nature methods

Article

https://doi.org/10.1038/s41592-024-02201-0

scGPT: toward building a foundation model for single-cell multi-omics using generative AI

In the format provided by the authors and unedited

Supplementary Notes

1 Reference mapping

Reference mapping aligns new cells with curated reference datasets, facilitating automated annotation of new datasets [1]. Existing approaches [1, 2, 3] were often optimized on the reference datasets to capture the heterogeneity of cells. Thanks to the pretraining on whole-human scale cell corpus, scGPT can additionally perform reference mapping in a similar fine-tuning manner, and additionally in a zero-shot manner as well. In the zero-shot configuration, the model leverages its pre-trained weights to encode both the reference and query datasets into a unified latent space, obviating the need for additional training and thus accelerating the integration of disparate or novel biological data. In the fine-tuning setting, scGPT undergoes additional optimization processes on reference data, further aligning the cell embeddings with the unique biological context.

To annotate the cell state of each query cell, we employ a similarity search based on the ℓ_2 distance between the reference and query cell embeddings. A majority voting mechanism is applied to the k nearest neighbors from the reference set, and each query cell is assigned the cell type that is most frequent among its k-nearest reference cell neighbors. This workflow represents a notable departure from earlier methodologies that relied on trained classifiers for cell type annotation (Methods). Here, only the cell embeddings are required, thereby availing the opportunity for direct utilization of pretrained models in a zero-shot manner.

We present a comprehensive evaluation of reference mapping performance on two distinct datasets of disease samples, assessed via classification metrics *Accuracy*, *Precision*, *Recall*, and $MacroF1$ (Supplementary Table 1). Intriguingly, the zero-shot capabilities of scGPT yielded performance metrics comparable to those of established baselines, even when applied to datasets containing disease samples that were absent during the model's pre-training phase. This outcome underscores the robust generalizability intrinsic to the zero-shot capabilities of scGPT, thereby potentially reducing the need for dataset-specific training.

With fine-tuning on reference datasets using integration objectives, scGPT outperforms strong baselines of Azimuth[4] and expiMap[2] on both COVID-19 and Lung-Kim datasets. Specifically on the COVID-19 dataset, scGPT enhances MacroF1 by 23.2% over Azimuth and 63.3% over expiMap (Supplementary Table 1). We further analyze the performance gain of fine-tuning, as detailed in Supplementary Figure 11. We showcase the confusion matrix for cell type annotation for fine-tuning and zero-shot setting of scGPT. Fine-tuning improves annotation performance in the COVID-19 dataset for Dendritic cells, IGSF21+ Dendritic cells, and Erythrocytes. Notably, fine-tuning allows scGPT to identify a rare IGSF21+ Dendritic cell state in the query dataset. In the Lung-Kim dataset, we also observed increased annotation accuracy for Epithelial and Natural killer (NK) cells.

To summarize, in a zero-shot setting, scGPT builds a robust latent space using pre-trained weights, achieving comparable performance with existing methods. Further enhancements in performance can be achieved through fine-tuning on the reference datasets.

Dataset	Method	Evaluation Metrics			
		Accuracy	*Precision*	*Recall*	$MacroF1$
COVID-19	scGPT (fine-tuned)	**0.891**	**0.577**	**0.567**	**0.547**
	scGPT (zero-shot)	0.867	0.513	0.476	0.468
	Azimuth	0.878	0.515	0.418	0.444
	expiMap	0.730	0.441	0.325	0.335
Lung-Kim	scGPT (fine-tuned)	**0.974**	0.959	**0.967**	**0.962**
	scGPT (zero-shot)	0.968	**0.970**	0.933	0.948
	Azimuth	0.970	0.957	0.962	0.959
	expiMap	0.920	0.940	0.846	0.878

Supplementary Table 1: Reference Mapping Benchmark Results. scGPT was benchmarked with Azimuth [4] and expiMap [2] on the COVID-19[1] and Lung-Kim[5] datasets for cell type annotation via reference mapping. scGPT fine-tuned and zero-shot settings are described in Supplementary Note 1. We present four classification evaluation metrics *Accuracy*, *Precision*, *Recall*, and $MacroF1$. See metric details in Supplementary Note 12.

