baseline and representative model for the following three reasons. Firstly, scGPT is an open-source single-cell FM with the largest datasets for pre-training, and it is well-defined with detailed tutorials. In addition, the architecture of scGPT is easy to adjust and includes multiple loss function components. The functions of Gene Function Prediction, Imputation, and Simulation for scGPT were designed in scEval for evaluation. The settings of hyper-parameters for these tasks are transferred from the design for cell-type annotation and batch effect correction tasks. Moreover, different single-cell FMs have overlaps and unique terms in the pre-training and fine-tuning framework, but scGPT is the most general one. In each task, we also included task-specific methods as comparisons. Moreover, for Geneformer, scBERT, CellPLM and CellLM, we also evaluated their performance based on shared hyper-parameters with scGPT to verify if our rules found in scGPT can be extended for other single-cell FMs.

5 Data availability and reproducibility

We used the resources from the Yale High Performance Center (Yale HPC) to conduct all of the experiments. Our maximum running time for each dataset is 24 hours. The version of GPU we used is NVIDIA A100 (40 GB). The random seed of all experiments is the same as the default setting of the original papers.

The information of datasets and the download link can be found in Appendix F. The codes of **scEval** can be found in https://github.com/HelloWorldLTY/scEval with MIT license.

6 Acknowledgements

We appreciate the comments, feedback, and model explanations from Yingxin Lin, Haotian Cui, Christina Theodoris, Rex Ying, Minsheng Hao and Malte D. Luecken. We also appreciate the constructive comments from reviewers and editors of this manuscript. This project is supported in part by NIH grants R01 GM134005 and P50CA196530.

7 Author contributions

T.L. designed the study with H.Z. and Y.W. T.L. ran the experiments with K.L. H.L. and T.L. designed the website. T.L., K.L. and H.Z. wrote the manuscript. H.Z. supervised this work.

References

[1] Han, X., Zhou, Z., Fei, L., Sun, H., Wang, R., Chen, Y., Chen, H., Wang, J., Tang, H., Ge, W., *et al.*: Construction of a human cell landscape at single-cell level. Nature **581**(7808), 303–309 (2020)

[2] Saliba, A.-E., Westermann, A.J., Gorski, S.A., Vogel, J.: Single-cell rna-seq: advances and future challenges. Nucleic acids research **42**(14), 8845–8860 (2014)

[3] Stubbington, M.J., Rozenblatt-Rosen, O., Regev, A., Teichmann, S.A.: Single-cell transcriptomics to explore the immune system in health and disease. Science **358**(6359), 58–63 (2017)

[4] Mathys, H., Davila-Velderrain, J., Peng, Z., Gao, F., Mohammadi, S., Young, J.Z., Menon, M., He, L., Abdurrob, F., Jiang, X., *et al.*: Single-cell transcriptomic analysis of alzheimer's disease. Nature **570**(7761), 332–337 (2019)

[5] Zhang, L., He, C.H., Coffey, S., Yin, D., Hsu, I.-U., Su, C., Ye, Y., Zhang, C., Spurrier, J., Nicholson, L., et al.: Single-cell transcriptomic atlas of alzheimer's disease middle temporal gyrus reveals region, cell type and sex specificity of gene expression with novel genetic risk for mertk in female. medRxiv, 2023–02 (2023)

[6] Evrony, G.D., Hinch, A.G., Luo, C.: Applications of single-cell dna sequencing. Annual review of genomics and human genetics **22**, 171–197 (2021)

[7] Hwang, B., Lee, J.H., Bang, D.: Single-cell rna sequencing technologies and bioinformatics pipelines. Experimental & molecular medicine **50**(8), 1–14 (2018)

[8] Zheng, G.X., Terry, J.M., Belgrader, P., Ryvkin, P., Bent, Z.W., Wilson, R., Ziraldo, S.B., Wheeler, T.D., McDermott, G.P., Zhu, J., *et al.*: Massively parallel digital transcriptional profiling of single cells. Nature communications **8**(1), 1–12 (2017)

