

# scGPT: Towards Building a Foundation Model for Single-Cell Multi-omics Using Generative AI

Haotian Cui<sup>1,2,3\*</sup>, Chloe Wang<sup>1,2,3\*</sup>, Hassaan Maan<sup>1,3,4</sup>, Bo Wang<sup>1,2,3,4,5†</sup>

<sup>1</sup>*Peter Munk Cardiac Centre, University Health Network, Toronto, ON, Canada*

<sup>2</sup>*Department of Computer Science, University of Toronto, Toronto, ON, Canada*

<sup>3</sup>*Vector Institute, Toronto, ON, Canada*

<sup>4</sup>*Department of Medical Biophysics, University of Toronto, Toronto, ON, Canada*

<sup>5</sup>*Department of Laboratory Medicine and Pathobiology, University of Toronto, Toronto, ON, Canada*

## Abstract

Generative pre-trained models have achieved remarkable success in various domains such as natural language processing and computer vision. Specifically, the combination of large-scale diverse datasets and pre-trained transformers has emerged as a promising approach for developing foundation models. While texts are made up of words, cells can be characterized by genes. This analogy inspires us to explore the potential of foundation models for cell and gene biology. By leveraging the exponentially growing single-cell sequencing data, we present the first attempt to construct a single-cell foundation model through generative pre-training on over 10 million cells. We demonstrate that the generative pre-trained transformer, scGPT, effectively captures meaningful biological insights into genes and cells. Furthermore, the model can be readily finetuned to achieve state-of-the-art performance across a variety of downstream tasks, including multi-batch integration, multi-omic integration, cell-type annotation, genetic perturbation prediction, and gene network inference. The scGPT codebase is publicly available at <https://github.com/bowang-lab/scGPT>.

## 1 Main

Generative pre-trained models have recently achieved unprecedented success in many domains. The most well-known applications include computer vision and natural language generation (NLG) [44, 43, 45]. These foundation models such as DALL-E2 and GPT-4 follow a similar paradigm of pre-training transformers on large-scale diverse datasets [43, 45]. These foundation models can be readily tailored to a variety of downstream tasks and scenarios. More interestingly, they demonstrate improved performance on multiple tasks compared to task-specific models trained from scratch [22, 58, 47]. This showcases strong evidence of a task-agnostic and “deep” understanding

---

\*These authors contributed equally.

†Corresponding author. Email: [bowang@vectorinstitute.ai](mailto:bowang@vectorinstitute.ai)

31 of knowledge in these domains. Despite the success of foundation models in other domains, cur-  
32 rently machine-learning based discovery in single-cell research is rather distributed, with specific  
33 models dedicated to specific analysis tasks [32, 42, 27]. Often the breadth and scale of the datasets  
34 used in each study are also limited, due to sequencing capacity as well as the scope of the research  
35 question [2]. This calls for a foundation model pre-trained on large-scale data to achieve a general  
36 understanding of single-cell biology. We expect such a model to serve as a strong foundation and  
37 contribute to the discovery of new biological insights with the help of the learned knowledge from  
38 millions of sequenced cells.

39 While the feasibility of generative pre-training in single-cell biology remains largely unexplored,  
40 we can draw inspirations about modelling and the data-centric perspectives from other domains.  
41 From a modelling perspective, the self-attention transformer has been verified as an efficient and  
42 effective architecture to model input tokens of words. While texts are made up of words, cells can  
43 be characterized by genes. We can learn and extract gene and cell representations simultaneously  
44 in a similar fashion as word and sentence embeddings in NLG. The flexible vocabulary structure  
45 also allows easy addition of new features and meta information. The attention mechanism of the  
46 transformers could be further explored to inform on gene-to-gene and cross-modality associations  
47 [60]. From a data perspective, the vast-scale atlases of single-cell RNA sequencing (scRNA-seq),  
48 such as the Human Cell Atlas, already encompass tens of millions of cells, and the scale of accessible  
49 omic data continues to grow exponentially [26, 51, 23, 2]. This opens ample opportunities to employ  
50 self-supervised learning techniques to learn from diverse cell types and tissues, and integrate across  
51 different organs and species.

52 In this work, we present the first attempt to build a single-cell foundation model, scGPT, by  
53 generative pre-training on over 10 million cells. We introduce several new techniques to address  
54 the methodology and engineering challenges of pre-training on large-scale single-cell omic data.  
55 To handle the large-scale data, we use an in-memory data structure to store hundreds of datasets  
56 that allow fast access. We establish a unified generative pre-training workflow specifically for the  
57 non-sequential omic data, and adapt the transformer architecture to simultaneously learn cell and  
58 gene representations. We also provide reusable finetuning pipelines and objectives designed for a  
59 variety of downstream tasks to help users apply the pre-trained model with ease.

60 We demonstrate that the pre-trained model captures meaningful biological insights on both gene  
61 and cell levels. The learned gene embedding maps decode known pathways by grouping together  
62 genes that are functionally relevant. With zero-shot learning, the pre-trained model is able to  
63 reveal meaningful cell clusters on unseen datasets. With finetuning in a few-shot learning setting,  
64 the model achieves state-of-the-art performance on a wide range of downstream tasks, including  
65 batch correction, multi-omic integration, cell type annotation, genetic perturbation prediction,  
66 pseudo-cell generation, and gene network inference. The release of the scGPT model and workflow  
67 aims to facilitate future research in all related areas. We envision that the adoption of pre-trained  
68 foundation models will further our understanding of cellular biology, and pave the foundation for  
69 future discoveries.

## 70 2 Results

### 71 2.1 Single-cell transformer foundation model overview

72 Single-cell sequencing captures the genetic profiles at the individual cell level. For instance, scRNA-  
73 seq measures transcriptomic activities from RNA abundance, which informs on cell identity, stage,  
74 and functionality. Recent cellular reference maps such as the human cell atlas comprise of millions

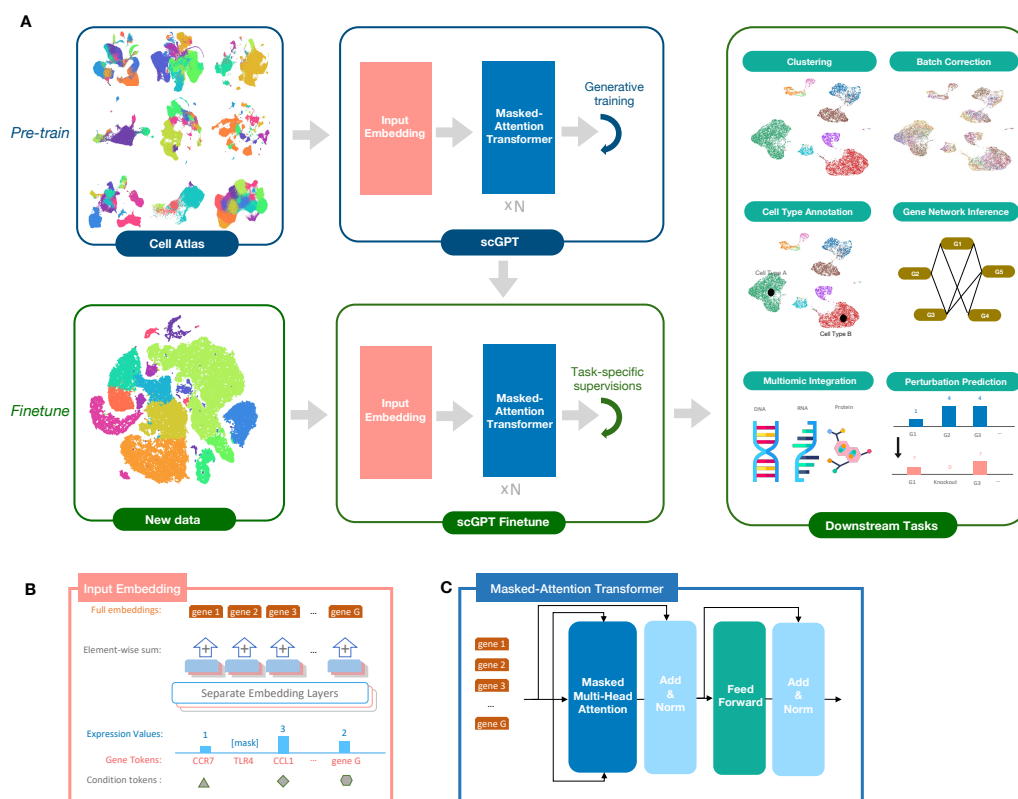


Figure 1: Model Schematic. (a) The workflow of scGPT. The model is firstly generatively trained on large-scale scRNA-seq data from cell atlas. For a downstream application, the pre-trained model weights can be finetuned on the new data. The core component of scGPT contains stacked transformer blocks with specialized attention masks for generative training. We applied scGPT on a variety of tasks including clustering, batch correction, cell type annotation, multi-omics, genetic perturbation prediction, and gene network inference. (b) The zoom-in view of the input embeddings. The input contains three layers of information, the gene token, expression value, and condition tokens (modality, batch, perturbation condition, et al.). (c) The zoom-in view of the scGPT transformer layer. We introduced a specially designed attention mask to conduct generative pre-training on single-cell sequencing data.

75 of single cells from diverse organs and tissues, offering an unparalleled representation of cellular  
76 heterogeneity [26, 52]. We introduce scGPT as the generative pre-trained foundation model in the  
77 single-cell domain. The core model has stacked transformer layers with multi-head attention that  
78 learns cell and gene embeddings simultaneously (See Online Methods 4.2). Figure 1A illustrates a  
79 two-stage workflow involving pre-training and fine-tuning of scGPT. In the pre-training stage, we  
80 collected over 10.3 million scRNA-seq data of blood and bone marrow cells from the CellXGene  
81 portal [11] for training. We introduce a specially designed attention mask and generative training  
82 pipeline to train scGPT in a self-supervised manner to jointly optimize cell and gene representations  
83 (See Online Methods 4.3). During training, the model gradually learns to generate gene expression  
84 of cells based on simple cell or gene expression cues. In the fine-tuning stage, researchers can  
85 apply the pre-trained model to new datasets and specific tasks (See online Methods 4.5). We  
86 offer flexible finetuning pipelines suitable for a variety of downstream tasks essential in single-cell  
87 research, including scRNA-seq integration with batch correction, cell type annotation, multi-omic  
88 integration, perturbation prediction, and gene regulatory network inference.

89 scGPT learns cell and gene representations from diverse single-cell data through gene expres-  
90 sion modelling. To facilitate gene representation learning, we employed Gene Expression Prediction  
91 (GEP) as the generative self-supervised objective to iteratively predict gene expression values of  
92 unknown tokens from known tokens in an auto-regressive manner (See Online Methods 4.4). To  
93 enhance cell representation learning, we designed Gene Expression Prediction for Cell Modelling  
94 (GEPc) objective, where the model predicts gene expression values from cell representations (See  
95 Online Methods 4.4). This creates a direct link between the gene expression profile and cellular  
96 heterogeneity, allowing for joint optimization within the scGPT framework. Furthermore, scGPT's  
97 embedding architecture can easily extend to multiple sequencing modalities, batches, and pertur-  
98 bation states. This is achieved by adding new condition tokens for sequencing modalities, batches,  
99 and perturbation states. This unique model flexibility enables the pre-trained model to seam-  
100 lessly combine with any additional information required for specific downstream tasks. See model  
101 architecture illustration in Figure 1B and more details in Online Methods 4.

102 scGPT serves as a powerful single-cell feature extractor on previously unseen datasets. In  
103 benchmark experiments, scGPT outperformed recent methods and achieved state-of-the-art results  
104 across all downstream tasks. This demonstrates the benefits of pre-training and the transferability  
105 of learned knowledge across diverse use cases. By providing a robust and unified framework, scGPT  
106 enables single-cell researchers to easily leverage the pre-trained foundation model in related studies.

## 107 **2.2 Integration of multiple scRNA-seq data with batch correction**

108 Clustering and visualization of single-cell sequencing data encounter a significant challenge in the  
109 presence of batch effects arising from the utilization of multiple datasets or sequencing batches as  
110 input. By employing a finetuning workflow, the scGPT framework effectively tackles this challenge  
111 by incorporating customized finetuning objectives (refer to Online Methods 4.4). This approach  
112 successfully corrects for batch effects while preserving the true biological signals inherent in the  
113 data.

