Evaluating the Utilities of Large Language Models in Single-cell Data Analysis

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

[1]Department of Biostatistics, Yale University, CT, USA.
[2]Interdepartmental Program in Computational Biology & Bioinformatics, 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

Large Language Models (LLMs) or foundation models have made significant strides in both industrial and scientific domains. In this paper, we evaluate the performance of LLMs in single-cell sequencing data analysis through comprehensive experiments across eight downstream tasks pertinent to single-cell data. By comparing seven different single-cell LLMs with task-specific methods, we found that single-cell LLMs 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 LLMs are promising. In addition, we present a systematic evaluation of the effects of hyper-parameters, initial settings, and stability for training single-cell LLMs based on a proposed **scEval** framework, and provide guidelines for pre-training and fine-tuning. Our work summarizes the current state of single-cell LLMs, and points to their constraints and avenues for future investigation.

Keywords: single-cell data, large language 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 [6], RNA [7, 8], and proteins [9]. Furthermore, single-cell sequencing can facilitate epigenetic studies, including chromatin accessibility [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. Similar to single-cell data, natural language processing (NLP) also boasts extensive datasets, where 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. 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 [19].

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 [20, 21]. Secondly, both require contemplation of intra-data and inter-data relationships [22, 23]. Thirdly, both benefit from high-quality databases [16, 24].

While LLMs have seen marked success in the realms of DNA analysis [25] and biomedical NLP [26, 27], their application in single-cell research remains largely uncharted. There is a limited number of robust pre-trained models (known as single-cell LLMs) capable of managing multiple tasks in single-cell research. Some single-cell LLMs focus on cell-type annotation or gene function prediction, including scBERT [28], tGPT [29], CellLM [30], and Geneformer [31], while others aim to create a foundational model in this area that can handle multiple tasks, including scGPT [32], scFoundation [33] and SCimilarity [34]. 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 provide 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 present a framework for assessing various single-cell LLMs and tasks (shown in Figure 1 (a)), termed 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 LLMs. We also examine the potential emergent abilities of single-cell LLMs, substantiating that the latter also possess distinctive abilities.

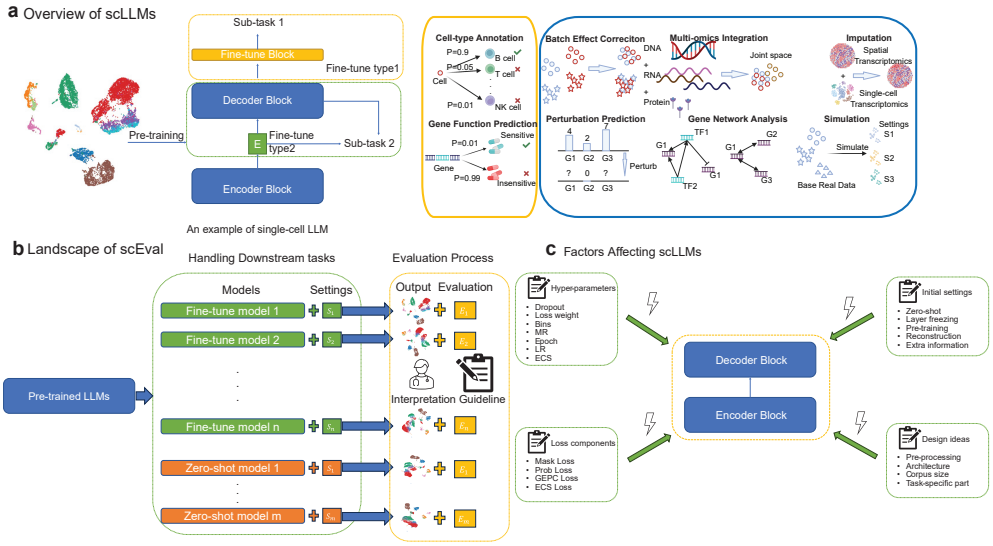

Fig. 1 Overview of single-cell LLMs, landscape of scEval and factors affecting single-cell LLMs. (a): Overview of single-cell LLMs describing the typical structure of LLMs and general tasks for single-cell data analysis. The right two blocks represent two types of downstream tasks. Yellow block: Sub-task 1, including Cell-type Annotation and Gene Function Prediction (top to bottom). Blue block: Sub-task 2, including Batch Effect Correction, Multi-omics 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 analysis. (c): Factors which can affect the performance of single-cell LLMs. 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.

Model	Open Source?	Batch Effect Correction	Cell-type Annotation	Multi-omics Data Integration	Imputation	Gene Function Prediction	Perturbation Prediction	Gene Network Analysis	Simulation
scGPT Benchmark	✓	✓	✓	✓	✓	✓	✓	✓	✓
scGPT	✓	✓	✓	✓			✓	✓	
Geneformer	✓		✓			✓			
scBERT	✓		✓						
CellLM	✓		✓						
tGPT	✓	✓							
SCimilarity		✓	✓						
scFoundation			✓						

✓ Fulfills the criterion

Data for this table were collected until August 1st, 2023.

Better model

Fig. 2 Table of criteria to consider when choosing a model based on the breadth of tasks and usability. The white blanks represent that selected models do not fill the criterion because they do not have such design for specific tasks. Here **scGPT benchmark** represents the modified version we used for this benchmark, **scGPT** represents the model from original paper [32]. The top 3 methods are highlighted.

2 Results

Overview of our evaluations. We evaluated the performance of five open-source single-cell LLMs (scGPT, Geneformer, scBERT, CellLM and tGPT) by assessing the performance of LLM om eight tasks with 22 datasets. We did not evaluate all the models with all the datasets with the rationales of reasons provided in Supplementary files 1 and 2. The tasks that can performed for different models as well as the overall ranks are summarized in Figure 2. We also compared their performance with state-of-the-art (SOTA) task-specific methods. 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 LLMs. Finally, we evaluate the stability and usability of different single-cell LLMs and make recommendations for preferred models. To ensure fairness, we fixed all models to be from versions before August 1, 2023.

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

Batch Effect Correction. We considered scGPT, tGPT and ResPAN [35] for this task. We also provide a detailed analysis of the influence of various hyper-parameters on the performance of scGPT in batch effect correction. The description of the evaluation metric S_{final} can be found in Appendix C. As shown in Figure 3 (a), scGPT outperformed ResPAN in three of the nine datasets and outperformed tGPT in all datasets for batch effect correction, while ResPAN had an overall best correction. In addition, scGPT outperformed the scGPT full model overall, raising the issue of the need for increasing the size of pre-training datasets. For the MHSP dataset, batch effect correction using scGPT was worse than the raw dataset as reflected in the S_{final}. Moreover, scGPT had worse performance in reducing the batch effect for large-scale datasets, as their biology conservation scores were lower than those of raw datasets (Extended Data Figure 1). Extended Data Figures 2 and 3 show the Uniform Manifold Approximation and Projection (UMAP) [36] plots for the raw data and the scGPT results. We could still observe batch effect in the output of scGPT on certain datasets. Therefore, single-cell LLMs do not show advantage 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 Figure 3 (b) and Extended Data Figure 4. 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 [37], 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.