[9] Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P.K., Swerdlow, H., Satija, R., Smibert, P.: Simultaneous epitope and transcriptome measurement in single cells. Nature methods **14**(9), 865–868 (2017)

[10] Cusanovich, D.A., Daza, R., Adey, A., Pliner, H.A., Christiansen, L., Gunderson, K.L., Steemers, F.J., Trapnell, C., Shendure, J.: Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing. Science **348**(6237), 910–914 (2015)

[11] Chen, X., Miragaia, R.J., Natarajan, K.N., Teichmann, S.A.: A rapid and robust method for single cell chromatin accessibility profiling. Nature communications **9**(1), 1–9 (2018)

[12] Luo, C., Keown, C.L., Kurihara, L., Zhou, J., He, Y., Li, J., Castanon, R., Lucero, J., Nery, J.R., Sandoval, J.P., *et al.*: Single-cell methylomes identify neuronal subtypes and regulatory elements in mammalian cortex. Science **357**(6351), 600–604 (2017)

[13] Mulqueen, R.M., Pokholok, D., Norberg, S.J., Torkenczy, K.A., Fields, A.J., Sun, D., Sinnamon, J.R., Shendure, J., Trapnell, C., O'Roak, B.J., *et al.*: Highly scalable generation of dna methylation profiles in single cells. Nature biotechnology **36**(5), 428–431 (2018)

[14] Teichmann, S., Efremova, M.: Method of the year 2019: single-cell multimodal omics. Nat. Methods **17**(1), 2020 (2020)

[15] Flynn, E., Almonte-Loya, A., Fragiadakis, G.K.: Single-cell multiomics. Annual Review of Biomedical Data Science **6** (2023)

[16] Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)

[17] OpenAI: GPT-4 Technical Report (2023)

[18] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: LLaMA: Open and Efficient Foundation Language Models (2023)

[19] Thirunavukarasu, A.J., Ting, D.S.J., Elangovan, K., Gutierrez, L., Tan, T.F., Ting, D.S.W.: Large language models in medicine. Nature medicine **29**(8), 1930–1940 (2023)

[20] Imani, S., Du, L., Shrivastava, H.: MathPrompter: Mathematical reasoning using large language models. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track), pp. 37–42. Association for Computational Linguistics, Toronto, Canada (2023). https://doi.org/10.18653/v1/2023.acl-industry.4 . https://aclanthology.org/2023.acl-industry.4

[21] Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023)

[22] Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W.M., Hao, Y., Stoeckius, M., Smibert, P., Satija, R.: Comprehensive integration of single-cell data. Cell **177**(7), 1888–1902 (2019)

[23] Kuzman, T., Ljubešić, N., Mozetič, I.: Chatgpt: beginning of an end of manual annotation? use case of automatic genre identification. arXiv preprint arXiv:2303.03953 (2023)

[24] Chen, J., Xu, H., Tao, W., Chen, Z., Zhao, Y., Han, J.-D.J.: Transformer for one stop interpretable cell type annotation. Nature Communications **14**(1), 223 (2023)

[25] Heumos, L., Schaar, A.C., Lance, C., Litinetskaya, A., Drost, F., Zappia, L., Lücken, M.D., Strobl, D.C., Henao, J., Curion, F., et al.: Best practices for single-cell analysis across modalities. Nature Reviews Genetics, 1–23 (2023)

[26] Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J.R., Grabska-Barwinska, A., Taylor, K.R., Assael, Y., Jumper, J., Kohli, P., Kelley, D.R.: Effective gene expression prediction from sequence by integrating long-range interactions. Nature methods **18**(10), 1196–1203 (2021)

[27] Xu, H., Woicik, A., Poon, H., Altman, R.B., Wang, S.: Multilingual translation for zero-shot biomedical classification using biotranslator. Nature Communications **14**(1), 738 (2023)