2 Benchmarking results on downstream tasks

Dataset	Model	Classification Metrics			
		Accuracy	*Precision*	*Recall*	*MacroF1*
Myeloid	scGPT (fine-tuned)	**0.642**	**0.366**	**0.347**	**0.346**
	scGPT (from-scratch)	0.606	0.304	0.339	0.309
	TOSICA	0.488	0.316	0.276	0.275
	scBert	0.525	0.331	0.323	0.298
Multiple Sclerosis	scGPT (fine-tuned)	**0.856**	**0.729**	**0.720**	**0.703**
	scGPT (from-scratch)	0.798	0.660	0.623	0.600
	TOSICA	0.758	0.664	0.585	0.578
	scBert	0.785	0.604	0.624	0.599
hPancreas	scGPT (fine-tuned)	**0.968**	**0.735**	**0.725**	**0.718**
	scGPT (from-scratch)	0.936	0.665	0.668	0.622
	TOSICA	0.960	0.661	0.681	0.656
	scBert	0.964	0.699	0.689	0.685

Supplementary Table 2: Cell Type Annotation Benchmark Results. scGPT was benchmarked with TOSICA [6] and scBert [7] on the Myeloid (Mye.), Multiple Sclerosis (M.S.), and hPancreas [6] datasets for cell type annotation performance. The scGPT (from-scratch) is a smaller four-layer transformer model trained from random parameter initialization. We present four classification evaluation metrics *Accuracy*, *Precision*, *Recall*, and $MacroF1$. See metric details in Supplementary Note 12.

Dataset	Model	Biological Conservation				Batch Correction			
		AvgBIO	NMI_{cell}	ARI_{cell}	ASW_{cell}	**AvgBATCH**	ASW_{batch}	*GraphConn*	**Overall**
COVID-19 [1]	scGPT (fine-tuned)	**0.504**	0.659	0.400	0.452	**0.850**	0.826	0.874	**0.642**
	scGPT (from-scratch)	0.450	0.602	0.318	0.429	0.744	0.712	0.776	0.567
	scVI [8]	0.502	0.638	0.408	0.461	0.838	0.833	0.844	0.637
	Seurat [9]	0.413	0.513	0.289	0.437	0.790	0.799	0.781	0.564
	Harmony [10]	0.327	0.482	0.185	0.313	0.680	0.642	0.720	0.468
PBMC 10K [11]	scGPT (fine-tuned)	**0.821**	0.850	0.873	0.740	0.923	0.950	0.895	**0.862**
	scGPT (from-scratch)	0.645	0.689	0.614	0.636	0.882	0.911	0.852	0.740
	scVI	0.753	0.819	0.847	0.592	**0.947**	0.967	0.928	0.831
	Seurat	0.724	0.808	0.722	0.641	0.940	0.960	0.920	0.810
	Harmony	0.784	0.860	0.902	0.591	0.940	0.975	0.906	0.846
Perirhinal Cortex [12]	scGPT (fine-tuned)	**0.899**	0.930	0.919	0.848	0.930	0.898	0.964	0.911
	scGPT (from-scratch)	0.889	0.886	0.895	0.886	0.884	0.892	0.875	0.887
	scVI	0.869	0.980	0.990	0.637	0.966	0.939	0.992	0.908
	Seurat	0.878	0.914	0.897	0.822	0.965	0.938	0.992	0.913
	Harmony	0.890	0.960	0.960	0.749	**0.975**	0.957	0.992	**0.924**

Supplementary Table 3: scRNA-seq Integration Benchmark Results. scGPT was benchmarked with scVI [8], Seurat [9], and Harmony [10] on the COVID-19 (18 batches) [1], PBMC 10K (2 batches) [11] and Perirhinal Cortex (2 batches) [12] datasets for cell type clustering and batch correction performance. The scGPT (from-scratch) is a smaller four-layer transformer model trained from random parameter initialization. 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 Note 12