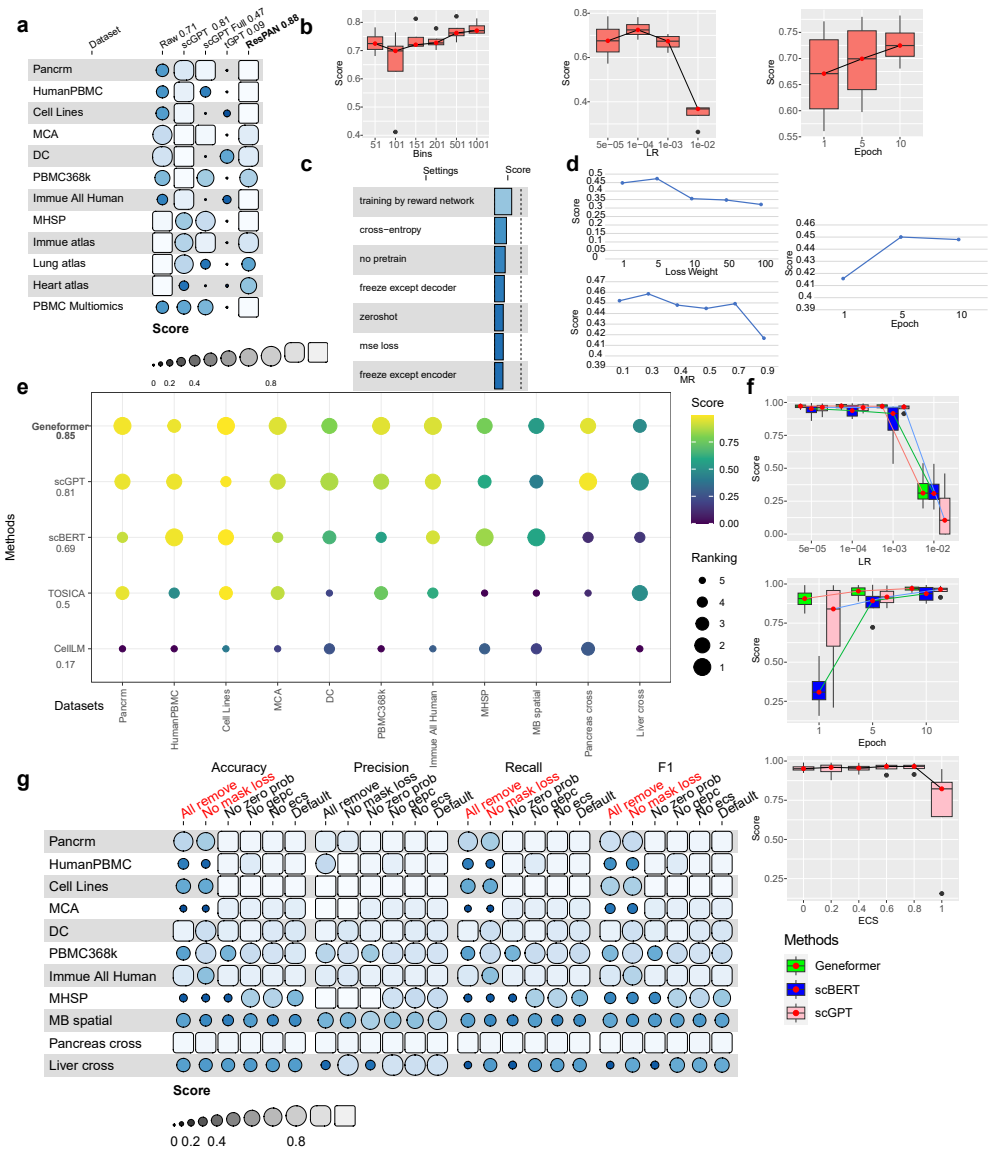

Fig. 3 Experimental results of single-cell LLMs and benchmarking methods for cell-level tasks. (a): An overall assessment of raw data and data after batch effect correction based on different methods. scGPT full represents scGPT model with larger pre-training datasets comparing to scGPT. (b): The effect of hyper-parameters including Bins, Learning rate (LR), and Epoch for scGPT training in the Batch Effect Correction task. (c): Results of different initial settings for the multi-omics data integration task. (d): The effect of hyper-parameters including loss weight, mask ratio, and epoch for scGPT training in the Multi-omics Data Integration task. (e): Comparison among models in the Cell-type Annotation task. The scores on the left represent the average accuracy of different models across different datasets. (f): The effect of hyper-parameters including LR, Epoch and ECS for scGPT training in the Cell-type Annotation. LR and Epoch are shared hyper-parameters for Geneformer, scBERT and scGPT. (g) Ablation tests of the loss function components for cell-type annotation. The red component is significant (p-value¡0.05).

Extended Data Figure 5 (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 tasks. 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 loss for most datasets. Freezing weights is not crucial for batch effect correction. Interestingly, the encoder structure appears to play a more significant role in the training process, as freezing the encoder layer led to a larger decrease in score. Incorporating cell type 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 5 (b) shows the performance metrics versus the choices of different optimizers for the batch effect correction. Adam and AdamW [38] had comparable performance in this task, while SGD [39], Sophia-G [40] (a novel optimizer which is designed for training LLMs) and Lion [41] (an optimized version of Adam) were worse. Therefore, the optimizers in the Adam family are still preferred.

Extended Data Figure 5 (c) illustrates the impact of different loss function components on the performance of batch effect correction using scGPT. 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 approach 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 LLMs for batch effect correction, with each loss function component contributing differently to the model performance.

Multi-omics Data Integration. For this task, we seek to integrate single-cell RNA-sequencing (scRNA-seq) datasets with single-cell ATAC-sequencing (scATAC-seq) datasets. We assessed the integration quality through the same score as Batch Effect Correction. The results presented in Figure 3 (c) summarize the impact of initial setting choices on the performance of scGPT for the multi-omics integration tasks. As was the case with batch effect correction, the cross-entropy loss function led to better performance compared to the Mean Squared Error (MSE) loss for this task. Interestingly, pre-training did not significantly influence the performance for this task. The encoder part of the single-cell LLM played a more important role than the decoder. 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.

We illustrate the evaluation metrics versus different parameter settings in Extended Data Figure 5 (d) and Extended Data Figure 6. scGPT did not perform well on this task as shown by the low score (below 0.5). This is also shown in the UMAP results.

Certain parameters affected the training process, with a smaller weight for the loss function and mask ratio, and more epochs, improved the model's performance. Setting the learning rate too high caused the model to collapse.

Cell-type Annotation. We considered Geneformer, scGPT, scBERT, CellLM and TOSICA [23] for this task. We assessed the performance of different single-cell LLMs in assigning cell types based on the four metrics discussed in Appendix C.2. The UMAPs for the raw data and scGPT are shown in Extended Data Figures 7 and 8. We observed accurate annotation results on these UMAPs. Figure 3 (e) displays the Accuracy for different datasets for the five models. On average, models with pre-training performed better than those without pre-training. However, CellLM did not perform well across all the datasets. Moreover, for the intra-dataset prediction task, all the single-cell LLMs were comparable even if they had different pre-training settings. For the inter-dataset prediction task, scGPT was worse than Geneformer but better than scBERT. For cross-species cell type annotation, the performance of Geneformer, scGPT and scBERT were comparable for the MCA dataset, but scGPT did not perform well for the MHSP and MB spatial datasets. Different single-cell LLMs also had large divergences in performance. CellLM had running errors for Pancrm, HumanPBMC, PBMC368k, and Liver cross datasets, which raises the issue of the reliability and usability of this model.

In Figure 3 (f) and Extended Data Figure 9, 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 LLMs under the condition of altering their shared hyper-parameters. For Geneformer and scBERT, lower learning rate and higher epoches 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 scartch*, where we did not use the pre-trained weights. Extended Data Figure 10 (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 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 the 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 10 (b).