[28] Singhal, K., Azizi, S., Tu, T., Mahdavi, S.S., Wei, J., Chung, H.W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al.: Large language models encode clinical knowledge. Nature, 1–9 (2023)

[29] Yang, F., Wang, W., Wang, F., Fang, Y., Tang, D., Huang, J., Lu, H., Yao, J.: scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. Nature Machine Intelligence **4**(10), 852–866 (2022)

[30] Zhao, S., Zhang, J., Nie, Z.: Large-scale cell representation learning via divide-and-conquer contrastive learning. arXiv preprint arXiv:2306.04371 (2023)

[31] Theodoris, C.V., Xiao, L., Chopra, A., Chaffin, M.D., Al Sayed, Z.R., Hill, M.C., Mantineo, H., Brydon, E.M., Zeng, Z., Liu, X.S., et al.: Transfer learning enables predictions in network biology. Nature, 1–9 (2023)

[32] Cui, H., Wang, C., Maan, H., Wang, B.: scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. bioRxiv, 2023–04 (2023)

[33] Hao, M., Gong, J., Zeng, X., Liu, C., Guo, Y., Cheng, X., Wang, T., Ma, J., Song, L., Zhang, X.: Large scale foundation model on single-cell transcriptomics. bioRxiv, 2023–05 (2023)

[34] Shen, H., Liu, J., Hu, J., Shen, X., Zhang, C., Wu, D., Feng, M., Yang, M., Li, Y., Yang, Y., et al.: Generative pretraining from large-scale transcriptomes for single-cell deciphering. Iscience **26**(5) (2023)

[35] Yang, X., Liu, G., Feng, G., Bu, D., Wang, P., Jiang, J., Chen, S., Yang, Q., Zhang, Y., Man, Z., et al.: Genecompass: Deciphering universal gene regulatory mechanisms with knowledge-informed cross-species foundation model. bioRxiv, 2023–09 (2023)

[36] Heimberg, G., Kuo, T.C., DePianto, D., Heigl, T., Diamant, N., Salem, O., Scalia, G., Biancalani, T., Rock, J., Turley, S., et al.: Scalable querying of human cell atlases via a foundational model reveals commonalities across fibrosis-associated macrophages. bioRxiv (2023)

[37] Rosen, Y., Roohani, Y., Agrawal, A., Samotorcan, L., Consortium, T.S., Quake, S.R., Leskovec, J.: Universal cell embeddings: A foundation model for cell biology. bioRxiv, 2023–11 (2023)

[38] Wen, H., Tang, W., Dai, X., Ding, J., Jin, W., Xie, Y., Tang, J.: Cellplm: Pre-training of cell language model beyond single cells. bioRxiv, 2023–10 (2023)

[39] Jackson, H.W., Fischer, J.R., Zanotelli, V.R., Ali, H.R., Mechera, R., Soysal, S.D., Moch, H., Muenst, S., Varga, Z., Weber, W.P., *et al.*: The single-cell pathology landscape of breast cancer. Nature **578**(7796), 615–620 (2020)

[40] Franzén, O., Gan, L.-M., Björkegren, J.L.: Panglaodb: a web server for exploration of mouse and human single-cell rna sequencing data. Database **2019**, 046 (2019)

[41] Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.-r., Raychaudhuri, S.: Fast, sensitive and accurate integration of single-cell data with harmony. Nature methods **16**(12), 1289–1296 (2019)

[42] Wang, Y., Liu, T., Zhao, H.: Respan: a powerful batch correction model for scrna-seq data through residual adversarial networks. Bioinformatics **38**(16), 3942–3949 (2022)

[43] McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)

[44] Xue, F., Fu, Y., Zhou, W., Zheng, Z., You, Y.: To repeat or not to repeat: Insights from scaling llm under token-crisis. arXiv preprint arXiv:2305.13230 (2023)