Dataset	Model	Biological Conservation				Batch Correction			Overall
		AvgBIO	NMI_{cell}	ARI_{cell}	ASW_{cell}	**AvgBATCH**	ASW_{batch}	$GraphConn$	
10X Multiome PBMC [13]	scGPT (fine-tuned)	**0.758**	0.807	0.822	0.645	-	-	-	-
(Paired RNA+ATAC)	scGPT (from-scratch)	0.728	0.775	0.759	0.651	-	-	-	-
	scGLUE [14]	0.747	0.815	0.806	0.619	-	-	-	-
	Seurat v4 [4]	0.722	0.784	0.691	0.691	-	-	-	-
BMMC [15]	scGPT (fine-tuned)	**0.697**	0.783	0.725	0.582	**0.871**	0.834	0.908	**0.766**
(Paired RNA+Protein)	scGPT (from-scratch)	0.696	0.777	0.718	0.593	0.822	0.789	0.854	0.746
	Seurat v4 [4]	0.600	0.737	0.470	0.594	0.678	0.685	0.671	0.631
ASAP PBMC [16]	scGPT (fine-tuned)	**0.587**	0.645	0.469	0.648	**0.951**	0.909	0.992	**0.732**
(Mosaic RNA+ATAC+Protein)	scGPT (from-scratch)	0.508	0.549	0.286	0.689	0.941	0.891	0.991	0.681
	scMoMat [17]	0.546	0.448	0.557	0.633	0.916	0.849	0.983	0.667

Supplementary Table 4: scMultiomic Integration Benchmark Results. For the paired 10X Multiome PBMC[13] dataset, scGPT was benchmarked with scGLUE[14] and Seurat v4[4] for cell type clustering performance evaluated on four biological conservation metrics. The data has only one technical batch. Batch correction metrics are not applicable to this setting. For the paired BMMC[15] and mosaic ASAP PBMC[16] datasets, scGPT was benchmarked with Seurat v4 and scMoMat [17] respectively on cell type clustering and multi-omic integration performance.

Common Pathways ($N = 15$)	Unique Pathways from Co-expression ($N = 6$)	Unique Pathways from scGPT ($N = 22$)
APC/C-mediated Degradation Of Cell Cycle Proteins R-HSA-174143	APC/C:Cdc20 Mediated Degradation Of Cyclin B R-HSA-174048	**Adaptive Immune System R-HSA-1280218**
APC/C:Cdc20 Mediated Degradation Of Mitotic Proteins R-HSA-176409	Cell Cycle Checkpoints R-HSA-69620	**Antiviral Mechanism By IFN-stimulated Genes R-HSA-1169410**
Activation Of APC/C And APC/C:Cdc20 Mediated Degradation Of Mitotic Proteins R-HSA-176814	ESR-mediated Signaling R-HSA-893921	**Classical Antibody-Mediated Complement Activation R-HSA-173623**
Cell Cycle R-HSA-1640170	Estrogen-dependent Gene Expression R-HSA-9018519	**Costimulation By CD28 Family R-HSA-388841**
Cell Cycle, Mitotic R-HSA-69278	Resolution Of Sister Chromatid Cohesion R-HSA-2500257	**Creation Of C4 And C2 Activators R-HSA-166786**
Cytokine Signaling In Immune System R-HSA-1280215	Separation Of Sister Chromatids R-HSA-2467813	**Downstream TCR Signaling R-HSA-202424**
Immune System R-HSA-168256		G1/S Transition R-HSA-69206
Interferon Alpha/Beta Signaling R-HSA-909733		G1/S-Specific Transcription R-HSA-69205
Interferon Signaling R-HSA-913531		Generation Of Second Messenger Molecules R-HSA-202433
M Phase R-HSA-68886		**ISG15 Antiviral Mechanism R-HSA-1169408**
Mitotic Anaphase R-HSA-68882		**Initial Triggering Of Complement R-HSA-166663**
Mitotic Metaphase And Anaphase R-HSA-2555396		**Interferon Gamma Signaling R-HSA-877300**
NGF-stimulated Transcription R-HSA-9031628		**MHC Class II Antigen Presentation R-HSA-2132295**
Neutrophil Degranulation R-HSA-6798695		Mitotic G1 Phase And G1/S Transition R-HSA-453279
Phosphorylation Of APC/C R-HSA-176412		Muscle Contraction R-HSA-397014
		Nuclear Events (Kinase And Transcription Factor Activation) R-HSA-198725
		PD-1 Signaling R-HSA-389948
		Phosphorylation Of CD3 And TCR Zeta Chains R-HSA-202427
		RHO GTPases Activate PAKs R-HSA-5627123
		Smooth Muscle Contraction R-HSA-445355
		TCR Signaling R-HSA-202403
		Translocation Of ZAP-70 To Immunological Synapse R-HSA-202430

Supplementary Table 5: Comparison of Common and Unique Pathways Identified by scGPT and the Co-expression Network From the Reactome Database. The enriched pathways from gene programs extracted by both methods at Leiden resolution 40 are listed here for comparison, corresponding to the Venn diagram in Figure 5E. The 14 immune-related pathways uniquely identified by scGPT are highlighted in bold.