114 scGPT achieves the state-of-the-art performance in preserving the biological variance of the  
115 integrated datasets upon batch correction. We benchmarked scGPT with three popular integration  
116 methods, scVI [34], Seurat [55], and Harmony [29] on two integration datasets Immune Human  
117 (10 batches) [36] and PBMC 10K (2 batches) [21]. As shown in Figure 2A in the Immune Human  
118 dataset, scGPT successfully integrated all batches of CD4+ T cells, CD8+ T cells, and CD14+  
119 Monocytes into their individual clusters, whereas Seurat produced a few sub-clusters corresponding  
120 to sequencing batches within each of these cell types (See batch visualizations in Supplementary

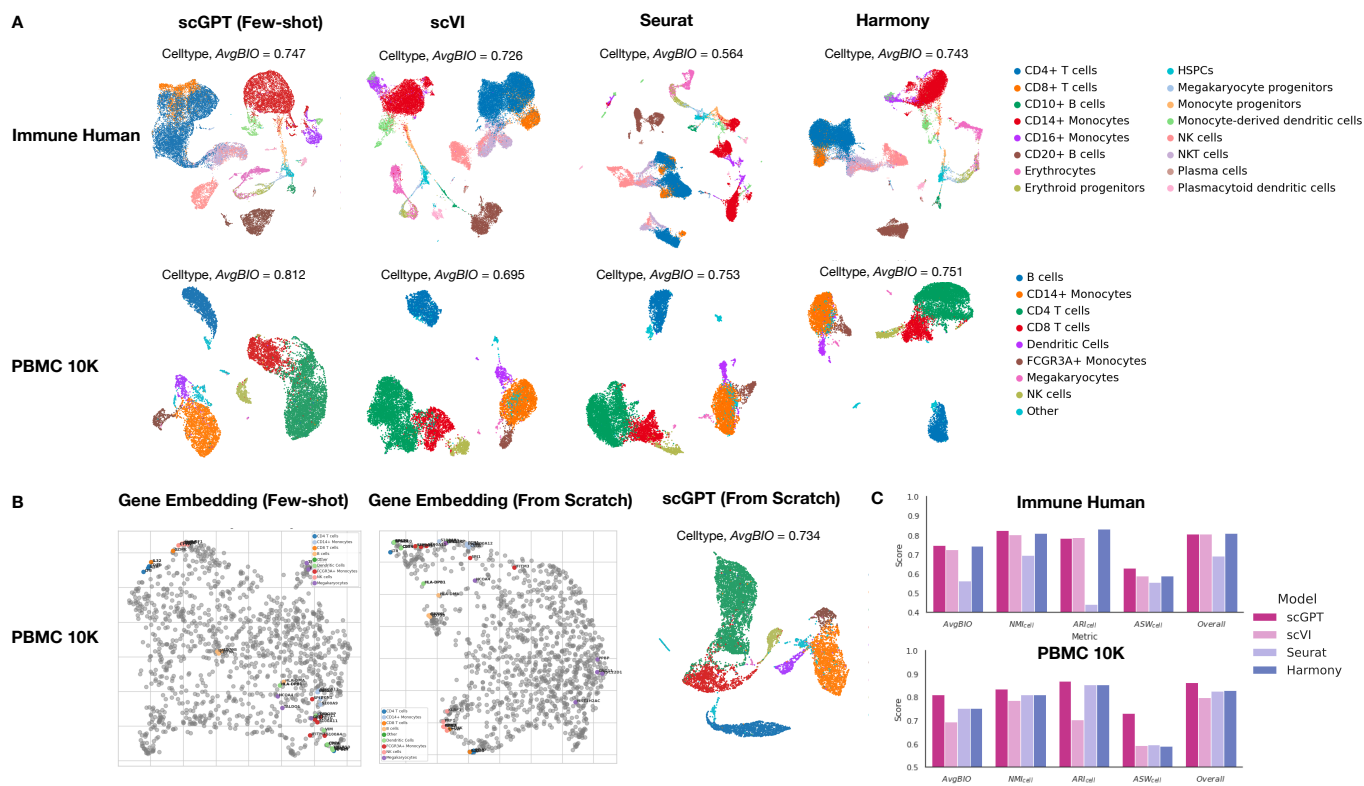


Figure 2: (a) Benchmark of the few-shot scGPT model with scVI [34], Seurat Seurat [55], and Harmony Harmony [29] on the Immune Human (10 batches) [36] and PBMC 10K (2 batches) [21] datasets for cell type clustering performance upon batch integration. The UMAP plot of learned cell embeddings was colored by *cell types*. (b) Comparison of gene embedding maps of the few-shot and trained-from-scratch scGPT models. The highly variable genes from each *cell type* were highlighted. The UMAP plot of learned cell embeddings from trained-from-scratch scGPT model was visualized. (c) Comparison of the scGPT model with other benchmarked methods on *AvgBIO*, the detailed biological conservation metrics ( $NMI_{cell}$ ,  $ARI_{cell}$ ,  $ASW_{cell}$ ), and the *Overall* score.

121 Figure S3). scGPT also managed to separate the Monocyte-derived dendritic cells from the CD16+  
 122 Monocytes, but scVI and Harmony both saw a significant overlap of the two clusters. Moreover,  
 123 in the PBMC 10K dataset, scGPT is the only method that clearly separated out the cell type  
 124 Other from the annotated clusters. In contrast, scVI, Seurat and Harmony all confused this  
 125 Other cell type with CD14+ Monocytes and CD8 T cells. This inaccuracy is visualized as these  
 126 blue dots from cell type Other scattering all over the orange and red clusters. scGPT's superior  
 127 clustering performance is also reflected in the biological conservation score, where scGPT achieves  
 128 an *AvgBIO* score of 0.812, which is 5% higher than Seurat and Harmony, and 10% higher than  
 129 another deep learning method, scVI. In Figure 2C, scGPT presents competitive scores across all  
 130 cell type clustering metrics attributing to biological conservation. scGPT also ranked top in the  
 131 Overall metric considering both biological conservation and batch correction performance (See  
 132 detailed metrics in Supplementary Table S.1, and batch visualizations in Supplementary Figure  
 133 S3).

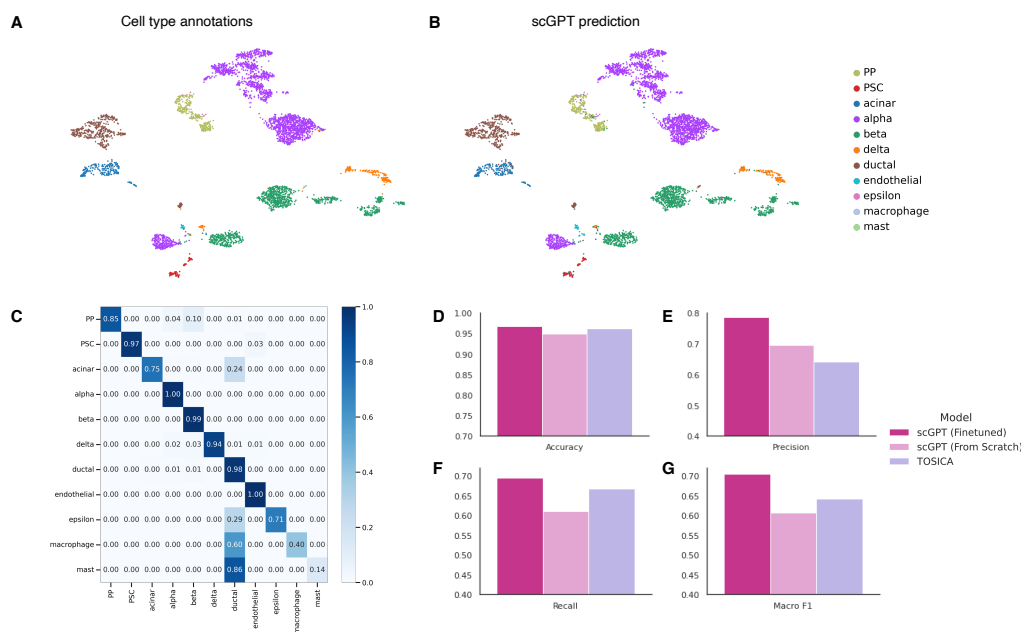


Figure 3: (a) UMAP visualization of cells colored by ground-truth *cell types* in the hPancreas query set. (b) UMAP visualization of cells colored by scGPT predicted *cell types*. (c) Normalized confusion matrix by *cell types*. (d) – (g) Comparison of the finetuned scGPT model with trained-from-scratch model and existing method on *Accuracy*, *Precision*, *Recall* and *MacroF1* scores.

134 We further highlight the benefits of pre-training by showing a significant performance boost in  
 135 the finetuned model in comparison to the trained-from-scratch model in the PBMC 10K dataset  
 136 (See Figure 2B). The finetuned gene embeddings produced more compact networks for highly vari-  
 137 able genes of the CD4 T cells and Megakaryocytes, compared to the trained-from-scratch model.  
 138 We observe similar results in the cell embeddings, as the cell type clusters become more defined in  
 139 the finetuned model with a 8% improvement in the *AvgBIO* score. As a sanity check, we validated  
 140 that the pre-trained model in zero-shot is also able to produce meaningful cell type clusters with an  
 141 *AvgBIO* score of 0.728, on par with the trained-from-scratch model (See Supplementary Figure  
 142 S2). This presents the zero-shot model as a generalizable feature extractor that can be readily  
 143 applied to unseen datasets. Furthermore, the 8% performance boost from pre-training demon-  
 144 strates the benefits of leveraging these pre-trained feature extractors and the knowledge learned.  
 145 The foundation model is proven to be not only easily transferable to new datasets but also more  
 146 powerful than learning from limited data from scratch.

## 147 2.3 Cell type annotation

148 Cell type annotation is a crucial step in single-cell analysis after clustering, as it resolves het-  
 149 erogeneity in sequenced tissues and lays the foundation for further investigation of cell and gene  
 150 functions to gain biological and pathological insights. While several methods have been proposed  
 151 for cell annotation, such as cellAssign [64], singleR [3], and Chetah [17], they typically require di-  
 152 mension reduction prior to model input, which can lead to information loss. In contrast, scGPT’s  
 153 transformer model can directly take in gene expressions in an unbiased manner, with full resolu-  
 154 tion on the entire highly variable gene set as input. This approach provides greater reliability and  
 155 improved accuracy in cell type classification, as demonstrated in our subsequent analyses.

156 For the cell type annotation task, we finetuned the pre-trained scGPT model using cross-  
157 entropy loss against ground-truth labels from a new reference dataset (See Online Methods 4).  
158 Using the hPancreas dataset of human pancreas cells as an example, we trained the scGPT model  
159 on the reference set and validated the classification performance on a different query set. Figure 3  
160 panels A and B present the cell embeddings colored by ground-truth versus predicted cell types,  
161 where the scGPT model demonstrates faithful prediction with high accuracy score of 96.7%. The  
162 model also achieved high precision in predicting majority of the cell types, except for the rare cell  
163 types with extremely low cell numbers in the reference set (See Figure 3C). For example, fewer  
164 than 50 cells belong to the mast and macrophage cell types out of the 10.6 thousand cells in the  
165 reference set. To benchmark the performance of scGPT, we compared it with TOSICA [12], a  
166 recent transformer-based annotation method. scGPT outperforms TOSICA on all classification  
167 metrics, including *Accuracy*, *Precision*, *Recall*, and *MacroF1*, as shown in Figure 3D-G. We  
168 also trained a separate scGPT model from scratch without the pre-trained model parameters. It  
169 achieves reasonable accuracy on the query set. The performance improvement of the finetuned  
170 model from the trained-from-scratch model demonstrates the benefits of transfer learning with  
171 pre-trained scGPT.

## 172 2.4 Perturbation Prediction

173 Sequencing and gene editing techniques have recently facilitated high-throughput experiments, en-  
174 abling the exploration of cellular responses to multiple genetic perturbations. The approach holds  
175 immense promise for uncovering novel gene interactions and advancing regenerative medicine. How-  
176 ever, the vast combinatorial space of potential gene perturbations quickly surpasses the practical  
177 limits of experimental feasibility. To overcome this limitation, scGPT can be employed to leverage  
178 the knowledge gained from cellular responses in known experiments and extrapolate to predict  
179 responses in unknown scenarios. The utilization of self-attention mechanisms over the gene dimen-  
180 sion enables the encoding of intricate interactions between perturbed genes and the responses of  
181 other genes. By leveraging this capability, scGPT can effectively learn from existing experimental  
182 data and accurately predict gene expressions following perturbation.

183 For the perturbation prediction task, we evaluated our model using two perturbation datasets  
184 pre-processed by Roohani, Huang, and Leskovec [53]: (1) the Pertub-seq dataset of K562 leukemia  
185 cell line [1], which comprises 87 one-gene perturbations, with approximately 100 cells per pertur-  
186 bation and a minimum of 7,000 unperturbed cells, and (2) the other Norman Perturb-Seq dataset  
187 [41], consisting of 131 two-gene perturbations and 105 one-gene perturbations.

188 We assessed the perturbation prediction by calculating the Pearson correlation ( $corr$ ) between  
189 the predicted and the corresponding ground-truth expression values after perturbation. In addition,  
190 we introduced a variant of the Pearson metric, denoted as  $corr(\Delta)$ , which measures the correlation  
191 based on the magnitude of expression change post-perturbation compared to the control. We have  
192 presented Pearson metrics for various gene sets, namely all genes ( $ALL$ ) and the top 20 differentially  
193 expressed genes ( $DE$ ). For detailed information on how these metrics were calculated, refer to  
194 Supplementary Online Methods S.2.

Table 1: Results of perturbation prediction

Model	Norman et al. [41]				Adamson et al. [1]			
	DE		ALL		DE		ALL	
	corr	corr( $\Delta$ )	corr	corr( $\Delta$ )	corr	corr( $\Delta$ )	corr	corr( $\Delta$ )
MLP	0.909	0.428	0.987	0.408	0.948	0.729	0.991	<b>0.656</b>
GEARS	0.917	0.508	0.986	0.387	0.961	0.726	0.991	0.652
scGPT	<b>0.923</b>	<b>0.546</b>	<b>0.988</b>	<b>0.459</b>	<b>0.971</b>	<b>0.775</b>	<b>0.992</b>	0.647

195 We conducted a performance comparison between scGPT, the recent GEARS method [53], and  
196 the multi-layer perception (MLP) baseline. Our results demonstrate that scGPT achieves the high-  
197 est correlation across seven out of eight evaluation metrics. It is worth noting that approximately  
198 50% of the gene expression counts are zero in the raw scRNA-seq data. Therefore, we contend that  
199 evaluating differentially expressed genes, specifically the DE columns in Table 1, provides more  
200 compelling evidence. Notably, scGPT exhibits significant improvements in the correlation of the  
201 ( $\Delta$ ) change of the top 20 differentially expressed genes, which is arguably the pivotal metric.

## 202 2.5 scGPT facilitates multi-omic integration and multi-modal represen- 203 tation learning

204 Single-cell multi-omic (scMultiomic) data presents multiple views of genetic regulation all at once,  
205 including epigenetic, transcriptomic, and translation activities [57, 37]. It provides an ample oppor-  
206 tunity to enhance feature and cell representation learning beyond gene expressions. However, the  
207 challenge lies in how to reliably aggregate cell representations from multiple views while preserving  
208 biological signals.