In Extended Data Figure 11, we explored 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, and the results are shown in Figure 3 (g). Here we included three extra metrics, and details can be found in Appendix C. Based on the Accuracy of Figure 3 (g), the existence of mask loss is important. Moreover, the default setting is generally good across different tasks. Based on precision and recall of Figure 3 (g), the effect of different loss function terms had less effect on precision and more effect on recall. Such 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, single-cell LLMs can handle the Cell-type Annotation task with suitable pre-training data and model structure, but one pre-training framework is not consistently good across all the datasets.

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

Gene Function Prediction. We considered Geneformer, scGPT and *vanilla* NN for this task. The evaluation results with the same four metrics as the Cell-type Annotation task are shown in Figure 4 (a). On average, Geneformer and scGPT performed well in this task, and there is a performance gap between single-cell LLMs and *Vanilla* NN.

Figure 4 (b) and Extended Data Figure 12 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 13 (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 13 (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 13 (c) shows the ablation test results of scGPT for this task. There was no significant difference by comparing the default setting and those the model without certain components. Therefore, the task-specific loss function is the most important design for this task.

Perturbation Prediction. We considered scGPT and GEARS [42] for this task. In the training process, we masked the genes under perturbation and tried to reconstruct the expression levels of all the genes of the input cells rather than only the masked genes. 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. The datasets include two perturbation conditions: a single-gene perturbation and a double-gene perturbation. Based on our experiments, scGPT could predict gene perturbation with relatively high MPC.

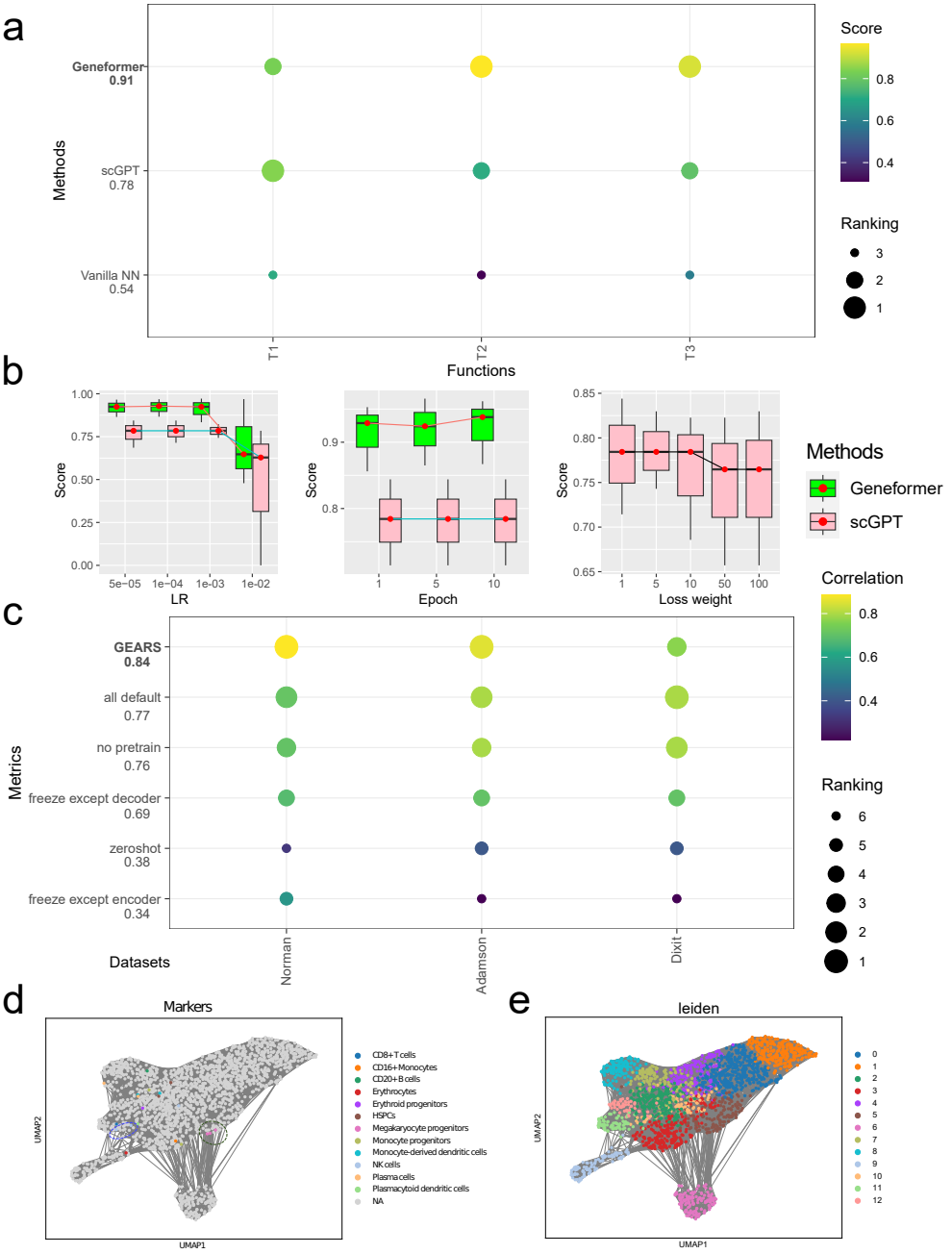

Fig. 4 Experimental results of single-cell LLMs and benchmarking methods for gene-level tasks. (a): Comparisions among Geneformer, scGPT and *vanilla* NN in the Gene Function Prediction task. (b): The effect of hyper-parameters including Loss weight, Bins and Learning rate for scGPT and Geneformer in 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 colored by the marker genes of different cell types. (e): Dataset-level gene embeddings colored by the Leiden cluster.

Figure 4 (c) summarizes results for different initial settings of scGPT for 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. Interestingly, this option did not lead to a large increase in error, suggesting that the encoder carries a significant portion of the important information needed for the prediction task.

Regarding the effect of hyper-parameters, Extended Data Figure 14 shows that scGPT is sensitive to adjusting the learning rate and epochs. Decreasing learning rate and increasing the number of epochs improved MPC. Increasing the dropout rate decreased MPC. The rest of hyper-parameters did not contribute much in this task.

Extended Data Figure 15 (a) shows that AdamW and Adam are comparable for scGPT in perturbation prediction. The rest of the optimizers could significantly reduce the performance of scGPT. Moreover, based on the ablation tests shown in Extended Data Figure 15 (b), we found that different loss function components, except the masked gene expression reconstruction loss, did not contribute much in this task. Therefore, the task-specific loss component is the most important design for perturbation prediction.

Gene Network Analysis. The GRNs defined in scGPT are GCNs, because the construction process is based on embeddings similarity. In the effort to infer GCNs using scGPT, two types of GCNs are defined: 1. Type 1 GCN (Tissue-specific GCN): This type of GCN is generated by applying the scGPT model to the entire dataset under a zero-shot learning framework to create gene embeddings. The Pearson correlation 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 scGPT model to generate cell-type-specific gene embeddings under the zero-shot learning framework, and cosine 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) [43] for cell-type 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 [44, 45].

We used the Immune Human Atlas dataset to evaluate the performance of inferring these two types of GCNs. Since we knew the gene enrichment information of this dataset as prior knowledge, we could evaluate the performance of scGPT based on this dataset. The known information including marker genes, cell types, and go pathways was utilized to evaluate the performance of the GCNs inferred by scGPT.