Dataset	Model	$\boldsymbol{Pearson_{delta}}$	$\boldsymbol{Pearson_{delta+de}}$
Adamson	scGPT (fine-tuned)	**0.615**	**0.789**
	GEARS	0.531	0.678
	LR	0.387	0.620
Norman	scGPT (fine-tuned)	**0.583**	**0.742**
	GEARS	0.547	0.715
	LR	0.532	0.697
Replogle (All Perturbations)	scGPT (fine-tuned)	**0.242**	**0.464**
	GEARS	0.154	0.277
	LR	0.141	0.298
Replogle (TF Only)	scGPT (fine-tuned)	**0.154**	**0.340**
	GEARS	0.091	0.241
	LR	0.096	0.221

Supplementary Table 6: scGPT's Perturbation Prediction Results on the Adamson, Norman, and Replogle perturbation datasets. scGPT was benchmarked with GEARS [18] and the Linear Regression (LR) model on the test perturbations in each dataset. In the Replogle dataset, we further evaluated all methods on a TF-specific subset of test perturbations. We present two correlation evaluation metrics $Pearson_{delta}$, and $Pearson_{delta+de}$. See metric details in Supplementary Note 12.

3 Enhancing speed and decreasing memory usage in fine-tunings

In an attempt to hasten the fine-tuning phase and render scGPT more accessible for users, we explored a variety of fine-tuning methods in the context of the scRNA-seq integration task. The pre-trained weights from the whole-human dataset were employed to initialize our entire model. As the standard practice, the baseline Full fine-tune procedure involves gradient updates for all model parameters and includes all zero-expressed genes. We undertook experiments that involved freezing the embedding layers and omitting zero-expressed genes during fine-tuning. From data presented in Supplementary Table 7, we noticed a significant drop in the training time per epoch as well as in GPU Memory usage throughout the fine-tuning process. Specifically, the removal of zero-expressed genes resulted in a substantial reduction in our model's maximum sequence length to approximately $40 \sim 60\%$ of its initial length, effectively halving both the time spent on training epochs and peak GPU Memory utilization. Furthermore, by freezing the embedding layer, we achieved an additional reduction in Peak GPU Memory usage by 1GB, a slight increase in training speed, while preserving a comparable AvgBio score.

Dataset	Fine-tuning Option	Fine-tuning Metrics		**AvgBio**
		Avg Epoch Time(s)	Peak GPU Mem(G)	
COVID-19 [1]	Default	93.89	18.784	**0.504**
	Accelerated	**28.84**	**7.848**	0.473
PBMC 10K [11]	Default	56.91	18.248	0.821
	Accelerated	**26.40**	**8.088**	**0.828**
Perirhinal Cortex [12]	Default	81.00	18.248	**0.899**
	Accelerated	**37.22**	**7.816**	**0.899**

Supplementary Table 7: Benchmarking Results for scRNA-seq Integration Speed Enhancement Techniques. scGPT was evaluated based on different fine-tuning strategies on four datasets: COVID-19 (18 batches) [1], PBMC 10K (2 batches) [11], and Perirhinal Cortex (2 batches) [12]. These evaluations were performed to assess cell type clustering and batch correction performance. This table encapsulates two system metrics: the average duration of training per epoch and the maximum GPU memory usage on an A100 GPU. The resulting *AvgBio* score is also presented for performance comparisons.

4 Influence of pre-training data size

To investigate the effect of pre-training data size on scGPT's performance, we pre-trained the scGPT model on smaller scales of the normal human cells, i.e. 30K, 300K and 3M cells. The resultant models were evaluated in both *zero-shot* and *fine-tuned* configurations across multiple tasks. Supplementary Figure 13 reveals that, with growth in pre-training data size, scGPT has increasing performance for both zero-shot and fine-tuning scenarios.

We demonstrate the influence of pre-training data size spanning various tasks of cell type annotation, reference mapping, and perturbation response prediction. Constant trends of increasing fine-tuning performance were observed across all datasets (Supplementary Figure 13A-C). On cell type annotation (utilizing softmax classifiers) and reference mapping, we observed significant increases regarding all evaluation metrics. On the other hand, the performance gain on perturbation is marginal. From the 30k data pre-training to 33M, the result fine-tuning differences are less than 7%, exemplified by the $Pearson_{delta}$ improvement from 0.229 to 0.242. One of the reasons can be the Perturb-seq data sparsity, which we further depicted in Supplementary Note 10.