[45] Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR), San Diega, CA, USA (2015)

[46] Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)

[47] Liu, H., Li, Z., Hall, D., Liang, P., Ma, T.: Sophia: A scalable stochastic second-order optimizer for language model pre-training. arXiv preprint arXiv:2305.14342 (2023)

[48] Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Liu, Y., Pham, H., Dong, X., Luong, T., Hsieh, C.-J., Lu, Y., Le, Q.V.: Symbolic Discovery of Optimization Algorithms. arXiv (2023). https://arxiv.org/abs/2302.06675

[49] Lin, Y., Wu, T.-Y., Wan, S., Yang, J.Y., Wong, W.H., Wang, Y.R.: scjoint integrates atlas-scale single-cell rna-seq and atac-seq data with transfer learning. Nature biotechnology **40**(5), 703–710 (2022)

[50] Cao, Z.-J., Gao, G.: Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. Nature Biotechnology **40**(10), 1458–1466 (2022)

[51] Du, J., Jia, P., Dai, Y., Tao, C., Zhao, Z., Zhi, D.: Gene2vec: distributed representation of genes based on co-expression. BMC genomics **20**, 7–15 (2019)

[52] Roohani, Y., Huang, K., Leskovec, J.: Predicting transcriptional outcomes of novel multigene perturbations with gears. Nature Biotechnology, 1–9 (2023)

[53] Thomas, P.D., Ebert, D., Muruganujan, A., Mushayahama, T., Albou, L.-P., Mi, H.: Panther: Making genome-scale phylogenetics accessible to all. Protein Science **31**(1), 8–22 (2022)

[54] Badia-i-Mompel, P., Wessels, L., Müller-Dott, S., Trimbour, R., Ramirez Flores, R.O., Argelaguet, R., Saez-Rodriguez, J.: Gene regulatory network inference in the era of single-cell multi-omics. Nature Reviews Genetics, 1–16 (2023)

[55] Sebastian, S., Roy, S., Kalita, J.: A generic parallel framework for inferring large-scale gene regulatory networks from expression profiles: application to alzheimer's disease network. Briefings in Bioinformatics **24**(1), 482 (2023)

[56] Luecken, M.D., Büttner, M., Chaichoompu, K., Danese, A., Interlandi, M., Müller, M.F., Strobl, D.C., Zappia, L., Dugas, M., Colomé-Tatché, M., *et al.*: Benchmarking atlas-level data integration in single-cell genomics. Nature methods **19**(1), 41–50 (2022)

[57] Fabregat, A., Jupe, S., Matthews, L., Sidiropoulos, K., Gillespie, M., Garapati, P., Haw, R., Jassal, B., Korninger, F., May, B., *et al.*: The reactome pathway knowledgebase. Nucleic acids research **46**(D1), 649–655 (2018)

[58] Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Scientific reports **9**(1), 5233 (2019)

[59] Li, X.-h., Wang, Z.-x., Lu, T.-y., Che, X.-j.: Modelling immune system: Principles, models, analysis and perspectives. Journal of Bionic Engineering **6**(1), 77–85 (2009)

[60] Davis, M.M., Tato, C.M., Furman, D.: Systems immunology: just getting started. Nature immunology **18**(7), 725–732 (2017)

[61] Kang, J.B., Shen, A.Z., Sakaue, S., Luo, Y., Gurajala, S., Nathan, A., Rumker, L., Aguiar, V.R., Valencia, C., Lagattuta, K., et al.: Mapping the dynamic genetic regulatory architecture of hla genes at single-cell resolution. medRxiv, 2023–03 (2023)

[62] Meyer, D., C. Aguiar, V.R., Bitarello, B.D., C. Brandt, D.Y., Nunes, K.: A genomic perspective on hla evolution. Immunogenetics **70**, 5–27 (2018)