In the analysis of the Immune Human Atlas dataset, we also considered another approach to define the GCN similar to scGPT's definition. We considered finding the nearest neighbors of genes based on gene embeddings. The focus was initially on Type 1 GCN, with the results presented in Figures 4 (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 Immune Human Atlas

dataset [46]. As this relationship was determined based on k-nearest neighbors, it can be viewed as a form of gene co-expression relationship. From Figure 4 (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 4 (e), on the other hand, represents the cluster labels based on the Leiden clustering method [47]. These clusters can be interpreted as groups of genes that share common functions, or "gene co-function clusters". For marker genes from other cell types, some of them are in different clusters shown in Figure 4 (e), 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.

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 Figure 16 (a) and (b) focus on gene embeddings categorized by cell types. Extended Data Figure 16 (a) shows 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 this could be due to two reasons. One reason is that the quality of gene embeddings was unsatisfactory. Since scGPT adopts the zero-shot learning scheme embeddings, it does not incorporate cell-type-specific information. The other reason is that the complex biological network in the human immune system makes the communication between cell-cell or gene-gene difficult to decompose [48, 49]. Additional analysis is needed for gene embeddings. Extended Data Figure 16 (b) shows the cluster results based on the Leiden algorithm [47], which can be interpreted as co-functional gene groups. Most of the clusters contained marker genes of different cell types.

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 dataset, a system known for its complexity due to interactions among various cell types. Using the scGPT model, the goal was to establish GCNs for the immunology system and validate them against the established CD antigen networks. We omitted the analysis for HLA genes because HLA genes carry a higher risk for misreading based on previous research [50–53], and the network result is not reliable. Therefore, we selected genes with the prefixes CD-, and utilized the embeddings of the remaining genes to calculate the GCNs. A publicly accessible database containing established GCN relationships - the Immune System R-HSA-168256 pathway from the Reactome 2022 database [54] - was used as a reference. The GCN was constructed based on the correlation between the two genes. For the CD- genes (3rd gene set), the overlap was only 0.0, indicating that the inferred GCN is not especially reliable (less than 0.5).

The scGPT results for major CD- genes are shown in Extended Data Figure 16 (c), which shows the GCN for major CD- genes. The value above the edges represents the strength of correlation. The threshold of strong correlation here is set as 0.4. Our results highlight the importance of critical evaluation and cross-referencing in the development of gene regulatory networks, as well as the potential and limitations of using machine learning models like scGPT for this purpose.

We also note that GCN inferred based on interactivity could only imply gene regulation because we could not determine the direction of regulation. We need extra information, for example, a list of transcription factors with known regulatory relations, to extract the unified or standard GCN.

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

Imputation. We considered scGPT and Tangram [55] for this task. The evaluation metrics for this task based on clustering and correlation are provided in Appendix C. The imputation results for scRNA-seq are summarized in Figure 5 (a), which suggest 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 5 (b), scGPT performed well in the spatial transcriptomic data imputation task compared to the SOTA spatial imputation method, Tangram [55, 56]. 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 17 shows the results of Deferentially Expressed Genes (DEGs) discovery based on pre-imputation data and post-imputation data. Results in Extended Data Figure 17 (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 17 (b), we found that DEGs after imputation based on scGPT did not contain MT genes. These genes were identified as DEGs in the raw dataset. MT genes should be omitted in the downstream analysis [57]. Moreover, based on Extended Data Figure 17 (b) and (c), the DEG patterns after imputation based on scGPT and Tangram are similar. Thus, scGPT has the potential to produce biologically meaningful imputation results.

Simulation Analysis. We considered scGPT and scDesign3 [58] for this task. We benchmarked the output of scGPT against the output of scDesign3 [58]. 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 Figure 5 (c)-(e), scDesign3 outperformed scGPT across two conditions of the simulation task. 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

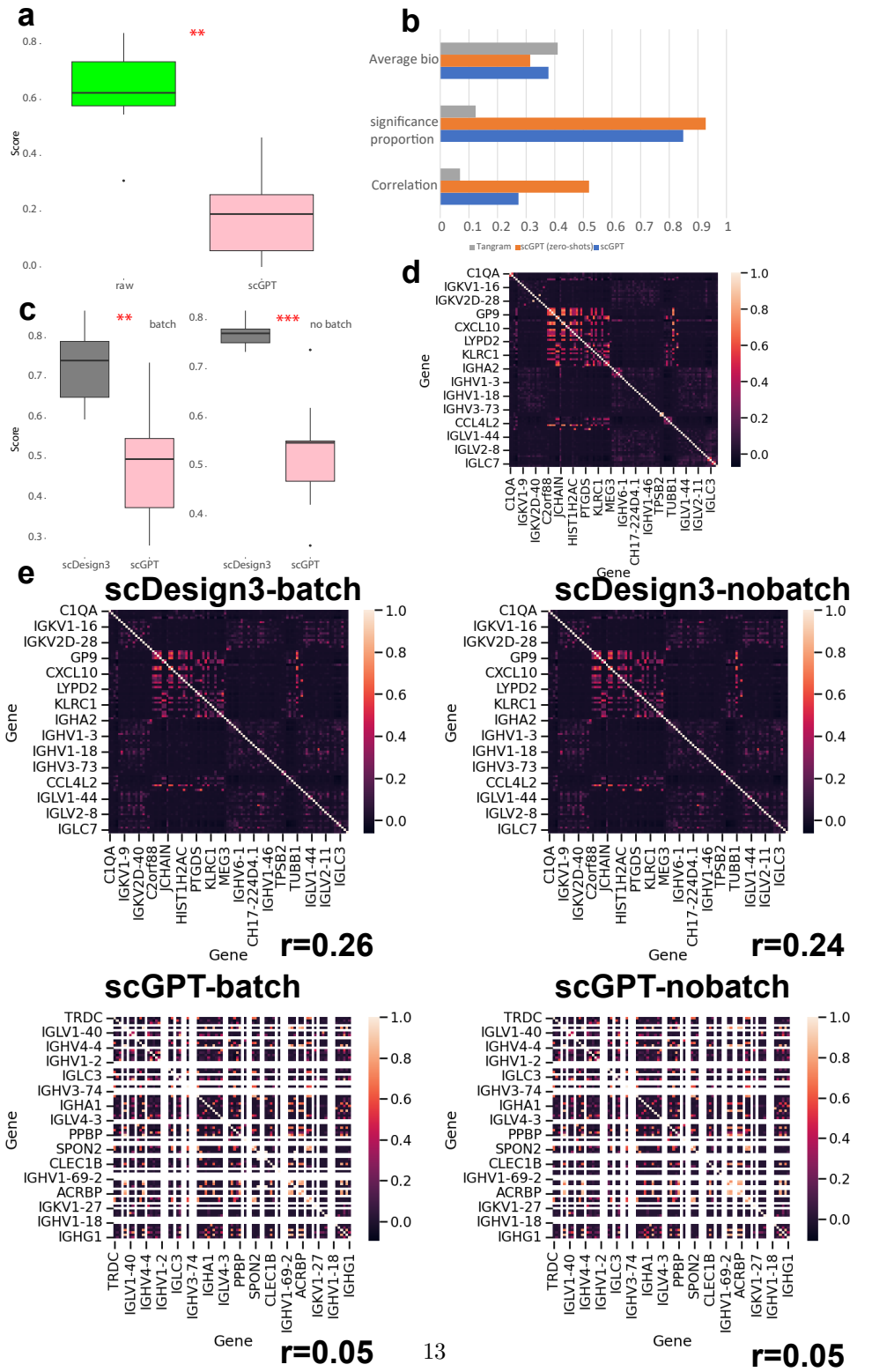

Fig. 5 Experimental results of the Imputation task and the Simulation task. The significance level was computed based on paired Student's t-test. The number of stars represents the significance level ($*** : p - value < 0.005, ** : p - value < 0.05$). (a): Comparison of the average bio score between 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 between scDesign3 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 the HumanPBMC dataset.

shown in Figure 5 (d)-(e). The gene-gene correlation from scDesign3 is also more similar to the gene-gene correlation of the raw data. Therefore, the simulation task needs to be improved for single-cell LLMs.