209 The scGPT framework can be readily extended to integrate multiple sequencing data modal-  
210 ities. Each omic type in scMulti-omic data (e.g., gene expression, chromatin accessibility, and  
211 protein abundance) is similar to a different language in NLG. Analogously, scGPT supports joint  
212 optimization of multi-omic tokens from diverse sequencing modalities. This framework also allows  
213 seamless addition of new sequencing modalities to existing pre-trained network by extending the  
214 “vocabulary”. In the benchmark experiments, scGPT demonstrates outstanding performance in  
215 cell representation learning and multiomic batch integration tasks compared to existing state-of-  
216 the-art methods (See Figure 4).

217 scGPT effectively extracts integrated cell embeddings from paired scMultiomic data. In this  
218 paired data integration setting, each sequenced cell contains all the data modalities. We used the  
219 10X Multiome PBMC [14] dataset with joint gene expression and chromatin accessibility mea-  
220 surements as an example. We benchmarked scGPT with two state-of-the-art methods scGLUE[9]  
221 and Seurat v4[24] on cell type clustering performance. As shown in Figure 4, scGPT is the only  
222 method that produced a clear separate cluster for CD8 Naive cells, whereas the other two methods  
223 failed. scGPT also differentiated Memory B cells from Naive B and Intermediate B cell clusters,  
224 yet scGLUE produced a merged cluster of all three types of B cells. scGPT separated the CD4 and  
225 CD8 cell groups into two distinct groups of clusters, surpassing the results of Seurat v4. scGPT  
226 demonstrates superior cell type clustering performance overall ( $AvgBIO=0.767$ ) and robustness  
227 across the diverse biological conservation metrics benchmarked (See Figure 4 and Supplementary  
228 Table S.1).

229 scGPT simultaneously integrates multi-modal batches from mosaic scMultiomic data via joint  
230 representation learning. In the mosaic data integration setting, sequenced samples share a few  
231 but not all data modalities. We used the ASAP human PBMC dataset [38] with four sequencing  
232 batches and three data modalities as an example. The first two sequencing batches contain gene  
233 expression and protein abundance data from CITE-seq, and the second two batches have chromatin  
234 accessibility and protein abundance measurements from ASAP-seq. In the benchmark experiment  
235 with scMoMat[65], scGPT demonstrates superior batch correction performance as shown in Figure  
236 4, especially in the rarer cell group NK cell. In comparison, scMoMat produced two distinct  
237 clusters for each cell type corresponding to the first two and second two batches, indicating failure  
238 to mitigate modality differences. scGPT achieves a overall batch correction score  $AvgBATCH$   
239 of 0.948, with a close-to-perfect  $GraphConn$  score of 0.992 and a significantly higher  $ASW_{batch}$   
240 score of 0.904 compared to scMoMat ( $ASW_{batch} = 0.849$ ). scGPT’s biological conservation metrics



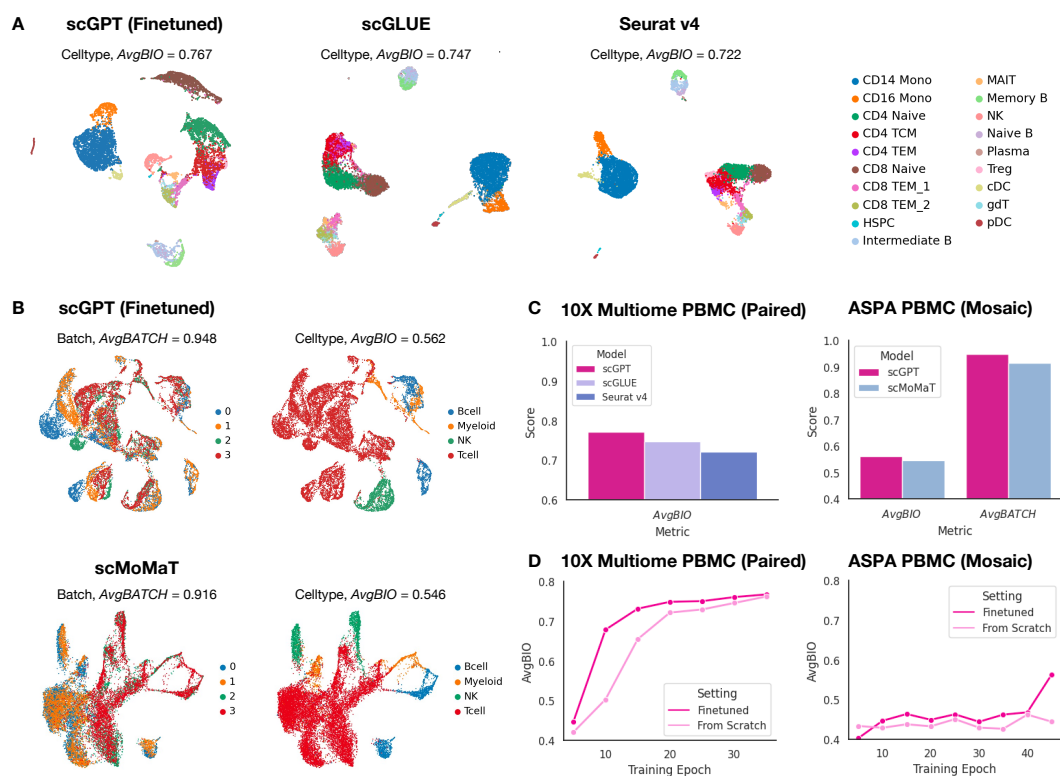


Figure 4: (a) Benchmark of the few-shot scGPT model with scGLUE[9] and Seurat v4[24] on the 10x Multiome PBMC [14] dataset (paired data setting) for cell type clustering task. The UMAP plot of learned cell embeddings was colored by *cell types*. (b) Benchmark of scGPT with scMoMat[65] on the ASAP PBMC [38] dataset (mosaic data setting) for batch correction and cell type clustering tasks. The UMAP plots of learned gene embeddings were colored by *sequencing batches* (left) and *cell types* (right). (c) Comparison of the scGPT model with other benchmarked methods on *AvgBIO* and *AvgBATCH* scores. (d) Comparison of training progress curves of the few-shot and trained-from-scratch models with *AvgBIO* scores evaluated every five epochs.

241 also compare favorably to scMoMat’s, which further indicates the robustness of multi-modal batch  
 242 correction without interfering with the biological signals (See Figure 4 and Supplementary Table  
 243 S.1).

244 scGPT readily supports the addition of new data modalities to existing pre-trained models.  
 245 We compared scGPT’s training progress curves in two settings, finetuned on the pre-trained model  
 246 and trained from-scratch, to demonstrate the benefits of pre-training in the multi-omic integration  
 247 task. As shown in Figure 4, for the 10X Multiome PBMC dataset, the finetuned model reached  
 248 the best AvgBIO scores in the 70% range 5 epochs earlier than the trained-from-scratch model.  
 249 This demonstrates the benefits of the pre-trained model in leading training progress and faster  
 250 convergence. In the more challenging mosaic integration setting with the ASAP PBMC dataset,  
 251 the finetuned model’s AvgBIO scores increased steadily as training proceeded, whereas the trained-  
 252 from-scratch model did not see much progress with training. At epoch 45, the pre-trained model  
 253 finished at an AvgBIO score of 0.562, which is more than 10% higher than the trained-from-scratch  
 254 model with a score of 0.444. This suggests that the model leverages the learned gene embeddings  
 255 from large atlases to guide the learning of peak and protein embeddings in the pre-trained setting.

256

## 2.6 Gene embeddings for Gene Regulatory Network inference

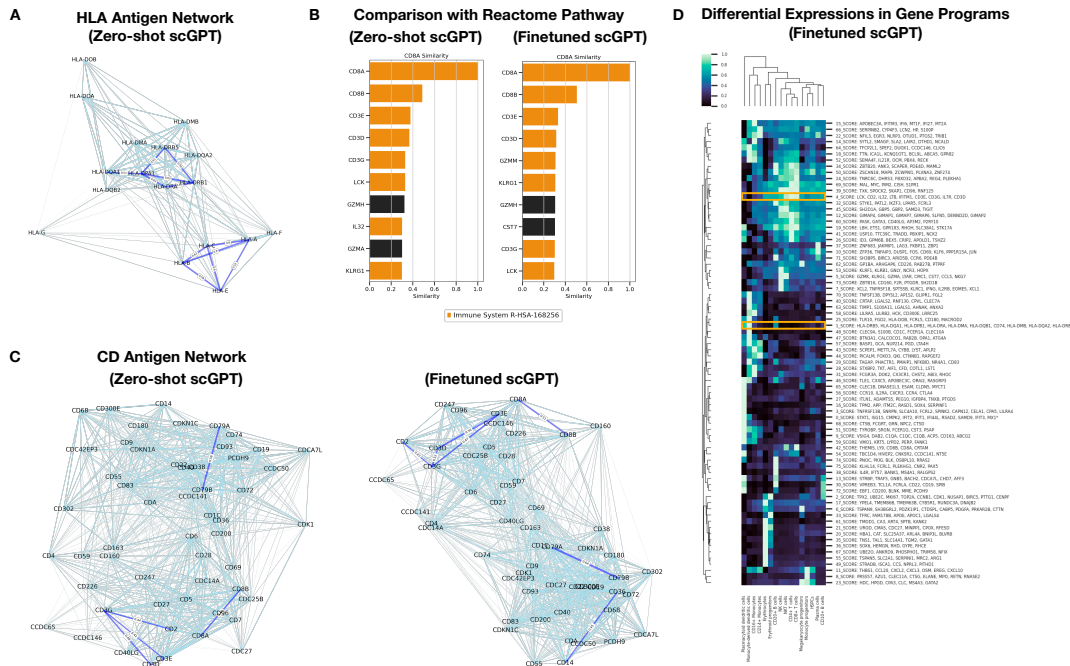


Figure 5: (a) HLA antigen network from zero-shot scGPT. (b) CD8A gene neighbors from zero-shot and finetuned scGPT models, ranked by embedding similarity colored by ground-truth signalling pathway from Reactome. (c) CD antigen network from zero-shot and finetuned scGPT on the Immune Human dataset. (d) Differential expressions among scGPT-extracted gene programs by cell types in the Immune Human dataset.

257 The interactivity between transcription factors, cofactors and target genes underlying a Gene  
 258 Regulatory Network (GRN) mediates important biological processes. Existing GRN inference  
 259 methods often rely on correlation in static gene expressions or pseudo-time estimates as a proxy  
 260 for causal graphs [46]. scGPT, optimized by the generative training of gene tokens, implicitly  
 261 encodes such relationships in its gene embeddings. The gene embeddings can therefore be applied  
 262 to construct similarity networks that entail gene-gene interactions. We hereby validate the scGPT's  
 263 gene embedding network against known biology, and then explore its applicability to gene program  
 264 discovery.

265 scGPT demonstrates its ability to group the functionally related genes and differentiate func-  
 266 tionally different genes from its gene embedding network. In Figure 5A, we visualized the similarity  
 267 network of the human leukocyte antigens (HLA) antigens from the pre-trained gene embeddings.  
 268 In the zero-shot setting, the scGPT model highlights two clusters corresponding to the two well-  
 269 characterized HLA classes that trigger different immune responses, namely HLA class I and HLA  
 270 class II. The HLA class I antigens HLA-A, -C, and -E are recognized by CD8+ T cells to mediate  
 271 cell killing, whereas HLA class II antigens HLA-DR, -DP, and -DQ are recognized by CD4+T  
 272 cells to trigger broader helper functions [13]. For the finetuned scGPT model on the Immune  
 273 Human dataset, we explored the CD antigen network specific to the immune cell types present in  
 274 this dataset (See Figure 5C). The pre-trained scGPT is able to identify CD3E, D, and G genes  
 275 as a group that encodes the T3 complex for T-cell activation, CD79A and B for B-cell signalling,  
 276 and CD8A and B as co-receptors for HLA class 1 molecules [40]. The finetuned scGPT further

277 highlights the connection between CD36 and CD14 as markers for monocytes and macrophages.  
278 This demonstrates scGPT’s ability to generalize from the knowledge learned in pre-training and  
279 extract specific information related to the dataset at hand.

280 scGPT reconstructs meaningful gene programs in a purely unsupervised workflow. In Figure 5D,  
281 we visualized the gene programs extracted by the finetuned scGPT model on the Immune Human  
282 dataset, and their differential expressions by cell types. These gene programs are selected in an  
283 unsupervised manner by first clustering the gene embeddings and then thresholding on clusters  
284 that consist of 5 or more genes, following Ceglia et al. [10]’s proposed pipelines (See Online Methods  
285 4). We observed that the same HLA antigen cluster was identified as group 1. Similarly, the CD3  
286 genes involved in T3 complex were identified as group 4, with highest expressions in T cells. This  
287 confirms that scGPT’s inferred gene programs correspond to function groups that are biologically  
288 meaningful. We further validated the gene similarity relationships encoded by the scGPT model  
289 against the known Reactome database [50]. Using the CD8A gene as an example, we demonstrate  
290 that its closest neighbors are more likely to be part of the Immune System R-HSA-168256 pathway  
291 than genes that are further away, from both zero-shot and finetuned scGPT models (See Figure  
292 5B). The top three genes closest to CD8A remain consistent before and after finetuning. The  
293 finetuned model highlighted one additional gene GZMM that is commonly enriched in NK, NKT  
294 and T cell subclusters [4] which are major cell types found in the Immune Human dataset. We  
295 further validated this correlation on the entire gene set across the Reactome database. Among  
296 pairs of genes, there exists a positive correlation between the cosine similarity score of the gene  
297 embeddings and the number of common pathways shared by these genes, with a Pearson correlation  
298 score of 0.316.