[63] Brandt, D.Y., Aguiar, V.R., Bitarello, B.D., Nunes, K., Goudet, J., Meyer, D.: Mapping bias overestimates reference allele frequencies at the hla genes in the 1000 genomes project phase i data. G3: Genes, Genomes, Genetics **5**(5), 931–941 (2015)

[64] Sakaue, S., Gurajala, S., Curtis, M., Luo, Y., Choi, W., Ishigaki, K., Kang, J.B., Rumker, L., Deutsch, A.J., Schönherr, S., et al.: Tutorial: a statistical genetics guide to identifying hla alleles driving complex disease. Nature Protocols, 1–17 (2023)

[65] Biancalani, T., Scalia, G., Buffoni, L., Avasthi, R., Lu, Z., Sanger, A., Tokcan, N., Vanderburg, C.R., Segerstolpe, Å., Zhang, M., *et al.*: Deep learning and alignment of spatially resolved single-cell transcriptomes with tangram. Nature methods **18**(11), 1352–1362 (2021)

[66] Li, H., Zhou, J., Li, Z., Chen, S., Liao, X., Zhang, B., Zhang, R., Wang, Y., Sun, S., Gao, X.: A comprehensive benchmarking with practical guidelines for cellular deconvolution of spatial transcriptomics. Nature Communications **14**(1), 1548 (2023)

[67] Lun, A.T., McCarthy, D.J., Marioni, J.C.: A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor. F1000Research **5** (2016)

[68] Satija, R., Farrell, J.A., Gennert, D., Schier, A.F., Regev, A.: Spatial reconstruction of single-cell gene expression data. Nature biotechnology **33**(5), 495–502 (2015)

[69] Wolf, F.A., Angerer, P., Theis, F.J.: Scanpy: large-scale single-cell gene expression data analysis. Genome biology **19**, 1–5 (2018)

[70] Zappia, L., Phipson, B., Oshlack, A.: Splatter: simulation of single-cell rna sequencing data. Genome biology **18**(1), 174 (2017)

[71] 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, 1–6 (2023)

[72] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E.H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., Fedus, W.: Emergent abilities of large language models. Transactions on Machine Learning Research (2022). Survey Certification

[73] 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 (2018)

[74] Marx, V.: Method of the year: spatially resolved transcriptomics. Nature methods **18**(1), 9–14 (2021)

[75] Wu, Z., Qiu, L., Ross, A., Akyürek, E., Chen, B., Wang, B., Kim, N., Andreas, J., Kim, Y.: Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. arXiv preprint arXiv:2307.02477 (2023)

[76] Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W.M., Zheng, S., Butler, A., Lee, M.J., Wilk, A.J., Darby, C., Zager, M., *et al.*: Integrated analysis of multimodal single-cell data. Cell **184**(13), 3573–3587 (2021)

[77] Hu, E.J., shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=nZeVKeeFYf9

[78] Wang, T., Roberts, A., Hesslow, D., Le Scao, T., Chung, H.W., Beltagy, I., Launay, J., Raffel, C.: What language model architecture and pretraining objective works best for zero-shot generalization? In: International Conference on Machine Learning, pp. 22964–22984 (2022). PMLR

[79] Hu, Z., Lan, Y., Wang, L., Xu, W., Lim, E.-P., Lee, R.K.-W., Bing, L., Poria, S.: Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. arXiv preprint arXiv:2304.01933 (2023)

[80] Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., *et al.*: Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence **5**(3), 220–235 (2023)

[81] Wang, Y., Si, S., Li, D., Lukasik, M., Yu, F., Hsieh, C.-J., Dhillon, I.S., Kumar, S.: Preserving in-context learning ability in large language model fine-tuning. arXiv preprint arXiv:2211.00635 (2022)

[82] Xu, R., Luo, F., Zhang, Z., Tan, C., Chang, B., Huang, S., Huang, F.: Raise a child in large language model: Towards effective and generalizable fine-tuning. arXiv preprint arXiv:2109.05687 (2021)