In addition, we present UMAPs of the output produced by different methods in Extended Data Figures 18 and 19 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.

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

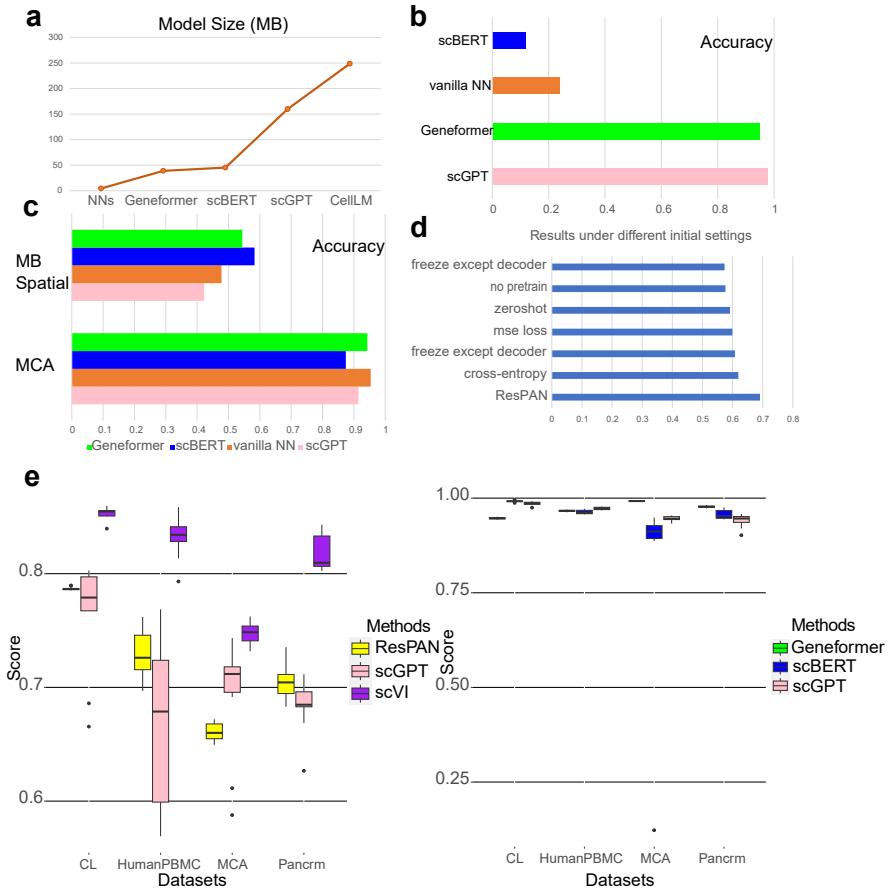

Fig. 6 Different comparison groups for emergent ability analysis and stability analysis. (a): The model scale of different methods. (b): Accuracy of LLMs and *vanilla* NN in Cell-type Annotation task. The dataset here is the Pancreas cross dataset. (c): Accuracy of LLMs and *vanilla* NN in Cell-type 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.

Emergent Ability Analysis. In this section, we explored the emergent abilities of single-cell LLMs, with scBERT, Geneformer, and scGPT. We considered three scenarios to investigate the emergent abilities: cross-data cell type prediction, cross-species analysis, and spatial transcriptomics analysis.

Based on [33], we compared models with large parameter size with pre-training to models with 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. Importantly, in the context of batch effect correction, we restricted our evaluation to batch-related metrics. This is primarily because we lack the information needed on the type of cell or cluster for human spatial transcriptomic data.

In the first scenario, our anticipated emergent ability would be a significant improvement in prediction accuracy for single-cell LLMs compared to *vanilla* NNs of a smaller size. Figure 6 (a) provides an overview of the different model sizes. From Figure 6 (b), it is evident that scGPT and Geneformer outperformed NNs in terms of accuracy in cross-data scenarios, suggesting the emergent abilities in the cross-data cell-type annotation task. The low accuracy of scBERT may be caused by its default fine-tuning setting and/or the pre-trained model weights.

In the second scenario, the desired emergent ability mirrors that of the first task. Nonetheless, there was no enhancement in performance from Figure 6 (c). 6 (c) shows that *vanilla* NN outperformed scGPT for both the MCA and MB Spatial datasets. *vanilla* NN was comparable to Geneformer for MCA dataset and worse than Geneformer for 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 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 6 (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 without pre-training yielded subpar results. However, the performance of scGPT in the integration of spatial data was still worse than that of ResPAN. Figure 6 (c) shows that we did not detect any emergent ability pertaining to the second component in the MB Spatial data annotation task.

Stability Analysis. To analyze the stability of single-cell LLMs, we selected Batch Effect Correction and Cell-type Annotation as two representative tasks and varied the seeds of single-cell LLMs to investigate the model stability. 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 LLMs should not vary substantially across different datasets. We also considered stability for other benchmarking tools. Our experiment results showed that the stability for single-cell LLMs is task-specific.

Based on Figure 6 (e), the variance of scVI [59] and ResPAN was generally lower than that of scGPT, and scVI and ResPAN also had a higher score on average. Therefore, single-cell LLMs were not as stable as SOTA deep-learning-based methods for the Batch Effect Correction task.

Figure 6 (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 LLMs might fail under certain random seeds.

3 Discussion

In this paper, we have evaluated the performance of single cell LLMs on eight different tasks in single cell data analysis. The ranks of single-cell LLMs are shown in Figure 2, where we not only consider the broader functions of the models, but also their usability. For example, 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 20.

We studied the impacts of the fine-tuning process. The pre-training phase plays an important role across multiple tasks, particularly for Cell-type Annotation and Gene Function Prediction. We observed a considerable performance improvement for these two tasks with pre-training. Moreover, even if the performance of single-cell LLMs is not outstanding, they can still annotate cell types with high accuracy. Thus, single-cell LLMs 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 cross-species task performance. As an illustration, pre-training using human single-cell datasets enabled scGPT to handle Batch Effect Correction and Cell-type Annotation tasks on mouse datasets effectively.

We also observed the emergent ability of single-cell LLMs. 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 in analyzing spatial transcriptomic data, which have attracted much attention in recent years.

Interestingly, we found that the optimal optimizer for single-cell LLMs was still Adam. When other factors are 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 LLMs [60].

For the Cell-type Annotation task, we compared results among scGPT, scBERT and Geneformer. Geneformer was generally better than scGPT and scBERT in this

task. Despite its smaller size, scBERT occasionally outperformed scGPT on certain datasets. 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, we showed that the pre-training process and gene embeddings 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 data 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 LLMs can transfer knowledge across different omic data.

Thus, the contributions of single-cell LLMs 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 LLMs, there are evident limitations in their construction and training steps. A more comprehensive understanding of scGPT 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, six parts, with five 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.

For the Batch Effect Correciton task, scGPT 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 generally better than scGPT in this task. For the data simulation task, scGPT did not perform very well, one reason is from [61], which suggests that LLMs are remembering things rather than making inferences or generating enough novel information.

In our stability analysis, we found that single-cell LLMs 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 LLMs 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 [62] or Scanpy [57]) with various functions.