299 While a more comprehensive evaluation pipeline is to be established, these findings showcase  
300 that scGPT has learned meaningful biological patterns from generative pre-training in the zero-  
301 shot setting. More specifically, we demonstrate its ability to perform unsupervised gene program  
302 discovery on new datasets along with other cell-level analysis tasks by leveraging the pre-trained  
303 model. We envision this attempt as one of the first steps towards knowledge discovery in the  
304 single-cell domain assisted by foundation models.

### 305 **3 Discussion**

306 We hereby present scGPT, the first foundation model that leverages pre-trained transformers  
307 learned from over 10 million single-cell data. The self-supervised pre-training paradigm with  
308 increasingly large amount of training data has created powerful language models such as chatGPT  
309 and GPT4 [44, 43]. These successes inspired us to apply the same pre-training paradigm to  
310 the single-cell domain, with the aim of decoding complex biological interactions with the pre-  
311 trained transformers. The transformer models naturally support the joint learning of gene and  
312 cell embeddings, analogous to word and sentence embeddings in NLG. These technical advantages  
313 create a solid foundation for modelling these different aspects of cellular processes together at once.

314 We demonstrate the benefits of pre-training with comprehensive experiments in both zero-shot  
315 and finetuning settings. The pre-trained model itself is a universal feature extractor. It showcases  
316 strong capabilities of extrapolating to unseen datasets, presenting meaningful cell clustering in  
317 zero-shot experiments. The learned gene networks also reflect known gene programs and their  
318 functional roles. These abilities give us confidence in the pre-trained model that it has not only  
319 memorized but also synthesized the patterns from the large-scale single-cell data. We also observed  
320 a consistent contribution of the pre-trained model in multiple downstream tasks via transfer learn-  
321 ing. For example, in both multi-batch and multi-omic integration tasks, the finetuned model

322 has demonstrated superior performance in cell-type clustering, with an 8 to 12% increase in the  
323 biological conservation score compared to the trained-from-scratch models.

324 To implement generative pre-training for the non-sequential single-cell data, we introduced  
325 specialized attention masking to support generation and joint gene and cell representation learning.  
326 In the finetuning pipeline, we offer setups for a diverse range of downstream tasks such as batch  
327 correction, cell type assignment, multi-omic integration, perturbation prediction, and gene network  
328 inference. We hereby release the scGPT codebase and the pre-trained model. We hope that this  
329 provides a unified framework to help researchers easily adapt the pre-trained models to their own  
330 tasks at hand.

331 For future directions, we plan to pre-train on a larger-scale dataset with more diversity, includ-  
332 ing multi-omic data, spatial omics, and diseased conditions. It is also interesting to incorporate  
333 perturbation and temporal data in the pre-training stage for causal discovery. More importantly,  
334 we would like to validate the pre-trained model on a wider range of biologically meaningful tasks  
335 to understand and interpret what the pre-trained model has learned. We also aim to explore  
336 in-context instruction learning for single-cell data. The goal is to have a pre-trained model that  
337 understands different tasks and contexts in the zero-shot setting without having to finetune. scGPT  
338 thus serves as the first step to use large-scale pre-trained foundation model to understand the con-  
339 text and nuances in cell biology. We envision that the pre-training paradigm be readily integrated  
340 into single-cell research, and serve as a foundation to leverage the existing knowledge from the  
341 exponentially growing cell atlases for new discoveries.

## 342 4 Methods

### 343 4.1 Input embeddings

344 The single-cell sequencing data is processed into a cell-gene matrix,  $\mathbf{X} \in \mathbb{R}^{N \times G}$ , where each element  
345  $X_{i,j} \in \mathbb{R}^+$  represents the read count of a RNA for scRNA-seq or a peak region if scATAC-seq. For  
346 example in scRNA-seq, the element denotes the RNA abundance for gene  $j \in 0, 1, \dots, G$  in cell  
347  $i \in 0, 1, \dots, N$ . In subsequent sections, we will refer to this matrix as the raw matrix. The input to  
348 scGPT consists of three main components: (1) gene (or peak) tokens, (2) expression values, and (3)  
349 condition tokens. For each modeling task, the gene tokens and expression values are pre-processed  
350 from the raw count matrix  $\mathbf{X}$  accordingly:

351 **Gene Tokens** In the scGPT framework, each gene is considered the smallest unit of information,  
352 equivalent to a word in natural language generation (NLG). We therefore use gene names as tokens,  
353 and assign each gene  $g_j$  a unique integer identifier  $id(g_j)$  within the complete vocabulary of tokens.  
354 This approach offers great flexibility to harmonize multiple studies with different gene sets (i.e.,  
355 generated by distinct sequencing technologies or pre-processing pipelines). Specifically, different  
356 sets of gene tokens can be integrated into a common vocabulary by taking the union set of all genes  
357 across studies. Additionally, we incorporate special tokens in the vocabulary, such as  $\langle cls \rangle$  for  
358 aggregating all genes into a cell representation, and  $\langle pad \rangle$  for padding the input to a fixed  
359 length. Conceptually, we draw parallels between gene tokens and word tokens in NLG. The input  
360 gene tokens of each cell  $i$  are hence represented by a vector  $\mathbf{t}_g^{(i)} \in \mathbb{N}^M$ :

$$\mathbf{t}_g^{(i)} = [id(g_1^{(i)}), id(g_2^{(i)}), \dots, id(g_M^{(i)})], \quad (1)$$

361 where  $M$  is a pre-defined input length, and usually equals to the number of selected highly variable  
362 genes.

363 **Expression Values** The gene expression matrix  $X$  requires additional processing before being  
364 used as input for modeling. A fundamental challenge in gene expression modeling is the variability  
365 in absolute magnitudes across different sequencing protocols [54]. Due to variations in sequencing  
366 depths and the presence of sparsely expressed genes, the data scales differ significantly among  
367 different batches of sequencing samples. These differences are not easily mitigated with common  
368 preprocessing techniques such as transcripts per million (TPM) normalization and  $\log_1 p$  trans-  
369 formation [25]. To make it more clear, the same absolute value can convey different "semantic"  
370 meanings across sequencing batches. To address this scale difference, we propose the *value bin-*  
371 *ning* technique to convert all expression counts into relative values. For each non-zero expression  
372 count in each cell, we calculate the raw absolute values and divide them into  $B$  consecutive inter-  
373 vals  $[b_k, b_{k+1}]$ , where  $k \in \{1, 2, \dots, B\}$ . Each interval represents an equal portion of all expressed  
374 genes ( $1/B$ ). It is important to note that a new set of bin edges is computed for each cell, so the  
375 interval edges  $b_k$  may vary among cells. The binned expression value  $x_j^{(i)}$  for cell  $i$  is defined as:

$$x_j^{(i)} = \begin{cases} k, & \text{if } X_{i,j} > 0 \text{ and } X_{i,j} \in [b_k, b_{k+1}], \\ 0, & \text{if } X_{i,j} = 0. \end{cases} \quad (2)$$

376 Through this binning technique, the semantic meaning of  $x_j^{(i)}$  is consistent across sequencing  
377 batches. For instance, a value of  $x_j^{(i)} = B$  consistently indicates the highest expression among  
378 genes. Before applying the value binning step, we performed  $\log_1 p$  transformation, and highly  
379 variable gene selection [35]. To simplify the notation, we use  $X_{i,j}$  to represent both the raw and  
380 preprocessed data matrices prior to binning. Therefore, the final input vector of binned expression  
381 values for cell  $i$  is denoted as

$$\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \dots, x_M^{(i)}]. \quad (3)$$

382 **Condition Tokens** The condition tokens encompass diverse meta information associated with  
383 individual genes, such as functional pathways (represented by pathway tokens) or perturbation  
384 experiment alterations (indicated by perturbation tokens). To represent position-wise condition  
385 tokens, we utilize an input vector that shares the same dimension as the input genes. This vector  
386 is denoted as:

$$\mathbf{t}_c^{(i)} = [t_{c,1}^{(i)}, t_{c,2}^{(i)}, \dots, t_{c,M}^{(i)}], \quad (4)$$

387 where  $t_{c,j}^{(i)}$  represents an integer index corresponding to a condition.

388 **Embedding layers** We utilize the conventional embedding layers<sup>1</sup>  $\text{emb}_g$  and  $\text{emb}_c$  for the gene  
389 tokens and condition tokens, respectively, to facilitate the mapping of each token to a fixed-  
390 length embedding vector of dimension  $D$ . We employ fully connected layers, denoted as  $\text{emb}_x$ ,  
391 for the binned expression values to enhance expressivity. This choice enables the modeling of the  
392 continuum of gene expression values. Consequently, the final embedding  $h^{(i)} \in \mathbb{R}^{M \times D}$  for cell  $i$  is  
393 defined as,

---

<sup>1</sup>pytorch embedding layer

$$\mathbf{h}^{(i)} = \text{emb}_g(\mathbf{t}_g^{(i)}) + \text{emb}_x(\mathbf{x}^{(i)}) + \text{emb}_c(\mathbf{t}_c^{(i)}). \quad (5)$$

## 394 4.2 Cell and gene expression modeling by transformers

### 395 4.2.1 scGPT Transformer

396 We employ the self-attention transformer [60, 18] to encode the complete input embedding  $\mathbf{h}^{(i)}$   
397 in equation 5. The self-attention mechanism operates on the sequence of  $M$  embedding vectors,  
398 making it particularly suitable for capturing interactions between genes. The output of the stacked  
399 transformer blocks can be defined as follows:

$$\begin{aligned} \mathbf{h}_0^{(i)} &= \mathbf{h}^{(i)} \\ \mathbf{h}_l^{(i)} &= \text{transformer\_block}(\mathbf{h}_{l-1}^{(i)}) \quad \forall l \in [1, n] \end{aligned} \quad (6)$$

400 We utilize the resulting representation  $\mathbf{h}_n^{(i)} \in \mathbb{R}^{M,D}$  for both gene-level and cell-level tasks.  
401 Gene-level finetuning objectives (See Online Methods 4.4) are directly applied. Examples in-  
402 clude the gene expression prediction (GEP) objective and the perturbed expression prediction  
403 task (perturb-GEP). For cell-level tasks, we first integrate  $\mathbf{h}_n^{(i)}$  into a cell embedding vector. (See  
404 Online Methods 4.2.2). An example would be the cell type assignment task, where the cell embed-  
405 dings are used to predict cell type labels by an added classifier in the CLS training objective.

406 The input dimension  $M$  can reach tens of thousands of genes, significantly exceeding the in-  
407 put length of conventional transformers commonly used in NLG. To address this challenge and  
408 ensure efficient self-attention mechanisms, we leverage more advanced approaches such as Flash-  
409 Attention [16]. This implementation effectively enhances the model capacity and enables effective  
410 processing of large-scale input dimensions. Other efficient transformers can also be utilized, such  
411 as Transformers with linear complexity (Linformer) [61] and Kernelized Self-Attention (KSA) [28].

### 412 4.2.2 Cell representation

413 Each cell is considered a "sentence" composed of genes, and its representation  $\mathbf{h}_c^{(i)} \in \mathbb{R}^D$  is obtained  
414 by aggregating the learned gene-level representations  $\mathbf{h}_n^{(i)}$ . Various pooling operations, such as  
415 element-wise mean-pooling or weighted-pooling, can be readily employed in this context. In this  
416 study, we opt to employ a special token  $\langle cls \rangle$  for the cell representation, enabling the model  
417 to learn the pooling operation within transformer blocks. The  $\langle cls \rangle$  token is appended to the  
418 beginning of the input tokens, and the final embedding at this position is extracted as the cell  
419 representation. Consequently, we have the cell embedding  $\mathbf{h}_c^{(i)}$  equals to a row in the stacked  
420 final-layer embeddings  $\mathbf{h}_n^{(i)}[\langle cls \rangle]$ , where the  $[\langle cls \rangle]$  operation retrieves the row at the index  
421 corresponding to the  $\langle cls \rangle$  token in the input.

### 422 4.2.3 Condition tokens for batch and modality

423 We use additional sets of tokens to represent different batches and sequencing modalities, specif-  
424 ically for the scRNA-seq and scMultiomic integration tasks. This is similar to condition tokens

425 introduced in Online Methods 4.1, and implemented similarly using the standard embedding lay-  
426 ers. The modality tokens  $\mathbf{t}_m^{(i)}$  are associated with individual input features  $g_j$  (e.g., to indicate  
427 whether it is a gene, region or protein). The batch tokens are on the cell level originally but can  
428 be propagated to all features of a single cell as well. In other words, the same batch token  $t_b^{(i)}$  can  
429 be repeated up to the length  $M$  of input features of single cell  $i$ :

$$\mathbf{t}_b^{(i)} = [t_{b,1}^{(i)}, t_{b,2}^{(i)}, \dots, t_{b,M}^{(i)}] = [t_b^{(i)}, t_b^{(i)}, \dots, t_b^{(i)}]. \quad (7)$$

430 The difference between the tokens described in Online Methods 4.1 and the batch and modal-  
431 ity tokens is that these embeddings are not used as input to the transformer blocks. Instead,  
432 they are concatenated with the transformer output on either feature or cell level prior to entering  
433 specific fine-tuning objectives. This is to prevent the transformer from amplifying the attention  
434 within features of same modalities while underestimating those of different modalities. Further-  
435 more, knowing the modality and/or batch identities facilitates gene expression modelling in the  
436 downstream fine-tuning objectives. As the model learns to predict expression values conditioned  
437 on modality and/or batch identities, such biases are implicitly removed from the gene and cell  
438 representations themselves. This serves as a technique to facilitate batch correction.

439 As an example, in the scMultiomic integration task, we concatenate the transformer output with  
440 the sum of batch and modality embeddings. This serves as input to the downstream fine-tuning  
441 objectives for expression modelling:

$$\mathbf{h}_n^{(i)} = \text{concat}(\mathbf{h}_n^{(i)}, \text{emb}_b(\mathbf{t}_b^{(i)}) + \text{emb}_m(\mathbf{t}_m^{(i)})), \quad (8)$$

442 where  $\text{emb}_b$  and  $\text{emb}_m$  denote the batch and modality embedding layers respectively.