[83] Chen, L., Zaharia, M., Zou, J.: How is chatgpt's behavior changing over time? arXiv preprint arXiv:2307.09009 (2023)

[84] Ji, Y., Deng, Y., Gong, Y., Peng, Y., Niu, Q., Zhang, L., Ma, B., Li, X.: Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases. arXiv preprint arXiv:2303.14742 (2023)

[85] Carta, T., Romac, C., Wolf, T., Lamprier, S., Sigaud, O., Oudeyer, P.-Y.: Grounding large language models in interactive environments with online reinforcement learning. arXiv preprint arXiv:2302.02662 (2023)

[86] Armingol, E., Officer, A., Harismendy, O., Lewis, N.E.: Deciphering cell–cell interactions and communication from gene expression. Nature Reviews Genetics **22**(2), 71–88 (2021)

[87] Chen, Y.T., Zou, J.: Genept: A simple but hard-to-beat foundation model for genes and cells built from chatgpt. bioRxiv, 2023–10 (2023)

[88] Liang, W., Tadesse, G.A., Ho, D., Fei-Fei, L., Zaharia, M., Zhang, C., Zou, J.: Advances, challenges and opportunities in creating data for trustworthy ai. Nature Machine Intelligence **4**(8), 669–677 (2022)

[89] Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=gEZrGCozdqR

[90] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (2018). https://doi.org/10.18653/v1/N18-1202 . https://aclanthology.org/N18-1202

[91] Sun, W., Shi, Z., Gao, S., Ren, P., Rijke, M., Ren, Z.: Contrastive learning reduces hallucination in conversations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 13618–13626 (2023)

[92] Jain, S., Wallace, B.C.: Attention is not Explanation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 3543–3556. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1357 . https://

aclanthology.org/N19-1357

[93] Wiegreffe, S., Pinter, Y.: Attention is not not explanation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 11–20. Association for Computational Linguistics, Hong Kong, China (2019). https://doi.org/10.18653/v1/D19-1002 . https://aclanthology.org/D19-1002

[94] Bibal, A., Cardon, R., Alfter, D., Wilkens, R., Wang, X., François, T., Watrin, P.: Is attention explanation? an introduction to the debate. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3889–3900. Association for Computational Linguistics, Dublin, Ireland (2022). https://doi.org/10.18653/v1/2022.acl-long.269 . https://aclanthology.org/2022.acl-long.269

[95] Lotfollahi, M., Klimovskaia Susmelj, A., De Donno, C., Hetzel, L., Ji, Y., Ibarra, I.L., Srivatsan, S.R., Naghipourfar, M., Daza, R.M., Martin, B., et al.: Predicting cellular responses to complex perturbations in high-throughput screens. Molecular Systems Biology, 11517 (2023)

[96] Li, B., Zhang, W., Guo, C., Xu, H., Li, L., Fang, M., Hu, Y., Zhang, X., Yao, X., Tang, M., *et al.*: Benchmarking spatial and single-cell transcriptomics integration methods for transcript distribution prediction and cell type deconvolution. Nature methods **19**(6), 662–670 (2022)

[97] Tran, H.T.N., Ang, K.S., Chevrier, M., Zhang, X., Lee, N.Y.S., Goh, M., Chen, J.: A benchmark of batch-effect correction methods for single-cell rna sequencing data. Genome biology **21**, 1–32 (2020)

[98] Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., Do, Q.V., Xu, Y., Fung, P.: A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. arXiv preprint arXiv:2302.04023 (2023)

[99] Chang, Y., Wang, X., Wang, J., Wu, Y., Zhu, K., Chen, H., Yang, L., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., Xie, X.: A survey on evaluation of large language models. arXiv preprint arXiv:2307.03109 (2023)