Next, we examined the performance of zero-shot settings for reference mapping and integration. For these two tasks, the cell embeddings computed from the pre-trained scGPT can be used to compute either the nearest neighbors for propagating reference labels, or for integrating similar cells across samples. We visualize the performances when using different pre-trained models in Supplementary Figure 13D,E. Again, we observed consistently increasing trends across tasks and datasets. Notably, the most pronounced performance leap was observed when transitioning from the 300K to the 3M pre-training data scale.

Overall, the positive correlation between pretraining data size and the performances on downstream performances showcases the efficacy of pre-training. Pre-trained models will continue to expand their capabilities, facilitating breakthroughs in single-cell research and unlocking new discoveries in cellular biology. We also acknowledge the importance of considering model size alongside data size. Previous research has shown a correlation between optimal model size and data size, as discussed [19]. While some scaling trend curves (e.g. annotation and reference mapping) in Supplementary Figure 13A may appear to reach a saturation point after the 3M data size, we hypothesize that increasing the model size would lead to improved performance even at 33M data size and beyond. We leave the exploration of varying model sizes to future work.

5 Context-specific pre-training and its influence on integration performance

We pre-trained scGPT on cell atlases including all human cell types, envisioning that it contributes to specific downstream applications primarily via fine-tuning. Given that downstream applications typically focus on a limited subset of tissues or cell types, we are prompted to consider whether it would be more beneficial to enhance performance for a specific fine-tuning task through context-specific pre-training. To be specific, for the downstream fine-tuning on specific *contexts* of certain cell types, we are interested in comparing the contributions of three different pre-trained models: (1) scGPT (whole-human), which is the model pre-trained generally on all human cell types; (2) in-context models, which during pre-training have seen similar cell types as in the downstream applications; (3) out-of-context models, which are pre-trained primarily on different tissues or cell types. To conduct this comparison, we tested a range of pre-trained models on the scRNA-seq data integration task. These pre-trained models were originally trained using distinct tissue-specific datasets, including lung, blood, heart, kidney, brain, pancreas, intestine, and an all-inclusive, whole-human dataset. In particular, we subsampled from the original whole-human dataset to generate a dataset of 13.2 million cells, aligning it in size with the blood (10.3 million cells) and brain (13.2 million cells) datasets for a more direct comparison of the impact of context. We employed each of the eight pre-trained models to perform data integration on the COVID-19 dataset [1], followed by a comparative performance analysis. Given its diverse cellular composition, including Lung, PBMC, and Bone Marrow cells, this dataset provides an ideal platform for investigating the effects of employing pre-trained models from different cellular contexts.

Our analysis produced notable results, revealing a clear correlation between the relevance of the model context used in the pre-training initiatives and their subsequent performance on the COVID-19 dataset. Supplementary Figure 8 visualizes these findings, showcasing the average $AvgBIO$ score along with the standard error derived from five integration experiments for each pre-trained model (Panel A). Moreover, the UMAP visualization (Panel B) presents an in-depth view of the cell embeddings colored by cell types, substantiating the quality of the learned representations and visually validating the models' integration performance. Notably, the top performers in this analysis were models pre-trained on whole-human, blood, and lung datasets, which correspond closely to the cell types present in the COVID-19 dataset.

In particular, even though the brain pre-trained model was trained on a substantial dataset of 13 million cells, it trailed in performance by 8% compared to the blood pre-trained model with a similar dataset size. This gap in performance sheds light on the importance of cellular context relevance. Specifically, the cellular context of the blood model aligns more closely with the COVID-19 dataset, which includes immune cells, bone marrow cells, lung cells, and PBMCs. Thus, it becomes evident that the alignment of the cellular context in the pre-training phase plays a critical role in achieving superior results for downstream data integration tasks, even when datasets of similar sizes are compared. In light of our findings, the whole-human pre-trained model, embodying a vast spectrum of cell types, consistently demonstrates robust performance across diverse analyses.

Our investigation underlined the significance of cellular context in single-cell RNA-seq data integration tasks. In certain circumstances, when the cellular context of the target dataset aligns with the tissue-specific pre-trained models, these models can excel. Overall, while it is essential to consider the cellular context, the whole-human pre-trained model emerges as a versatile and reliable option for a wide range of applications.