443 Alternatively, in the scRNA-seq integration task, concatenation of batch embedding with the  
444 cell representation yields the following representation:

$$\mathbf{h}_c^{(i)} = \text{concat}(\mathbf{h}_c^{(i)}, \text{emb}_b(t_b^{(i)})), \quad (9)$$

445 where  $t_b^{(i)}$  denotes the batch identity of cell  $i$ .

## 446 4.3 Generative pre-training

### 447 4.3.1 Foundation model pre-training

448 The foundation model is designed to be a generalizable feature extractor that can benefit a diverse  
449 range of downstream tasks. It contains the entire set of genes in the human genome. The expression  
450 values were normalized, log-transformed, and binned prior to model training (See Online Methods  
451 4.1). To speed up the training, we restrict the input to only genes with non-zero expressions for  
452 each input cell. This strategy provides useful pre-trained results contributing to the subsequent  
453 finetuning stage, where we include all genes with zero expressions as well by default. To efficiently  
454 train the model to capture gene-gene relation and gene-cell relation, we introduce a new generative  
455 training strategy with specialized attention masks as detailed in the following section.

### 4.3.2 Attention mask for generative pre-training

Self-attention has been widely used to capture the co-occurrence patterns among tokens. In natural language processing, this has been achieved mainly in two ways: (1) masked token prediction used in transformer encoder models such as BERT [18] and Roberta [33], where randomly masked tokens in the input sequence are predicted in the model’s output; (2) auto-regressive generation with sequential prediction in causal transformer decoder models such as the OpenAI GPT series [48, 49, 6, 43]. The generative pre-training used in OpenAI GPT3 [6] and GPT4 [43] employs a unified framework in which the model predicts the most likely next token from a “prompt” consisting of known input tokens. This framework offers great flexibility to be utilized in various natural language generation (NLG) applications and demonstrates new capabilities such as contextual awareness in zero-shot and few-shot settings [7]. We believe that the generative training can be beneficial to single-cell models in a similar manner. Specifically, we are interested in two tasks: (1) generating unknown gene expression values based on known gene expressions, i.e., generation by “gene prompts”, and (2) generating whole genome expressions given an input cell type condition, i.e., generation by “cell prompts”.

Despite similar usage of tokens and prompts, modelling genetic reads is inherently different from natural language due to the non-sequential nature of the data. Unlike words in a sentence, the order of genes within a cell is interchangeable, and there is no equivalent concept of “next gene” to predict. This makes it challenging to apply the causal masking formulation from GPT models directly in single-cell domain. To address this challenge, we developed a specialized attention masking mechanism for scGPT that defines the order of prediction based on attention scores.

The scGPT’s attention mask supports both gene-prompt and cell-prompt generations in a unified way. The binary attention mask is applied on the self-attention map in the transformer blocks. For an input  $\mathbf{h}_l^{(i)} \in \mathbb{R}^{M \times D}$  of  $M$  tokens (See Online Methods 4.2.1), the transformer block will generate  $M$  query and key vectors to compute the attention map,  $\mathbf{A} \in \mathbb{R}^{M \times M}$ . The attention mask is of the same size  $M \times M$ . We visualize the attention mask in Supplementary Figure S1A, where queries are organized in rows and keys in columns. The token identity associated with each column of the mask is annotated at the bottom of the figure, namely  $\langle cls \rangle$ , known genes, and unknown genes. Each token in the input embedding  $\mathbf{h}_l^{(i)}$  can be one of these three groups: (1) the reserved  $\langle cls \rangle$  token for cell embedding (introduced in Online Methods 4.2.2), (2) known genes with token embeddings and expression value embeddings, and (3) unknown genes whose expression values are to be predicted. The rule of thumb for scGPT’s attention-masking is to only allow attention computation between embeddings of the “known genes” and the query gene itself. In each generation iteration, scGPT predicts the gene expression values of a new set of genes, and these genes in turn become the “known genes” in the next iteration for attention computation. This approach reflects the casual masking design with next token prediction in the conventional transformer decoders by making sequential predictions in the non-sequential single-cell data.

As illustrated in Supplementary Figure S1A, during training, we randomly pick a ratio of the genes as unknown so their expression values are omitted in the input. The queries on the positions of these unknown genes are only allowed with attention computation on the known genes and the query gene itself. For example, the last gene to predict at position  $M$  has attention scores with the cell embedding, known genes and itself only, but not the other unknown genes, as illustrated in the last row of the attention mask. The scGPT model predicts the expressions for these unknown genes via the stacked transformer blocks with the masked attention map described above. The inference steps are illustrated in Supplementary Figure S1B. During the inference for cell-prompt generation, scGPT generates all genome-wide gene expressions conditioned on the specific cell types. A trained cell embedding is inputted at the first position representing the cell type condition. The whole generation process of thousands of gene expressions is conducted in  $K$  iterative steps (i.e.,  $K = 3$



504 steps in Supplementary Figure S1B). For example, in one iteration  $i \in \{1, 2, \dots, K\}$ , the attention  
505 masking mechanism allows attention with all predicted genes from previous 0 to  $i - 1$  iterations. In  
506 each iteration, scGPT selects the top  $1/K$  genes from the unknown set with the highest prediction  
507 confidence to be included as known genes in the next iteration  $i + 1$ . Intuitively, this workflow  
508 streamlines the generation of large groups of gene expressions in an auto-regressive manner, where  
509 gene expressions with highest prediction confidence are first generated and used to help subsequent  
510 rounds of generation. The gene-prompt generation works similarly in an iterative manner. The  
511 difference is that it starts with a set of known genes with observed expression values, instead of a  
512 cell embedding.

513 The scGPT attention masking unifies the encoding process of known genes and the generation  
514 on unknown genes. It also stands as one of the first transformer schemes to conduct auto-regressive  
515 generation for non-sequential data.

## 516 4.4 Fintuning objectives

517 scGPT leverages various fine-tuning objectives to facilitate the learning of biologically valid repre-  
518 sentations of cells and genes, as well as for regularization purposes such as batch correction.

519 **Gene Expression Prediction (GEP)** To encourage the learning of gene-gene interactions,  
520 scGPT incorporates gene expression prediction. Within each cell, a subset of genes and their  
521 corresponding expression values  $\mathbf{x}^{(i)}$  are randomly masked. scGPT is optimized to accurately  
522 predict the expression values at the masked positions. This fine-tuning objective benefits the  
523 model in effectively encoding co-expressions among the genes in the dataset. Specifically, we  
524 employ a fully connected MLP to estimate the expression value for  $M$  genes, on the transformer  
525 output. The optimization of this objective involves utilizing the cross entropy loss at the masked  
526 positions, denoted as  $\mathcal{M}_{mask}$ . The GEP works as follows,

$$\begin{aligned} \tilde{\mathbf{x}}^{(i)} &= \text{MLP}(\mathbf{h}_n^{(i)}), \\ \mathcal{L}_{GEP} &= \frac{1}{|\mathcal{M}_{mask}|} \sum_{j \in \mathcal{M}_{mask}} \text{ce}(\tilde{x}_j^{(i)}, x_j^{(i)}). \end{aligned} \quad (10)$$

527 Here,  $\tilde{\mathbf{x}}^{(i)} \in \mathbb{N}^M$  represents the row of expression estimates for cell  $i$ , and  $\text{ce}$  denotes the cross  
528 entropy function. It is worth noting that in integration tasks, we use  $\mathbf{h}_n^{(i)}$  in Equation 8 instead  
529 of  $\mathbf{h}_n^{(i)}$ .

530 GEP presents a general self-supervised finetuning objective, which aims to forecast gene ex-  
531 pression values. In certain downstream tasks, such as the perturbation prediction, the model is  
532 required to predict perturbed gene expression values instead of the original values. We refer to this  
533 variation as **perturb-GEP**. We maintain the MLP estimator in equation 10, but utilize the gene  
534 expressions post-perturbation as the target  $x_j^{(i)}$ . In perturb-GEP, the predicted expression values  
535 are simply altered to apply to all valid target positions instead of solely the masked positions in  
536 GEP.

537 **Gene Expression Prediction for Cell Modelling (GEPC)** This finetuning objective oper-  
538 ates similarly to GEP, but predicts gene expression values based on the cell representation  $\mathbf{h}^{(i)}_c$  to

539 explicitly foster cell representation learning. For each gene  $j$  in an input cell  $i$ , we create a query  
540 vector  $\mathbf{q}_j$  and utilize the parameterized inner product of  $\mathbf{q}_j$  and the cell representation  $\mathbf{h}^{(i)}_c$  as the  
541 predicted expression value.

$$\begin{aligned}\mathbf{q}_j &= \text{MLP}(\text{emb}_g(\mathbf{t}_g^{(i)})), \\ \tilde{x}_j^{(i)} &= \mathbf{q}_j \cdot \mathbf{W} \mathbf{h}_c^{(i)}, \\ \mathcal{L}_{GEPC} &= \frac{1}{|\mathcal{M}_{mask}|} \sum_{j \in \mathcal{M}_{mask}} \text{ce}(\tilde{x}_j^{(i)}, x_j^{(i)})\end{aligned}\tag{11}$$

542 GEPC inherits the gene token embedding,  $\text{emb}_g(\mathbf{t}^{(i)}_g)$ , from Equation equation 5. In integration  
543 tasks, we utilize  $\mathbf{h}_c^{(i)}$  from Equation 9 instead of  $\mathbf{h}_c^{(i)}$ , and we also concatenate the modality and/or  
544 batch embeddings to  $\text{emb}_g$ . In our experiments, we observed that combining GEP and GEPC leads  
545 to significantly improved performance compared to using either method individually.

546 **Elastic Cell Similarity (ECS)** This finetuning objective enhances cell representations through  
547 the utilization of a similarity learning loss [31]:

$$\mathcal{L}_{ECS} = -(\text{sim}(\mathbf{h}_c^{(i)}, \mathbf{h}_c^{(i')}) - \beta)^2,\tag{12}$$

548 where  $\text{sim}$  represents the cosine similarity function, while  $i$  and  $i'$  refer to two cells within the mini-  
549 batch. Additionally,  $\beta$  denotes a predefined threshold. The underlying idea behind this approach is  
550 to enhance the similarity between pairs exhibiting cosine similarity values above  $\beta$ , thereby making  
551 them even more similar. Conversely, dissimilar pairs are encouraged to be further apart.

552 **Domain Adaptation via Reverse Back-propagation (DAR)** Cell representation learning  
553 is hindered by the presence of batch effects, which result from non-biological batch differences  
554 introduced by sequencing technologies [19, 59]. To mitigate this problem, we employ a distinct  
555 multi-layer perceptron (MLP) classifier to predict the sequencing batch associated with each input  
556 cell, and modify the back-propagation process by reversing the gradients within the model. This  
557 approach leverages insights from the robust domain adaptation method proposed by Ganin and  
558 Lempitsky [20].

559 **Cell Type Classification (CLS)** This finetuning objective is designed to leverage the learned  
560 cell representations to annotate single cells. We use a separate MLP classifier to predict the cell  
561 types from their cell representations  $\mathbf{h}_c^{(i)}$ . This finetuning objective is optimized with cross entropy  
562 loss  $\text{ce}$  between the predicted cell type probabilities and ground-truth labels.

## 563 4.5 Finetuning on downstream tasks

564 **Batch correction on integrating multiple scRNA-seq datasets** Batch effects can be a  
565 major confounder in cell type clustering when the input contains multiple datasets from different  
566 sequencing batches or technologies. Therefore, we aim to correct batch effects while preserving bio-  
567 logical variances when integrating multiple scRNA-seq datasets. For finetuning on this integration  
568 task, the common set of gene tokens between the pre-trained foundation model and the current

569 dataset were retained. We further selected a subset of highly variable genes from the common  
570 set as input. The gene expression values were normalized, log-transformed and binned prior to  
571 model training. All pre-trained model weights were used to initialize the finetuned model. All gene  
572 tokens with both zero and non-zero expression values were used in training. In addition to GEP  
573 and GEPC, the ECS, DAR and DSBN finetuning objectives were optimized simultaneously for  
574 enhanced cell contrastive learning and explicit batch correction through reverse back-propagation  
575 and domain-specific normalization.

576 **Cell type annotation** For the cell type annotation task, we finetuned the model on a reference  
577 set with ground-truth labels, and validated the annotation performance on an external query set.  
578 The common set of gene tokens between the pre-trained foundation model and the reference set was  
579 retained. We pre-processed the expression values prior to model training similar to the integration  
580 task. All pre-trained model weights were used to initialize the finetuned model. All gene tokens  
581 with both zero and non-zero expression values were used in training. The CLS finetuning objective  
582 was used to minimize the classification loss.

583 **Perturbation prediction** Gene editing techniques applied to scRNA-seq experiments have re-  
584 vealed cellular responses to various genetic perturbations. However, the vast combinatorial space  
585 of potential gene perturbations quickly surpasses the limits of feasible experimentation. Conse-  
586 quently, machine learning approaches have been employed to leverage known perturbations and  
587 predict unknown ones. To fine-tune the perturbation prediction task, we initially selected highly  
588 variable genes and pre-processed the expression values prior to model training. All pre-trained  
589 model weights were utilized for initializing the fine-tuned model. During training, all gene tokens  
590 with both zero and non-zero expression values were included. To achieve this, we adopted the  
591 perturb-GEP finetuning objective with two modifications to the training setup. Firstly, instead  
592 of utilizing the masked and unmasked versions of the same cell as input and learning target, we  
593 employed a control cell as the input and the perturbed cell as the target. This was achieved by ran-  
594 domly pairing a non-perturbed control cell with each perturbed cell to construct the input-target  
595 pairs. Secondly, rather than randomly masking gene positions as in the original GEP setting, we  
596 employed the target perturbed genes as positions for prediction. The input values consisted of the  
597 non-perturbed gene expression values rather than mask values. Consequently, the model learned  
598 to predict the post-perturbation responses based on the control gene expressions.