[100] Leek, J.T., Scharpf, R.B., Bravo, H.C., Simcha, D., Langmead, B., Johnson, W.E., Geman, D., Baggerly, K., Irizarry, R.A.: Tackling the widespread and critical impact of batch effects in high-throughput data. Nature Reviews Genetics **11**(10), 733–739 (2010)

[101] Büttner, M., Miao, Z., Wolf, F.A., Teichmann, S.A., Theis, F.J.: A test metric for assessing single-cell rna-seq batch correction. Nature methods **16**(1), 43–49

(2019)

[102] Huang, Q., Liu, Y., Du, Y., Garmire, L.X.: Evaluation of cell type annotation r packages on single-cell rna-seq data. Genomics, proteomics & bioinformatics **19**(2), 267–281 (2021)

[103] Pullin, J.M., McCarthy, D.J.: A comparison of marker gene selection methods for single-cell rna sequencing data. bioRxiv, 2022–05 (2022)

[104] Pliner, H.A., Shendure, J., Trapnell, C.: Supervised classification enables rapid annotation of cell atlases. Nature methods **16**(10), 983–986 (2019)

[105] Ravindra, N., Sehanobish, A., Pappalardo, J.L., Hafler, D.A., Dijk, D.: Disease state prediction from single-cell data using graph attention networks. In: Proceedings of the ACM Conference on Health, Inference, and Learning, pp. 121–130 (2020)

[106] Salzberg, S.L.: Open questions: How many genes do we have? BMC biology **16**(1), 1–3 (2018)

[107] Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C.P., Jerby-Arnon, L., Marjanovic, N.D., Dionne, D., Burks, T., Raychowdhury, R., *et al.*: Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. cell **167**(7), 1853–1866 (2016)

[108] Hecker, M., Lambeck, S., Toepfer, S., Van Someren, E., Guthke, R.: Gene regulatory network inference: data integration in dynamic models—a review. Biosystems **96**(1), 86–103 (2009)

[109] Deshpande, A., Chu, L.-F., Stewart, R., Gitter, A.: Network inference with granger causality ensembles on single-cell transcriptomics. Cell reports **38**(6), 110333 (2022)

[110] Su, C., Xu, Z., Shan, X., Cai, B., Zhao, H., Zhang, J.: Cell-type-specific co-expression inference from single cell rna-sequencing data. Nature Communications **14**(1), 4846 (2023)

[111] Pearl, J., Mackenzie, D.: The Book of Why: the New Science of Cause and Effect. Basic books, ??? (2018)

[112] Li, W.V., Li, J.J.: An accurate and robust imputation method scimpute for single-cell rna-seq data. Nature communications **9**(1), 997 (2018)

[113] Hou, W., Ji, Z., Ji, H., Hicks, S.C.: A systematic evaluation of single-cell rna-sequencing imputation methods. Genome biology **21**, 1–30 (2020)

[114] Marco Salas, S., Czarnewski, P., Kuemmerle, L.B., Helgadottir, S., Mattsson Langseth, C., Tiesmeyer, S., Avenel, C., Rehman, H., Tiklova, K., Andersson, A., et al.: Optimizing xenium in situ data utility by quality assessment and best practice analysis workflows. bioRxiv, 2023–02 (2023)

[115] Schaeffer, R., Miranda, B., Koyejo, S.: Are emergent abilities of large language models a mirage? arXiv preprint arXiv:2304.15004 (2023)

[116] Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. Advances in neural information processing systems **30** (2017)

[117] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N.M., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B.C., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., García, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T.S., Pellat, M., Lewkowycz, A., Moreira, E.O., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K.S., Eck, D., Dean, J., Petrov, S., Fiedel, N.: Palm: Scaling language modeling with pathways. (2022)

[118] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.*: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

[119] Choromanski, K.M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J.Q., Mohiuddin, A., Kaiser, L., Belanger, D.B., Colwell, L.J., Weller, A.: Rethinking attention with performers. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=Ua6zuk0WRH

[120] Wu, C., Mark, A., Su, A.I.: Mygene. info: gene annotation query as a service. bioRxiv, 009332 (2014)