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

Although some tasks were only evaluated based on scGPT, the patterns we found in scGPT could be extened in the training of other single-cell LLMs based on Figure

3 (f). We have identified several beneficial factors that can optimize the performance of single-cell LLMs: 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 maintain performance levels while saving training time. The priority for freezing the **decoder** is higher than that of the encoder. Given that we have numerous tasks requiring the resolution of embeddings, freezing the encoder is not advisable. 3. Integrating **cell-type** into the fine-tuning process yields benefits in multiple tasks. 4. **Adam** is the most suitable optimizer for single-cell LLMs. 5. Possessing a GPU with **larger memory** (e.g., RTX3090, A6000, and A100) is advantageous for the fine-tuning process. 6. Cross-entropy loss is the optimal choice for masked gene expression reconstruction. 7. We provide a table shown in Figure 2 to evaluate the number of tasks that different single-cell LLMs can handle. Models with the **scGPT** framework achieve the best score based on consideration of both usability and stability.

The application of LLMs to single-cell data remains a promising avenue of exploration, given its impressive performance in Cell-type Annotation and perturbation prediction tasks. Since there is already plenty of work discussing how to optimize general LLMs [63–66] (including efficient tuning, model compression, and other research directions), we focus specifically on how to better train and apply single-cell LLMs. Therefore, several potential research topics and future directions are proposed here:

In terms of data collection, 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 LLMs otherwise the performance may be harmed [67, 68]. The qualify of pre-training data can also be verified with the online learning framework [69]. We can evaluate the performance of single-cell LLMs on downstream tasks and then choose to include some datasets in the pre-training or not. Moreover, the enrichment of biological information is crucial for the success of a foundational model in biology. With tissue information, we can also try to offer domain-specific or tissue-specific LLMs for single-cell data analysis.

As for model training, both the pre-training and fine-tuning steps 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 let 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 LLM training [70]. Using poisoned training single-cell datasets (for example, wrong data or make-up data) as an attack can test the robustness of single-cell LLMs. For model fine-tuning, instruction tuning [71] is a potential direction to explore. In this context, cells could be considered as prompts, as described in scGPT.

For model evaluations, we need more effective methods to assess results for certain tasks. For instance, evaluating the quality of gene embeddings solely based on genes from a single pathway is simplistic, and we may need to confirm the biological function

of genes using different approaches, including verification from biology experiments to avoid the harm of model hallucinations [72]. Moreover, we should also account for the pre-training cost (such as training time and power consumption). This information can guide users in selecting the most suitable single-cell LLM 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 [73–75]. Moreover, we should consider more meaningful and representative tasks related to single-cell and spatial data for single-cell LLMs. For instance, tasks such as cell-type deconvolution, marker gene selection, and spatial transcriptomics imputation warrant exploration.

In summary, the goal of studying single-cell data-based LLMs can be to come up with a foundation model that is capable of performing multiple tasks with stable and reliable results. Such a model should also be user-friendly with detailed tutorials or websites. Although our benchmark is subject to the number of current single-cell LLMs and the size of pre-training datasets, we hope that our analysis can provide insights into best practice and guide the development of ideal LLMs for single-cell data analysis.

4 Methods

Problem definition. We consider a pre-trained LLM, 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 LLMs 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 k. Formally, we denote the loss for distinct task k as $L_k(\cdot, \cdot)$, and use the evaluation dataset $\mathcal{D}_{\text{eval}} = \{x_i, y_i\}_1^n$, 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 LLMs, 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 LLMs, including: 1. zero-shot learning; 2. emergent ability [76]; 3. cross-species data analysis; 4. biological mechanism exploration; 5. stability.

Parameters and tasks. Most single-cell LLMs share the pre-training process. By considering the overlap among various single-cell LLMs, 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-omics Data Integration; 3. Cell-type Annotation; 4. Gene Function Prediction; 5. Perturbation Prediction; 6. Gene Network Analysis; 7. Imputation; 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 four representative datasets: 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*.

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 [77]. 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 problem, the metrics we consider here are inspired by scIB [46], 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) and Graph Connectivity (GC) 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-omics Data Integration. Multi-omics Data Integration is a key for multi-omics data analysis [21]. It is akin to an advanced form of batch effect correction. If unpaired multi-omics data are present, the objective is to map different datasets into a shared space for subsequent analysis. If paired multi-omics data are present, the goal is to assess whether the use of multi-omics 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 use the same metrics for multi-omics data integration as those for Batch Effect Correction.

Cell-type Annotation. Cell-type Annotation is another key step following single-cell data pre-procesing [78]. This step annotates each cell with its accurate cell type label, which can be achieved through prior knowledge [79] or computational methods [80]. 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 [81] can also be treated as a variation of this task. A common approach employed by single-cell LLMs 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 which originate from studies different from the training 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 the cell-type annotation task, we choose 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 Pancreas cross from pancreas, and Liver cross from 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 comparision, we consider 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*.

Gene Function Prediction. Gene Function Prediction is important to identify the properties of genes across different conditions [31]. Since we have approximately 20,000 protein-encoding genes for humans [82] 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 gene. 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 gene. 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 [42] 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 [83]. In perturbation prediction, we intend to predict the gene expression level after gene editing. This task contains may predict seen gene perturbation in the testing datasets (a 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 [42], which contains cells with three conditions: 1. Control; 2. One gene perturbation; and 3. Two genes perturbation. In the evaluation process, we combined case 2 and case 3. Here we select GEARS [42] as a benchmarking tool.

Gene Network Analysis. Gene Network Analysis is a downstream task for single-cell datasets [84]. 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 [85]. A GCN can be used to analyze genes with similar functions or uncover the characteristics of genes in some diseases [86]. GCN and GRN are two different tasks because correlation does not imply causal relation [87]. 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 consider using the overlap between ground truth genes and inferred genes as a 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 technology zeros with biologically meaningful values [88, 89]. 2. Perform imputation for spatial transcriptomic data because of unseen or unmeasured genes [55, 56]. According to [90], current spatial imputation methods do not show strong performance across different datasets. Using single-cell LLMs, 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 LLMs. 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 compare 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 is using scRNA-seq to perform fine-tuning and performing inference based on spatial transcriptomic data. The second setting is using a zero-shot learning framework to directly perform inference based on spatial transcriptomic data. We considered 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 new 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 [91], 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. 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 task 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. [76, 92] argue that due to the extensive number of parameters, LLMs can manage specific tasks that they have not been explicitly trained to handle, an attribute referred to as 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 LLMs 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 LLMs. Such scenarios include: 1. Cross-species Cell-type Annotation and 2. Spatial transcriptomic data analysis.

Ablation tests. Given that there is no existing work investigating the significance of different loss function components of single-cell LLMs, we conduct 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 employ here is the paired Students' t-test. The details of the different loss function components are enumerated 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,

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

where $\tilde{\boldsymbol{x}}^{(i)}$ represents the predicted results of the row of the expression for cell i, while $\tilde{\boldsymbol{x}}_j^{(i)}$ represents the ground truth of the gene j's expression in cell i. $\boldsymbol{h}_n^{(i)}$ 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 LLMs 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 masked gene expression levels 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:

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

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. LogProb is the log probability based on the relationship between gene expression levels of cell i and zero.

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,

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

where $\text{emb}_{\text{g}}(\boldsymbol{t}_g^{(i)}$ 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 $\mathcal{L}_{\text{mask}}$.

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:

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

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 LLMs. 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 the first version of scGPT as a baseline and representative model because of three reasons. Firstly, scGPT is an open-source single-cell LLMs model with 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. Moreover, different single-cell LLMs 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 and CellLM, we also evaluated their performance based on shared parameters with scGPT to verify if our rules found in scGPT can be extended for other single-cell LLMs.