599 **Integrative representation learning for scMultiomic data** scMultiomic data may contain  
600 different sequencing modalities in each batch, which presents a more challenging scenario for inte-  
601 grative analysis. We examined two data integration settings, paired and mosaic, for scMultiomic  
602 data. In the paired setting, all samples (cells) share all the data modalities sequenced. In the  
603 mosaic setting, some batches share a few common data modalities but not all. Due to the presence  
604 of additional ATAC and/or protein tokens, we inherited the trained gene embeddings for RNA  
605 data only, and trained the additional token embeddings and rest of the model from scratch. All  
606 tokens with both zero and non-zero expression values were used in training. We used an additional  
607 set of modality tokens to indicate the data type of each token (i.e., gene, region, or protein) and  
608 to facilitate the masked gene and value prediction in GEP and GEPC finetuning objectives (See  
609 Online Methods 4.2.3). In the paired setting, the model was optimized with GEP and GEPC  
610 finetuning objectives. In the mosaic setting, DAR was included to facilitate multi-modal batch  
611 correction.

612 **Gene Regulatory Network Inference** In the zero-shot setting, we extracted the gene similar-  
613 ity network from scGPT models’s gene embeddings based on cosine similarities. In the finetuned  
614 setting, we constructed the gene networks in a similar manner from the scGPT model finetuned  
615 on the Immune Human dataset. Following Ceglia et al. [10]’s pipelines, we further extracted gene  
616 programs from the gene embedding clusters that consist of five or more genes. See Online Methods  
617 4.7 for more details on gene network analysis and validation.

## 618 4.6 Datasets

619 **CELLxGENE scRNA-seq human PBMC Collection** We retrieved over 10.3 million human  
620 PBMC scRNA-seq samples from the CELLxGENE portal [11] for foundation model pre-training.  
621 A total of 65 datasets were collected from CELLxGENE by filtering on Organism (i.e., Homo  
622 sapiens), Tissue (i.e., blood, bone marrow), and Disease (i.e., normal, COVID-19, influenza).

623 **PBMC 10K** The PBMC 10K dataset comprises two scRNA-seq batches of human peripheral  
624 blood mononuclear cells (PBMCs) obtained from a healthy donor. The dataset was re-processed  
625 by Gayoso et al. [21], resulting in the identification of 3,346 differentially expressed genes. The  
626 first batch encompasses 7,982 cells, while the second batch encompasses 4,008 cells. The cell  
627 groups annotated using Seurat [55] consist of 9 categories, namely B cells, CD4 T cells, CD8 T  
628 cells, CD14+ Monocytes, Dendritic Cells, NK cells, FCGR3A+ Monocytes, Megakaryocytes, and  
629 Other.

630 **Immune Human** The Immune Human dataset encompasses five scRNA-seq datasets: one de-  
631 rived from human bone marrow and four from human peripheral blood. Various sequencing tech-  
632 nologies were employed, including 10X Genomics, 10X Genomics v2, 10X Genomics v3, and Smart-  
633 seq2. The dataset comprises a total of 33,506 cells and includes 12,303 genes. The ten distinct  
634 batches were defined based on the origin of the donors. The harmonized data encompass 16 cell  
635 groups. We used the data re-processed and the annotations provided by Luecken et al. [36].

636 **hPancreas** The hPancreas dataset contains five scRNA-seq datasets of human pancreas cells,  
637 re-processed by Chen et al. [12] for the cell type assignment task. The five datasets were split into  
638 reference and query sets by data sources. The reference set consists of Braon[5] and Muraro[39]  
639 datasets, and the inference set consists of Xin[63], Segerstolpe[56], and Lawlor[30] datasets. The  
640 reference and query sets both have 3,000 genes, and ground-truth annotations retained from their  
641 original publications. The reference set contains 10,600 cells of 13 cell groups (alpha, beta, ductal,  
642 acinar, delta, PSC, PP, endothelial, macrophage, mast, epsilon, schwann, and t\_cell). The query  
643 set contains 4,218 cells of 11 cell groups (alpha, beta, ductal, PP, acinar, delta, PSC, endothelial,  
644 epsilon, mast, MHC class II). Note that MHC class II is a new cell type in the query set not  
645 previously seen in the reference set.

646 **Adamson** The Adamson perturbation dataset contains gene expression data from the K562  
647 leukemia cell line perturbed by Pertub-seq [1]. This dataset includes 87 unique one-gene pertur-  
648 bations, each replicated in around 100 cells.

649 **Norman** The Norman perturbation dataset contains gene expression data from the K562 leukemia  
650 cell line perturbed by Pertub-seq [41]. This dataset has 131 two-gene perturbations and 105 one-

651 gene perturbations. Each perturbation is replicated in around 300-700 cells.

652 **10X Multiome PBMC** The 10X Multiome PBMC dataset [14] contains paired single-cell RNA  
653 and ATAC data on human PBMC cells sequenced by the 10X Single Cell Multiome protocol. In this  
654 dataset, all samples came from the same healthy donor. Each cell contains both gene expression  
655 and chromatin accessibility measurements. The processed data by Cao and Gao [9] contains 9,631  
656 cells with counts from 29,095 genes and 107,194 regions. The annotations include 19 cell groups  
657 (CD14 Mono, CD16 Mono, CD4 Naive, CD4 TCM, CD4 TEM, CD8 Naive, CD8 TEM\_1, CD8  
658 TEM\_2, HSPC, Intermediate B, MAIT, Memory B, NK, Naive B, Plasma, Treg, cDC, gdT, and  
659 pDC).

660 **ASAP PBMC** The ASAP PBMC dataset contains four sequencing batches with three data  
661 modalities (gene expression, chromatin accessibility, and protein abundance) [38]. The four batches  
662 each contain 5,023, 3,666, 3,517, and 4,849 cells respectively. In batches 1 and 2, all samples have  
663 4,768 genes and 216 protein measurements from CITE-seq. In batches 3 and 4, all samples have  
664 17,742 regions and the same 216 protein measurements from ASAP-seq. The annotations by [38]  
665 contain 4 cell groups (Bcell, Myeloid, NK, and Tcell).

## 666 4.7 Experiment Setup

667 **scRNA-seq batch integration** In this work, we compared the performance of scGPT with  
668 three other methods, namely Seurat [55], Harmony [29], and scVI [34]. The evaluation covers  
669 batch correction and cell type clustering on two integration datasets: PBMC 10K [21] and Immune  
670 Human [36]. Harmony and scVI are highlighted as the top-performing methods in the recent  
671 integration benchmark conducted by Luecken et al. [36]. To ensure a fair comparison, all methods  
672 were provided with the same number of 1,200 highly variable genes as input. Gene expression  
673 values were normalized per cell by considering the total counts across all genes and subsequently  
674 log-transformed. The integrated cell embeddings were obtained after the completion of training  
675 and were used for evaluation.

676 The evaluation of the integrated cell embeddings was performed using biological conservation  
677 metrics proposed by Luecken et al. [36]. These metrics include the normalized mutual informa-  
678 tion ( $NMI_{cell}$ ), adjusted Rand index ( $ARI_{cell}$ ), and average silhouette width ( $ASW_{cell}$ ). These  
679 scores measure the consistency between the derived cell type clusters and the ground truth labels.  
680 For easier comparison, we also computed the average of these metrics, referred to as  $AvgBIO$ .  
681 Additionally, we reported the batch correction metrics proposed by Luecken et al. [36] to assess  
682 batch mixing. The batch correction performance was quantified using the inverse of the average  
683 silhouette width for batch clustering, denoted as  $ASW_{batch}$ , and the graph connectivity measure,  
684 denoted as  $GraphConn$ . We computed  $AvgBATCH$  as the average of  $ASW_{batch}$  and  $GraphConn$   
685 to summarize the batch mixing performance. Furthermore, we introduced an *Overall* score, which  
686 is a weighted sum of  $AvgBIO$  and  $AvgBATCH$ , consistent with the approach taken by [36]. See  
687 Supplementary Online Methods S.2 for details of metric calculations.

688 **scRNA-seq cell type annotation** We benchmarked scGPT against the recent transformer-  
689 based cell type annotation method TOSICA [12] on the hPancreas dataset. We used the same  
690 pre-processed reference and query sets by Chen et al. [12] for model training and validation. The  
691 predicted cell type labels on the query set were retrieved for evaluation.

692 We evaluated cell type assignment performance based on four standard classification metrics,  
693 *Accuracy*, *Precision*, *Recall*, and *MacroF1*. *Accuracy*, *Precision*, and *Recall* are calculated  
694 globally for overall performance, whereas *MacroF1* is averaged per class to increase the weighing  
695 of rare cell types. We also reported a normalized confusion matrix with *Precision* by cell type for  
696 additional details. See Supplementary Online Methods S.2 for details on metric calculations.

697 **scRNA-seq perturbation** We compare scGPT against the recent perturbation prediction method  
698 GEARS [53]. To ensure consistency, we followed the pre-processing steps outlined by Roohani,  
699 Huang, and Leskovec [53] in their benchmark. Initially, gene expression values were normalized  
700 per cell using the total counts across all genes, and a logarithmic transformation was applied. Sub-  
701 sequently, we selected 5,000 highly variable genes and incorporated any perturbed genes that were  
702 not initially considered into the gene set. In the experiments, for one-gene perturbations in both  
703 datasets Adamson et al. [1] and Norman et al. [41], the perturbations are split to ensure that test  
704 perturbations are not seen in training, i.e., no cells in training set has undergone any of the test  
705 perturbations. For two-gene perturbations in the Norman et al. [41] dataset, the train-test split  
706 consists of three scenarios with increasing difficulty: (1) 0/2 unseen genes, (2) 1/2 unseen genes,  
707 and (3) 2/2 unseen genes in the training set.

708 To evaluate the accuracy of perturbation prediction, we employed the Pearson correlation co-  
709 efficient (*corr*) between the predicted gene expressions and the ground-truth expression values.  
710 Additionally, we calculated a variant of the Pearson metric based on the amount of change in ex-  
711 pression post-perturbation compared to the control, denoted as *corr*( $\Delta$ ). Furthermore, we reported  
712 these Pearson metrics for different gene sets, including all genes (*ALL*), and the top 20 differen-  
713 tially expressed genes (*DE*). Thus, we presented four evaluation metrics in total, namely *corr* and  
714 *corr*( $\Delta$ ) for the *ALL* and *DE* conditions, respectively. See Supplementary Online Methods S.2 for  
715 details of metric calculation.

716 **scMultiomic integration** We benchmarked scGPT in two integration settings, paired and mo-  
717 saic, against the recent scMultiomic integration methods Seurat v4 [24], scGLUE [9] and scMoMat  
718 [65] respectively. In the paired data integration experiment, we benchmarked scGPT with scGLUE  
719 [9] and Seurat v4 [24] on the 10X Multiome PBMC [14] dataset. The same 1,200 highly variable  
720 genes and 4,000 highly variable peaks were used as input to all methods. In the mosaic data inte-  
721 gration experiment, we benchmarked scGPT with scMoMat [65] on the ASAP PBMC [38] dataset.  
722 The same 1,200 highly variable genes, 4,000 highly variable peaks, and all 216 protein features  
723 were used as input to both methods. While keeping the input feature set consistent, we used each  
724 method's custom pre-processing pipeline to normalize the expression values. The integrated cell  
725 embeddings were retrieved for evaluation after training.

726 In both paired and mosaic data integration settings, we evaluated cell embedding quality on  
727 the four biological conservation metrics *NMI<sub>cell</sub>*, *ARI<sub>cell</sub>*, *AWS<sub>cell</sub>*, and *AvgBIO*. In the  
728 mosaic data integration setting, we further evaluated mixing of different omic batches with the  
729 three batch correction metrics *AWS<sub>batch</sub>*, *GraphConn*, and *Avg\_BATCH*. An overall score was  
730 also reported on the mosaic integration experiment. See Supplementary Online Methods S.2 for  
731 details on metric calculation.

732 **Gene Regulatory Network Inference** We validated scGPT's gene similarity network against  
733 the known HLA and CD antigen networks. For each network, we first defined the related gene set by  
734 filtering on gene names with set prefixes (i.e., HLA- and CD-). We then filtered on genes involved  
735 in the Immune System R-HSA-168256 pathway from the Reactome 2022 database [50]. For the CD

736 antigens, we used the common genes with the selected HVG set from the Immune Human dataset  
737 for the ease of comparison between pre-trained and finetuned models. We then extracted gene  
738 embeddings of these selected genes from the scGPT model and constructed a similarity network  
739 based on cosine similarity. We highlighted sub-networks of strong connections by selecting edges  
740 with cosine similarities greater than a certain threshold (i.e., 0.5 for HLA and 0.4 for CD antigen  
741 network). We then compared the sub-networks against known functional groups in immunology.  
742 Furthermore, we evaluated the gene similarity relationships encoded by the scGPT model with  
743 Reactome. Following Ceglia et al. [10]’s pipelines, we first evaluated whether the neighbors of a  
744 gene involved in a known pathway belong to that pathway. Using CD8A gene as an example,  
745 we ranked its 10 nearest neighbors (including itself) by cosine similarity in the selected HVG set  
746 from the Immune Human dataset, and examined their membership in the Immune System R-HSA-  
747 168256 pathway. Subsequently, on a system level, we examined the relationship between cosine  
748 similarity of pairs of genes and the number of common pathways that a gene pair is involved in.  
749 We reported the Pearson correlation score between cosine similarity score and pathway coverage  
750 on the entire gene set across all pathways in Reactome.