[121] Rubinsteyn, A., Nathanson, T., Kodysh, J., O'Donnell, T., Ahuja, A., Hammerbacher, J., Aksoy, B., Bioinformatics, B., Grouès, V., Hodes, I.: hammerlab/pyensembl, version 1.1. 0. (Zenodo) (2017)

[122] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 27. Curran Associates, Inc., ??? (2014). https://proceedings.neurips.cc/paper_files/paper/2014/

file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[123] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in neural information processing systems **30** (2017)

[124] Doersch, C.: Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908 (2016)

[125] Misra, D.: Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681 (2019)

[126] Church, K.W.: Word2vec. Natural Language Engineering **23**(1), 155–162 (2017)

[127] Schmidt, T.: Coping with copulas. Copulas-From theory to application in finance **3**, 1–34 (2007)

[128] Stephenson, E., Reynolds, G., Botting, R.A., Calero-Nieto, F.J., Morgan, M.D., Tuong, Z.K., Bach, K., Sungnak, W., Worlock, K.B., Yoshida, M., *et al.*: Single-cell multi-omics analysis of the immune response in covid-19. Nature medicine **27**(5), 904–916 (2021)

[129] Sikkema, L., Ramírez-Suástegui, C., Strobl, D.C., Gillett, T.E., Zappia, L., Madissoon, E., Markov, N.S., Zaragosi, L.-E., Ji, Y., Ansari, M., et al.: An integrated cell atlas of the lung in health and disease. Nature Medicine, 1–15 (2023)

[130] Litviňuková, M., Talavera-López, C., Maatz, H., Reichart, D., Worth, C.L., Lindberg, E.L., Kanda, M., Polanski, K., Heinig, M., Lee, M., *et al.*: Cells of the adult human heart. Nature **588**(7838), 466–472 (2020)

[131] Palla, G., Spitzer, H., Klein, M., Fischer, D., Schaar, A.C., Kuemmerle, L.B., Rybakov, S., Ibarra, I.L., Holmberg, O., Virshup, I., *et al.*: Squidpy: a scalable framework for spatial omics analysis. Nature methods **19**(2), 171–178 (2022)

[132] Lin, Y., Cao, Y., Kim, H.J., Salim, A., Speed, T.P., Lin, D.M., Yang, P., Yang, J.Y.H.: scclassify: sample size estimation and multiscale classification of cells using single and multiple reference. Molecular systems biology **16**(6), 9389 (2020)

[133] Stuart, T., Srivastava, A., Madad, S., Lareau, C.A., Satija, R.: Single-cell chromatin state analysis with signac. Nature methods **18**(11), 1333–1341 (2021)

[134] Pellin, D., Loperfido, M., Baricordi, C., Wolock, S.L., Montepeloso, A., Weinberg, O.K., Biffi, A., Klein, A.M., Biasco, L.: A comprehensive single cell transcriptional landscape of human hematopoietic progenitors. Nature communications **10**(1), 2395 (2019)

[135] Muto, Y., Wilson, P.C., Ledru, N., Wu, H., Dimke, H., Waikar, S.S., Humphreys, B.D.: Single cell transcriptional and chromatin accessibility profiling redefine cellular heterogeneity in the adult human kidney. Nature communications **12**(1), 2190 (2021)

[136] Zhu, Q., Gao, P., Tober, J., Bennett, L., Chen, C., Uzun, Y., Li, Y., Howell, E.D., Mumau, M., Yu, W., *et al.*: Developmental trajectory of prehematopoietic stem cell formation from endothelium. Blood, The Journal of the American Society of Hematology **136**(7), 845–856 (2020)

[137] Li, Z., Zhou, X.: Bass: multi-scale and multi-sample analysis enables accurate cell type clustering and spatial domain detection in spatial transcriptomic studies. Genome biology **23**(1), 168 (2022)