5 Data availability and reproducibility

We used the resources from the Yale High Performance Center (Yale HPC) to conduct all of the experiments. 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 explanation from Yingxin Lin, Haotian Cui, Christina Theodoris and Rex Ying.

7 Author contributions

T.L. designed the study with H.Z. and Y.W. T.L. run 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.

A Hyper-parameters

Here we explain the definition of different hyper-parameters.

1. Dropout: The dropout rate means the probability of making the neural unit inactive or closed. If a unit is closed, it will not participate in the training process but will take part in the inference process. The design of dropout is to avoid or reduce overfitting. The scope is [0, 0.1, 0.2, 0.3, 0.5].

2. Loss weight: Changing the weight of different loss function components is a well-known approach to increasing the performance of neural networks. Here the loss weight means the weight of the task-specific loss component. For example, for the Batch Effect Correction task, such weight represents the loss weight for the loss of the gradient reverse network. The scope is [1, 5, 10, 50, 100].

3. Bins: To reduce the batch effect during the pre-training or fine-tuning process in the pre-processing step, the gene expression is divided into different intervals and the endpoint values of the intervals are used to replace the original gene expression. The intervals here are the bins. Formally, the binned expression value $x_j^{(i)}$ for cell i is:

$$x_j^{(i)} = \begin{cases} k, & \text{if } x_j^{(i)} > 0 \text{ and } x_j^{(i)} \in [b_k, b_{k+1}]\,, \\ 0, & \text{if } x_j^{(i)} = 0. \end{cases}$$

where $k \in \{1, 2, ..., B\}$ and B represents the number of bins. The scope is [51, 101, 151, 201, 501, 1001].

4. Mask ratio: In the pre-training and fine-tuning process, we choose to mask a fixed ratio of gene expression levels and try to use our decoder to reconstruct it. Such a ratio is defined as a mask ratio. The scope is [0.1, 0.3, 0.4, 0.5, 0.7, 0.9].

5. Epoch: Epoch represents the number of steps we use to train our model based on the whole training dataset. The scope is [1, 5, 10].

6. Learning rate: Learning rate is designed for the optimizer, it represents the step size of gradient descent. The scope is [5e-5, 1e-4, 1e-3, 1e-2].

7. ECS threshold: The threshold β mentioned in the ECS Loss represents the upper bound for the similarity between paired cells. The scope is [0, 0.2, 0.4, 0.6, 0.8, 1.0].

B Initial Settings

In this section, we delineate the initial settings for Batch Effect Correction, Multi-omics Data Integration, Cell-type Annotation and Perturbation Prediction tasks.

1. Zero-shot learning: This refers to using our pre-trained model without any further training for unseen datasets on specific tasks. This setting aids in validating the significance of pre-training.

2. No pre-training: This setting entails starting the fine-tuning process without any pre-trained weights, allowing us to train a *de novo* single-cell LLM. This configuration helps ascertain the importance of pre-training.

3. Cross-entropy (default setting): Here, we utilize the cross-entropy loss function for the reconstruction of gene expression levels, which is the default setting in scGPT.

4. Mean Squared Error (MSE) loss: In this setting, we employ the MSE loss function for the reconstruction of gene expression levels.

5. Freeze except encoder: This setting involves freezing the weights of the encoder component during the fine-tuning process.

6. Freeze except decoder: Here, we freeze the weights of the decoder section during the fine-tuning process.

7. Training by reward network: The successful application of Human-feedback Reinforcement Learning (HFRL) to the training process for LLMs has been established [93], and we borrow ideas from [94, 95]. We consider cell types as the human label for

different cells, akin to the labels of sentences in the NLP area. A reward network is employed to predict the cell-types of the provided cells and minimize the loss in this process. This procedure can also be integrated into the pre-training process or prompt-based learning framework. This approach can also be regarded as joint fine-tuning for Batch Effect Correction and Cell-type Annotation.

The results from this analysis provide valuable insights into how to configure scGPT most effectively for this specific task.

C Metrics Information

Here we describe details about the metrics we used in the evaluation process for different tasks.

C.1 Batch Effect Correction, Multi-omics Data Integration and Simulation

1. Normalized Mutual Information (NMI): NMI is a score to evaluate the performance of biological information preservation. We calculate this score based on computing the mutual information between the optimal Leiden clusters and the known cell type labels and then take the normalization. Therefore, NMI $\in (0, 1)$ and higher NMI means better performance.

2. Adjusted Rand Index (ARI): ARI is a score to evaluate the performance of biological information preservation. ARI can measure the agreement between optimal Louvain clusters and cell type labels. ARI $\in (0, 1)$ and higher ARI means better performance.

3. Average Silhouette Width (ASW): Here we have cell type ASW (ASW_{cell}) and batch ASW (ASW_{batch}). For one cell point, ASW calculates the ratio between the inner cluster distance and the intra cluster distance for this cell. Therefore, higher ASW_{cell} means better biological information preservation and lower ASW_{batch} means better batch effect correction. To make them consistent, for ASW_{cell}, we take the normalization, that is:

$$ASW_{cell} = \frac{ASW_{cell}^{raw} + 1}{2}.$$

Similarly, for ASW_{batch}, we take the inverse value of the normalization, that is:

$$ASW_{batch} = 1 - \frac{ASW_{batch}^{raw} + 1}{2}.$$

Both of metrics are in (0,1), and a higher score means better model performance. For multi-omics data integration, all the metrics are the same.

4. Graph Connectivity (GC): GC means the connectivity of cells in different cell types. If the batch effect is substantially removed, the connectivity of cells of the same cell type from different batches will have a higher connectivity score based on the k-NN neighbor graph. Therefore, we can calculate the GC score for each cell type and take the average. GC score is in (0,1) and higher means better batch effect correction.

5. Principal Component Regression (PCR): PCR is a metric to evaluate the performance of batch effect correction. We calculate the R^2 for a linear regression of the

covariate of interest onto each principal component. The variance contribution of the batch effect for all the PCs is based on the sum of the product between the variance of each PC and the R^2 of each PC across all the PCs. Therefore, the score can be represented as:

$$PCR = \sum_{i=1}^{G} \mathrm{Var}\left(C \mid \mathrm{PC}_i\right) \times R^2\left(\mathrm{PC}_i \mid B\right).$$

where G denotes the number of PCs and B denotes the batch information. PCR is in (0,1) and a higher score means better performance.

These metrics are also used for Multi-omics Data Integration. For simulation analysis, we only evaluated the biological information preservation for the simulation output with batch effect setting.

C.2 Cell-type Annotation and Gene Function Prediction

1. Accuracy: We calculate the accuracy based on the ratio between the total number of cells (or genes) being classified correctly and the total number of cells (or genes).

2. Precision, Recall, and F1: For classification results, we calculate true positives (tp), false positives (fp), true negative (tn), and false negatives (fn). Precision, Recall, and F1 score are calculated as follows:

$$\begin{aligned} Precision &= \frac{tp}{tp + fp}, \\ Recall &= \frac{tp}{tp + fn}, \\ F1 &= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \end{aligned}$$

Note that precision, recall, and F1 are calculated based on the weighted average, where the weight of each cell type is determined by the cell number.

For the Gene Function Prediction task, all of the metrics are the same but for genes rather than cells.

C.3 Perturbation Prediction

We use the Mean Pearson Correlation (MPC) in the validation dataset to evaluate the performance of scGPT across different datasets.