## 751 4.8 Implementation Details

752 The pretrained foundation model has an embedding size of 512. It consists of 12 stacked transformer  
753 blocks with 8 attention heads each. The fully connected layer has hidden size of 512. In pre-training,  
754 we randomly split the data and used 97% (10 million) of the data for training and 3% (0.3 million)  
755 for validation. We set the ratio of genes to generate to be uniformly sampled from three options of  
756 0.25, 0.50 and 0.75. The model was optimized by the Adam optimizer, using a mini-batch size of  
757 32, at a starting learning rate of 0.0001 and a 0.9 weight decay after each epoch. The model was  
758 trained for a total of 6 epochs.

759 For the tasks of scRNA-seq batch integration, cell type annotation, and perturbation prediction,  
760 we utilized the same model configuration inherited from the pre-trained model. During the fine-  
761 tuning process, we initiated with a learning rate of 0.0001, which decayed to 90% after each epoch.  
762 The mask ratio for GEP and GEPC was set to 0.4, while the parameter  $\beta$  in ECS was set to 0.6.  
763 When combined with other losses, ECS was assigned a weighting of 10. To divide the datasets  
764 into training and evaluation sets, we employed a ratio of 9:1. The model was trained for a fixed  
765 duration of 30 epochs, and after each epoch, the GEP loss value was evaluated on the validation  
766 set. The reported results correspond to the model with the best validation score. Notably, for the  
767 perturbation task (refer to Section section 2.4), we observed that the model typically converged  
768 within 3 epochs, and we report the best-validated model accordingly.

769 For the multi-omic integration task, we loaded the gene embeddings from the pre-trained model  
770 and used the same embedding size of 512 for all tokens (i.e., including gene, ATAC-peak, and/or  
771 protein). The main model is set to have 4 stacked transformer blocks with 8 attention heads each,  
772 and a hidden layer size of 512. Each dataset is split into train and evaluation sets at 9:1 ratio.  
773 We used DAR weighing of 1.0 for batch integration. We used a starting learning rate of 0.001 and  
774 a weight decay of 0.95 after each epoch. We trained the model for fixed 60 epochs and similarly  
775 reported the best-validated model.

776 We used the SCANPY python library [62] for gene expression pre-processing, including nor-  
777 malization, log-transformation and highly variable gene selection. We used the EpiScanpy python  
778 library [15] on chromatin accessibility data for highly variable peak selection. In the scRNA-seq  
779 batch integration and scMultiomic integration tasks, the evaluation metrics are calculated using the  
780 implementation in `scib.metrics` by Luecken et al. [36]. In the cell-annotation task, the evaluation  
781 metrics are implemented using the `scikit-learn` package.

782

## 5 Acknowledgement

783  
784  
785

We would like to express our sincere gratitude to, Dr. Nan Duan, for his invaluable guidance and support throughout the project. We appreciate the valuable feedback from Dr. Lin Zhang during the writing of the manuscript.

786

## References

787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823

- [1] Britt Adamson et al. “A multiplexed single-cell CRISPR screening platform enables systematic dissection of the unfolded protein response”. In: *Cell* 167.7 (2016), pp. 1867–1882.
- [2] Philipp Angerer et al. “Single cells make big data: New challenges and opportunities in transcriptomics”. In: *Current opinion in systems biology* 4 (2017), pp. 85–91.
- [3] Dvir Aran et al. “Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage”. In: *Nature immunology* 20.2 (2019), pp. 163–172.
- [4] Britta Bade et al. “Differential expression of the granzymes A, K and M and perforin in human peripheral blood lymphocytes”. In: *International immunology* 17.11 (2005), pp. 1419–1428.
- [5] Maayan Baron et al. “A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure”. In: *Cell systems* 3.4 (2016), pp. 346–360.
- [6] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.
- [7] Sébastien Bubeck et al. *Sparks of Artificial General Intelligence: Early experiments with GPT-4*. 2023. arXiv: [2303.12712](https://arxiv.org/abs/2303.12712) [cs.CL].
- [8] Lars Buitinck et al. “API design for machine learning software: experiences from the scikit-learn project”. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.
- [9] Zhi-Jie Cao and Ge Gao. “Multi-omics single-cell data integration and regulatory inference with graph-linked embedding”. In: *Nature Biotechnology* 40.10 (2022), pp. 1458–1466.
- [10] Nicholas Ceglia et al. “GeneVector: Identification of transcriptional programs using dense vector representations defined by mutual information”. In: *bioRxiv* (2022), pp. 2022–04.
- [11] Chanzuckerberg Initiative. *CZ CELLxGENE Discover*. <https://cellxgene.cziscience.com/>. Online; accessed 26 December 2022. 2022.
- [12] Jiawei Chen et al. “Transformer for one stop interpretable cell type annotation”. In: *Nature Communications* 14.1 (2023), p. 223.
- [13] Paola Cruz-Tapias, John Castiblanco, and Juan-Manuel Anaya. “Major histocompatibility complex: antigen processing and presentation”. In: *Autoimmunity: From Bench to Bedside [Internet]*. El Rosario University Press, 2013.
- [14] Darren A Cusanovich et al. “Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing”. In: *Science* 348.6237 (2015), pp. 910–914.
- [15] Anna Danese et al. “EpiScanpy: integrated single-cell epigenomic analysis”. In: *Nature Communications* 12.1 (2021), p. 5228.
- [16] Tri Dao et al. “FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness”. In: *arXiv preprint arXiv:2205.14135* (2022).
- [17] Jurrian K De Kanter et al. “CHETAH: a selective, hierarchical cell type identification method for single-cell RNA sequencing”. In: *Nucleic acids research* 47.16 (2019), e95–e95.



- 824 [18] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language un-  
825 derstanding”. In: *arXiv preprint arXiv:1810.04805* (2018).
- 826 [19] Michael Eisenstein. “Single-cell RNA-seq analysis software providers scramble to offer solu-  
827 tions.” In: *Nature Biotechnology* 38.3 (2020), pp. 254–257.
- 828 [20] Yaroslav Ganin and Victor Lempitsky. “Unsupervised domain adaptation by backpropagati-  
829 on”. In: *International conference on machine learning*. PMLR. 2015, pp. 1180–1189.
- 830 [21] Adam Gayoso et al. “A Python library for probabilistic analysis of single-cell omics data”.  
831 In: *Nature Biotechnology* 40.2 (2022), pp. 163–166.
- 832 [22] Suchin Gururangan et al. “Don’t stop pretraining: Adapt language models to domains and  
833 tasks”. In: *arXiv preprint arXiv:2004.10964* (2020).
- 834 [23] Xiaoping Han et al. “Mapping the mouse cell atlas by microwell-seq”. In: *Cell* 172.5 (2018),  
835 pp. 1091–1107.
- 836 [24] Yuhan Hao et al. “Integrated analysis of multimodal single-cell data”. In: *Cell* 184.13 (2021),  
837 pp. 3573–3587.
- 838 [25] Ashraf Haque et al. “A practical guide to single-cell RNA-sequencing for biomedical re-  
839 search and clinical applications”. In: *Genome medicine* 9.1 (2017), pp. 1–12.
- 840 [26] HCA. *HCA DCP*. <https://data.humancellatlas.org/>. Online; accessed 12 April 2023.  
841 2023.
- 842 [27] Yuge Ji et al. “Machine learning for perturbational single-cell omics”. In: *Cell Systems* 12.6  
843 (2021), pp. 522–537.
- 844 [28] Angelos Katharopoulos et al. “Transformers are rnns: Fast autoregressive transformers with  
845 linear attention”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5156–  
846 5165.
- 847 [29] Ilya Korsunsky et al. “Fast, sensitive and accurate integration of single-cell data with Har-  
848 mony”. In: *Nature methods* 16.12 (2019), pp. 1289–1296.
- 849 [30] Nathan Lawlor et al. “Single-cell transcriptomes identify human islet cell signatures and  
850 reveal cell-type-specific expression changes in type 2 diabetes”. In: *Genome research* 27.2  
851 (2017), pp. 208–222.
- 852 [31] Chundi Liu et al. “Guided similarity separation for image retrieval”. In: *Advances in Neural  
853 Information Processing Systems* 32 (2019).
- 854 [32] Jiajia Liu et al. “Machine intelligence in single-cell data analysis: advances and new chal-  
855 lenges”. In: *Frontiers in Genetics* 12 (2021), p. 655536.
- 856 [33] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv  
857 preprint arXiv:1907.11692* (2019).
- 858 [34] Romain Lopez et al. “Deep generative modeling for single-cell transcriptomics”. In: *Nature  
859 methods* 15.12 (2018), pp. 1053–1058.
- 860 [35] Malte D Luecken and Fabian J Theis. “Current best practices in single-cell RNA-seq analysis:  
861 a tutorial”. In: *Molecular systems biology* 15.6 (2019), e8746.
- 862 [36] Malte D Luecken et al. “Benchmarking atlas-level data integration in single-cell genomics”.  
863 In: *Nature methods* 19.1 (2022), pp. 41–50.
- 864 [37] Zhen Miao et al. “Multi-omics integration in the age of million single-cell data”. In: *Nature  
865 Reviews Nephrology* 17.11 (2021), pp. 710–724.
- 866 [38] Eleni P Mimitou et al. “Scalable, multimodal profiling of chromatin accessibility, gene ex-  
867 pression and protein levels in single cells”. In: *Nature biotechnology* 39.10 (2021), pp. 1246–  
868 1258.

- 869 [39] Mauro J Muraro et al. “A single-cell transcriptome atlas of the human pancreas”. In: *Cell*  
870 *systems* 3.4 (2016), pp. 385–394.
- 871 [40] Philip S Norman. “Immunobiology: The immune system in health and disease”. In: *Journal*  
872 *of Allergy and Clinical Immunology* 96.2 (1995), p. 274.
- 873 [41] Thomas M Norman et al. “Exploring genetic interaction manifolds constructed from rich  
874 single-cell phenotypes”. In: *Science* 365.6455 (2019), pp. 786–793.
- 875 [42] Sergio Oller-Moreno et al. “Algorithmic advances in machine learning for single-cell expres-  
876 sion analysis”. In: *Current Opinion in Systems Biology* 25 (2021), pp. 27–33.
- 877 [43] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: [2303.08774](https://arxiv.org/abs/2303.08774) [cs.CL].
- 878 [44] OpenAI. *CZ CELLxGENE Discover*. <https://openai.com/blog/chatgpt>. Online; accessed  
879 10 April 2023. 2023.
- 880 [45] OpenAI. *CZ CELLxGENE Discover*. <https://openai.com/product/dall-e-2>. Online;  
881 accessed 10 April 2023. 2023.
- 882 [46] Aditya Pratapa et al. “Benchmarking algorithms for gene regulatory network inference from  
883 single-cell transcriptomic data”. In: *Nature methods* 17.2 (2020), pp. 147–154.
- 884 [47] Xipeng Qiu et al. “Pre-trained models for natural language processing: A survey”. In: *Science*  
885 *China Technological Sciences* 63.10 (2020), pp. 1872–1897.
- 886 [48] Alec Radford et al. “Improving language understanding by generative pre-training”. In:  
887 (2018).
- 888 [49] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog*  
889 1.8 (2019), p. 9.
- 890 [50] Reactome. *Reactome Pathway Database: Home*. <https://reactome.org/>. 2022.
- 891 [51] Aviv Regev et al. “Science forum: the human cell atlas”. In: *elife* 6 (2017), e27041.
- 892 [52] Dennis Thompson Healthday Reporter. ‘human cell atlas’ maps 1 million cell types in 33  
893 organs. 2022. URL: [https://medicalxpress.com/news/2022-05-human-cell-atlas-mill-](https://medicalxpress.com/news/2022-05-human-cell-atlas-million.html)  
894 [ion.html](https://medicalxpress.com/news/2022-05-human-cell-atlas-million.html).
- 895 [53] Yusuf Roohani, Kexin Huang, and Jure Leskovec. “GEARS: Predicting transcriptional out-  
896 comes of novel multi-gene perturbations”. In: *bioRxiv* (2022).
- 897 [54] Abhishek Sarkar and Matthew Stephens. “Separating measurement and expression models  
898 clarifies confusion in single-cell RNA sequencing analysis”. In: *Nature genetics* 53.6 (2021),  
899 pp. 770–777.
- 900 [55] Rahul Satija et al. “Spatial reconstruction of single-cell gene expression data”. In: *Nature*  
901 *biotechnology* 33.5 (2015), pp. 495–502.
- 902 [56] Åsa Segerstolpe et al. “Single-cell transcriptome profiling of human pancreatic islets in health  
903 and type 2 diabetes”. In: *Cell metabolism* 24.4 (2016), pp. 593–607.
- 904 [57] Indhupriya Subramanian et al. “Multi-omics data integration, interpretation, and its appli-  
905 cation”. In: *Bioinformatics and biology insights* 14 (2020), p. 1177932219899051.
- 906 [58] Chi Sun et al. “How to fine-tune bert for text classification?” In: *Chinese Computational*  
907 *Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20,*  
908 *2019, Proceedings 18*. Springer. 2019, pp. 194–206.
- 909 [59] Hoa Thi Nhu Tran et al. “A benchmark of batch-effect correction methods for single-cell  
910 RNA sequencing data”. In: *Genome biology* 21.1 (2020), pp. 1–32.
- 911 [60] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information pro-*  
912 *cessing systems* 30 (2017).

- 913 [61] Sinong Wang et al. “Linformer: Self-attention with linear complexity”. In: *arXiv preprint*  
914 *arXiv:2006.04768* (2020).
- 915 [62] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. “SCANPY: large-scale single-cell  
916 gene expression data analysis”. In: *Genome biology* 19.1 (2018), pp. 1–5.
- 917 [63] Yurong Xin et al. “RNA sequencing of single human islet cells reveals type 2 diabetes genes”.  
918 In: *Cell metabolism* 24.4 (2016), pp. 608–615.
- 919 [64] Allen W Zhang et al. “Probabilistic cell-type assignment of single-cell RNA-seq for tumor  
920 microenvironment profiling”. In: *Nature methods* 16.10 (2019), pp. 1007–1015.
- 921 [65] Ziqi Zhang et al. “scMoMaT jointly performs single cell mosaic integration and multi-modal  
922 bio-marker detection”. In: *Nature Communications* 14.1 (2023), p. 384.