The Pearson correlation for the predicted gene expression g_i' of cell i and the ground truth gene expression g_i of cell i is:

$$\begin{aligned} S_{\mathrm{PC}i} &= \frac{g_i' \cdot g_i}{||g_i'|| \cdot ||g_i||}, \\ S_{\mathrm{MPC}} &= \frac{\sum_i^n S_{\mathrm{PC}i}}{n}. \end{aligned}$$

Higher S_{MPC} means better performance.

C.4 Gene Network Analysis

Here we extract the genes starting with HLA and genes starting with CD from the specific pathways. Moreover, we extract genes that have significant correlations from our gene embedding dataset. We then measure the Jaccard similarity for these two gene sets G_{path} and G_{net} as follows,

$$Jaccard = \frac{|G_{path} \cap G_{net}|}{|G_{path} \cup G_{net}|}.$$

C.5 Imputation

1. Average bio: The average bio score is calculated based on cell-type NMI and ARI scores using the latent space.

2. Correlation: The correlation score is calculated based on the average correlation between the raw data and imputed data for known common genes. The idea of this metric is coming from the assumption that the target of spatial imputation is to predict the expression of missing or unmeasured genes while keeping the distribution of known gene expression levels. We use Pearson correlation here.

3. Significance proportion: The significance proportion means we compute the proportion of genes with the p-value of correlation smaller than a threshold, which is 0.005 in our evaluation. The proportion can be used as a metric to evaluate the similarity between two paired data.

D Benchmarking Methods

In this paper, we discuss extra models as benchmarking methods. Here we describe the details of the implementation of these methods.
Single-cell LLMs include:

- tGPT: tGPT is a single-cell LLM based on the GPT-2 [96] structure. It utilized large-scale scRNA-seq datasets for pre-training and set the pre-training task as predicting gene expression rankings. The downstream applications of tGPT follow the zero-shot learning framework and include clustering, batch effect correction, and bulk RNA-seq analysis.
- scBERT: scBERT is a pre-training-based single-cell LLM focusing on cell-type prediction. It is based on Performer [97] with gene embeddings initialized by Gene2vec [98]. It has six self-attention blocks. The default fine-tuning process of scBERT on downstream datasets is freezing the penultimate layer. scBERT is considered for the Cell-type Annotation task.
- Geneformer: Geneformer is a single-cell LLM using transfer learning to predict cell types and gene functions. The Geneformer tokenization step is done based on ranking gene expression values in single cells following scaling across the whole training dataset. Cells are represented as token strings with genes rankings as tokens.
  Geneformer is used for the Cell-type Annotation task and the Gene Function Prediction task. Both tasks were performed by fine-tuning on the basis of the published

pre-trained Geneformer network. The default hyper-parameters were used for Geneformer fine-tuning. Prior to tokenization, gene names in all datasets were converted to ENSEMBL IDs using python packages *mygene* [99] and *pyensembl* [100], and most of unmatched genes are non-coding RNAs.
For consistency with other benchmarked LLMs, Geneformer's workflow in Gene Function Prediction and non-cross Cell-type Annotation was altered to split all labeled gene into the training and the testing datasets using the same manners mentioned in our Methods section, instead of using the cross-validation evaluation method provided by the authors.

- CellLM: CellLM is a single-cell LLM using three different pre-training strategies. The pre-training loss function includes: 1. masked gene expression level reconstruction; 2. cell condition discrimination; and 3. self-supervised contrastive learning. Moreover, they incorporated protein-protein interaction networks as prior information during the pre-training process. The downstream tasks of CellLM are all related to cell-type annotation.
  Here we modified CellLM so that it can perform cross-dataset cell-type annotation, by separating the training and the testing datasets.

Task-specific methods include:

- ResPAN: ResPAN is a batch effect correction tool based on Generative Adversarial Network (GAN) [101, 102]. The high-level idea of ResPAN is based on the distribution alignment or domain adaption across data from different batches. Such requirement can be treated as optimal transport, which can be accomplished by training a GAN. ResPAN is used for the batch effect correction task and the Multi-omics Data Integration task.
- scVI: scVI is a batch effect correction tool based on variational inference and variational auto-encoder [103]. scVI encodes the gene expression data with batch information using a neural network and set the output of the network as parameters for a distribution. Based on such distribution of the latent space, scVI can correct the batch effect in the latent space as well as the original space, as long as we consider the output of the decoder model.
- *Vanilla* NNs: This neural network contains three MLP layers with batch normalization and uses Mish [104] as the activation function. *Vanilla* NNs are used for the Cell-type Annotation task and the Gene Function Prediction task.
- TOSICA: TOSICA is a deep learning-based method for one-stop cell type annotation. TOSICA is designed with the self-attention multi-heads transformer without pre-training. It also provides interpretation for researchers about the attention embeddings and uses attention embeddings to perform biological analysis.
- GEARS: GEARS is a tool for single and multi-gene perturbation prediction based on single-cell RNA sequencing datasets. It combines gene-gene interaction network as prior information and uses a cross-gene neural network with a graph neural network to predict gene expression after perturbation.
- Tangram: Tangram is a toolbox for spatial transcriptomic data analysis based on neural networks. The key idea behind Tangram is using neural networks to find a good mapping function from single-cell data space to spatial data space. After

the mapping process, by integrating the information from the single-cell level and spatial level, it can perform several downstream tasks, including data imputation, cell-type deconvolution and others.

- scDesign3: scDesign3 is a model based on Copula distribution [105] to generate different single-cell datasets. Such datasets can be multimodal. Moreover, based on the input parameters and requirements of scDesign3, it can also generate datasets with specific conditions, including batch effect, cell conditions, and the stages of cell differentiation. The data generation of scDesign3 is based on real datasets.

E Model Architecture

The model architecture of scGPT, scBERT, CellLM, Geneformer and tGPT are provided in the Supplementary file 3.

F Data availability

No novel sequencing data were generated for this current study. Supplementary file 4 provides the sources and download links for the datasets used in each task. These datasets come from the following papers:

Batch Effect Correction: [35, 46, 106–109].
Cell-type Annotation: [23, 35, 46, 106, 110, 111].
Gene Function Prediction: [112].
Multi-omics Data Integration: [113].
Perturbation Prediction: [42].
Imputation: [55].
Gene Network Analysis: [46, 106].
Simulation Analysis: [106].
Emergent Ability: [106, 110].

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] 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

[20] 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)

[21] 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)

[22] 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)

[23] 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)

[24] 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)

[25] 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)

[26] 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)

[27] 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)

[28] 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)

[29] 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)

[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] 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)

[35] 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)

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

[37] 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)

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

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

[40] 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)

[41] 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

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

[43] 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)

[44] 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)

[45] 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)

[46] 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)

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

[48] 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)

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

[50] 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)

[51] 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)

[52] 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)

[53] 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)

[54] 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)

[55] 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)

[56] 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)

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

[58] 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)

[59] 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)

[60] 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

[61] 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)

[62] 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)

[63] 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)

[64] 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)

[65] 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)

[66] 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)

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

[68] 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)

[69] 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)

[70] 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)

[71] 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

[72] 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)

[73] 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

[74] 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

[75] 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

[76] 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

[77] 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)

[78] 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)

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

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

[81] 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)

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

[83] 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)

[84] 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)

[85] 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)

[86] 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)

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

[88] 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)

[89] 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)

[90] 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)

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

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

[93] 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)

[94] 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

[95] 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)

[96] 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)

[97] 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

[98] 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)

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

[100] 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)

[101] 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

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

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

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

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

[106] 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)

[107] 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)

[108] 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)

[109] 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)

[110] 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)

[111] 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)

[112] 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)

[113] 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)