923        **S    Supplementary**

924        **S.1   Benchmarking results on downstream tasks**

Dataset	Model	Biological Conservation				Batch Correction			Overall
		AvgBIO	$NMI_{cell}$	$ARI_{cell}$	$ASW_{cell}$	AvgBATCH	$ASW_{batch}$	$GraphConn$	
Immune Human [36]	scGPT	<b>0.746</b>	0.824	0.784	0.630	0.894	0.817	0.970	<b>0.804</b>
	scVI [34]	0.726	0.803	0.787	0.587	<b>0.923</b>	0.871	0.975	0.805
	Seurat [55]	0.564	0.697	0.441	0.555	0.883	0.858	0.908	0.692
	Harmony [29]	0.743	0.810	0.832	0.588	0.914	0.859	0.968	0.811
PBMC 10K [21]	scGPT	<b>0.812</b>	0.834	0.869	0.732	0.940	0.949	0.931	<b>0.863</b>
	scVI	0.695	0.786	0.704	0.593	<b>0.950</b>	0.971	0.930	0.797
	Seurat	0.753	0.810	0.854	0.595	0.934	0.937	0.931	0.826
	Harmony	0.751	0.810	0.855	0.589	0.945	0.967	0.923	0.829

Table 2: scRNA-seq Integration Benchmark Results. scGPT was benchmarked with scVI [34], Seurat [55], and Harmony [29] on the Immune Human (10 batches) [36] and PBMC 10K (2 batches) [21] datasets for cell type clustering and batch correction performance. We present three aggregate scores *AvgBIO*, *AvgBATCH*, and *Overall*. These aggregate scores were calculated from three detailed biological conservation metrics ( $NMI_{cell}$ ,  $ARI_{cell}$ ,  $ASW_{cell}$ ) and two batch correction metrics ( $ASW_{batch}$ ,  $GraphConn$ ). See metric details in Supplementary Online Methods S.2

Dataset	Model	Classification Metrics			
		<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>MacroF1</i>
hPancreas [12]	scGPT	<b>0.967</b>	0.785	0.694	0.705
	TOSICA [12]	0.961	0.641	0.667	0.641

Table 3: Cell Type Annotation Benchmark Results. scGPT was benchmarked with TOSICA [12] on the hPancreas [12] dataset for cell type annotation performance. We present four classification evaluation metrics *Accuracy*, *Precision*, *Recall*, and *MacroF1*. See metric details in Supplementary Online Methods S.2.

Dataset	Model	Biological Conservation				Batch Correction			Overall
		AvgBIO	$NMI_{cell}$	$ARI_{cell}$	$ASW_{cell}$	AvgBATCH	$ASW_{batch}$	$GraphConn$	
10X Multiome PBMC [14] (Paired)	scGPT	<b>0.767</b>	0.818	0.822	0.661	-	-	-	-
	scGLUE [9]	0.747	0.815	0.806	0.619	-	-	-	-
	Seurat v4 [24]	0.722	0.784	0.691	0.691	-	-	-	-
ASAP PBMC [38] (Mosaic)	scGPT	<b>0.562</b>	0.601	0.472	0.614	<b>0.948</b>	0.904	0.992	<b>0.716</b>
	scMoMat [65]	0.546	0.448	0.557	0.633	0.916	0.849	0.983	0.667

Table 4: scMultiomic Integration Benchmark Results. For the paired 10X Multiome PBMC [14] dataset, scGPT was benchmarked with scGLUE [9] and Seurat v4 [24] for cell type clustering performance evaluated on four biological conservation metrics. Batch correction metrics are not applicable to this setting. For the mosaic ASAP PBMC [38] dataset, scGPT was benchmarked with scMoMat [65] for cell type clustering and multiomic integration performance, evaluated on eight biological conservation and batch correction metrics.

## 925 S.2 Evaluation Metric Calculations

### 926 S.2.1 Single-cell integration

927 We adopted the evaluation metric calculations outlined by Luecken et al. [36] in their benchmark  
928 study. Each metric is described below.

#### 929 Normalized Mutual Information

930 To quantify the concurrence between the cell type labels based on ground truth and the Lou-  
931 vain cluster labels obtained from integrated cell embeddings, we computed the normalized mutual  
932 information (NMI) score. The Louvain clustering was conducted across resolutions ranging from  
933 0.1 to 2, with increments of 0.1. The best score will be selected. The NMI score for cell types,  
934 referred to as  $NMI_{cell}$ , ranges between 0 and 1, where a higher score indicates a better match of  
935 cell types.

#### 936 Adjusted Rand Index

937 The adjusted rand index (ARI) was employed to assess both the agreement between the anno-  
938 tated labels and the MNI-optimized Louvain clusters. Furthermore, the rand index was adjusted  
939 to account for randomly correct labels. The ARI score for cell types, denoted as  $ARI_{cell}$ , ranges  
940 from 0 to 1, where 0 corresponds to random labeling and 1 represents a perfect match.

#### 941 Average Silhouette Width

942 The silhouette width assesses the relationship between a cell's within-cluster distances and its  
943 distances to the closest cluster boundaries. By averaging the silhouette widths of all cells, we  
944 calculate the average silhouette width (ASW) score. This score ranges from -1 to 1, where a score  
945 of 1 indicates well-separated clusters, while scores from -1 to 0 suggest overlapping clusters and  
946 misclassification.

947 For evaluating cell type clustering, we compute the ASW score based on cell type labels,  
948 represented as  $ASW_{cell}$ . To obtain this score, we utilize the following formula:

$$ASW_{cell} = (ASW_C + 1)/2$$

949 Here,  $C$  represents the cell types.

950 Regarding batch mixing evaluation, we calculate the ASW score considering batch labels and  
951 adjust it by subtracting 1. This score is denoted as  $ASW_{batch}$ . The calculation is as follows:

$$ASW_{batch} = 1 - |ASW_B|$$

952 Both  $ASW_{cell}$  and  $ASW_{batch}$  have values between 0 and 1. Higher scores indicate better  
953 cell-type clustering or batch-mixing performance.

#### 954 Graph Connectivity

955 The graph connectivity metric quantifies the average proportion of cells within each cell type  
956 that are connected through a kNN (k-nearest neighbors) graph. For every cell identity  $c$  in the  
957 set  $C$ , we compute the size of the largest connected component using kNN among cells exclusively



958 belonging to identity  $c$ . This value is divided by the total number of cells with identity  $c$  to obtain  
959 a normalized measure. The **GraphConn** score is then reported as the average across all cell  
960 types:

$$GraphConn = \frac{1}{|C|} \sum_{c \in C} \frac{|LCC(G_c^{kNN})|}{N_c}$$

961 Here,  $LCC$  represents the largest connected component, and  $N$  denotes the number of cells of  
962 each celltype.

### 963 **Aggregated Metrics**

964 The aggregated metric **AvgBIO** calculates the average of biological conservation metrics:

$$AvgBIO = (ARI_{cell} + NMI_{cell} + ASW_{cell})/3$$

965 Similarly, the aggregated metric **AvgBATCH** computes the average of batch mixing metrics:

$$AvgBATCH = (ASW_{batch} + GraphConn)/2$$

966 In accordance with the convention established in [36], an **Overall** metric is derived as the  
967 weighted average of **AvgBIO** and **AvgBATCH**:

$$AvgBATCH = 0.6 * AvgBIO + 0.4 * AvgBATCH$$

### 968 **S.2.2 Cell Type Assignment**

969 We used the standard classification metrics *Accuracy*, *Precision*, *Recall*, and *MacroF1* to evaluate  
970 cell type assignment performance. The *Accuracy*, *Precision*, *Recall*, and *MacroF1* scores are  
971 calculated from true positives ( $tp$ ), false positives ( $fp$ ), true negatives ( $tn$ ), and false negatives  
972 ( $fn$ ) globally or averaged per class.

973 The *Accuracy*, *Precision* and *Recall* scores are calculated globally:

$$Accuracy = \frac{tp}{tp + fp + tn + fn}, \quad Precision = \frac{tp}{tp + fp}, \quad Recall = \frac{tp}{tp + fn} .$$

974 The *MacroF1* score is calculated per cell type  $c$  first and averaged across cell types:

$$MacroF1 = \sum_{c \in C} \frac{F1_c}{N_c}, \quad \text{where} \quad F1_c = \frac{2 \times Precision_c \times Recall_c}{Precision_c + Recall_c} .$$

975 The above metrics are calculated using `scikit-learn`'s implementations [8].

976

## Supplementary Figures

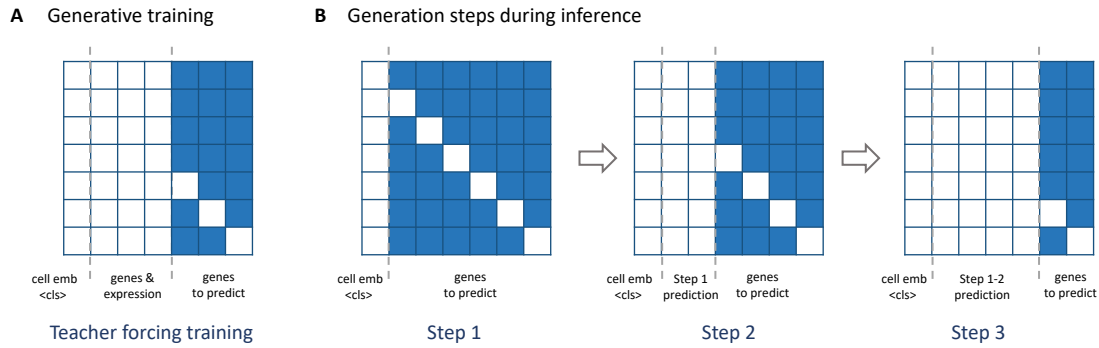
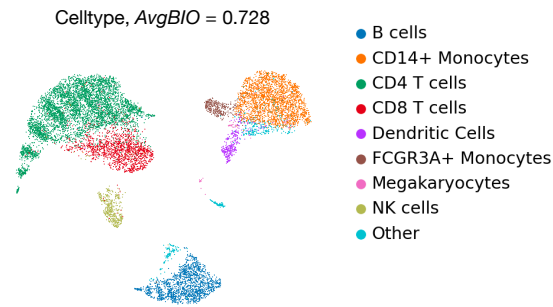


Figure S1: The scGPT Attention Mask. The masked positions are colored in blue, and the allowed positions in white. These masked and unmasked positions correspond to the  $M \times M$  attention map for  $M$  input tokens. The row indices correspond to queries and columns correspond to keys. In the self-attention computation of transformers, the attention scores on the marked positions will be removed. The token identity associated with each column is annotated below, namely cell emb  $\langle cls \rangle$  for cell embedding, genes & expressions for known genes, and genes to predict for unknown genes. (a) scGPT attention mask in training where only query gene and the known genes participate in attention computation. (b) At inference time, the attention mask at each step during the iterative process of scGPT cell-prompt generation.

### A scGPT (Zero-shot)



### B Gene Embedding (Zero-shot)

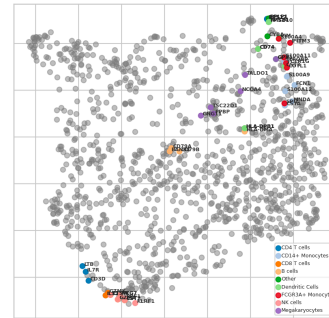


Figure S2: Visualization of Zero-shot scGPT Learned Cell and Gene Embeddings on the PBMC 10K Dataset. (a) UMAP visualization of cell embeddings colored by *cell types*. (a) UMAP visualization of gene embeddings. The highly variable genes corresponding to each *celltype* were colored accordingly.

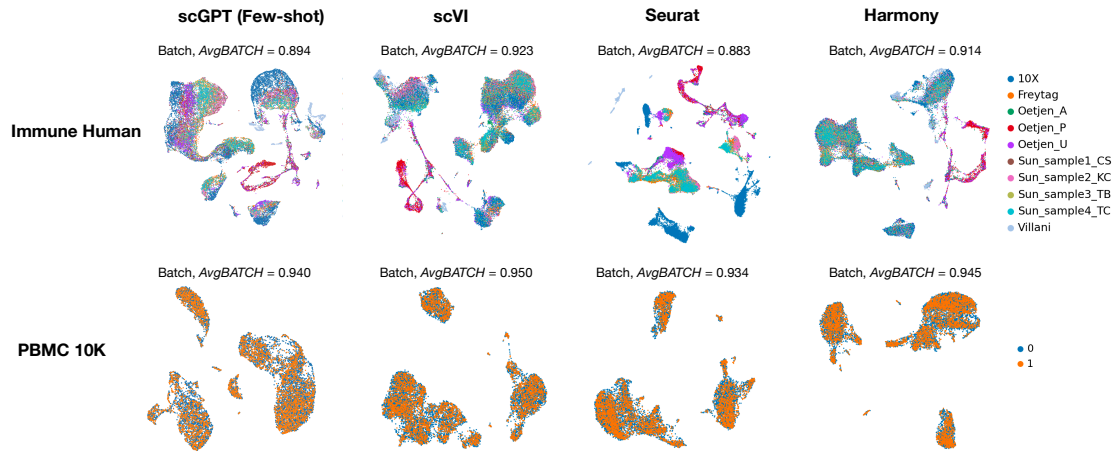


Figure S3: Benchmark of scGPT with scVI [34], Seurat [55], and Harmony [29] on the Immune Human (10 batches) [36] and PBMC 10K (2 batches) [21] Datasets for Batch Correction. UMAP visualization of cell embeddings colored by *sequencing batches*.