6 Choice of the number of bins in data processing

The binning strategy we introduced in Methods - Input embeddings reshaped the distribution of gene expressions across samples to a unified scale and thus served as an immediate step to partially remove technical batch effects at the preprocessing stage. This is especially crucial for effective pre-training and subsequent knowledge transfer during fine-tuning.

The number of bins controls the granularity of binned expression values. A trade-off exists in the selection of the bin count. Fewer bins may result in the loss of granular detail in the data but can be advantageous for mitigating batch effects. Conversely, a larger number of bins could reveal more nuanced variances but may also preserve high levels of noise and sequencing batch effects.

To verify the effectiveness of binning, we conducted experiments with varying settings, testing whether or not to use binned values and different numbers of bins. The experiments are conducted both in fine-tuning and from-scratch settings on the PBMC 10K [11] dataset for integration, and we examined the influence on task performance (Supplementary Figure 10). In both settings, we tested the number of bins in a large range from 3 to 2001, and found the proper settings of bins in the range of tens to a few hundred consistently lead to better performance. Particularly, the fine-tuning performance reaches optimal when the number of bins is around 51, which is also the setting we used for pretraining. This finding indicates that maintaining similar bins between pretraining and fine-tuning potentially eases the process of transfer learning. Lastly, we found utilizing the binning strategy outperforms the setting without binning for both fine-tuning and from-scratch training. Overall, the experiments empirically demonstrate that the binning approach serves as an effective method for data preprocessing for training single-cell foundation models.

7 Attention-based identification of gene interactions for functionally related genes

To demonstrate the interactions of functionally related genes in perturbation conditions, we performed an attention-based analysis (Methods) on two groups of transcription factors (TFs) that have been perturbed by Replogle et al. [20]. TFs within the group were reported to trigger similar genetic responses, thus annotated by Replogle et al. [20] as mRNA polyadenylation and histone

acetylation, respectively. The mRNA polyadenylation group includes TFs CPSF2, CPSF3, CPSF4, and CSTF3. The histone acetylation group includes TFs KAT8, MCRS1, and YEATS4. In our experiment, we fine-tuned scGPT on the Replogle dataset, and tested its ability to recapitulate these gene relations.

In both cases, scGPT identified putative gene targets validated in the CHIP-Atlas databases using the top 20 most influenced genes (Supplementary Figure 9). Common gene targets are shared among TFs in the same functional group. In the Reactome pathway analysis, with the predicted gene interactions for CPSF2, CPSF3, CPSF4, and CSTF3, scGPT identified mRNA 3-End Processing, the exact process of mRNA polyadenylation as annotated by Replogle et al. [20]. Moreover, 8/12 of these enriched pathways include one or more of these TFs, highlighting processes directly regulated by these genes (Supplementary Figure 9A). Enriched by the predicted genes associated with KAT8, MCRS1, and YEATS4, scGPT is able to capture a multitude of biological processes regulated by histone acetylation, including DNA replication [21, 22, 23, 24, 25], DNA repair [21, 23], and cell cycle regulation [21, 25] (Supplementary Figure 9B). These results validated scGPT's ability to uncover the gene interactions for specific processes defined by functionally related transcription factors, along with their downstream effects.

8 Perturbation expression prediction with KnockTF evaluation

We extended our perturbation prediction evaluation on the Replogle dataset [20] by comparing further with the Transcription Factor (TF) expression profiles of bulk perturbation experiments from the KnockTF database [26]. Specifically, we used the model trained on the Perturb-seq Replogle dataset to predict the perturbation response and test the alignment of the results with the records in the KnockTF. This represents a unique challenge stemming from differences between training and inference data. The scGPT model was originally trained on single-cell data and evaluated on its ability to generalize to unseen bulk expression inputs and predictions.

The KnockTF database contains bulk gene expression profiles for 72 TFs that used the matched cell type of K562 cell line and CRISPRi intervention methodology as in Replogle dataset. These profiles contain the pre- and post-perturbation mean expressions. After removing TFs with missing recorded gene expression, we retained 63 TF perturbations to be tested. These 63 test TFs can be split into two groups: 1) The perturbation TFs that have been included in the gene expression profiles of Replogle dataset (n=40), and 2) perturbations of totally unseen TFs (n=23). For each TF perturbation, we obtained a predicted bulk expression profile from the scGPT model by inputting the bulk gene expression record pre-perturbation and the information of the perturbed TF. We then compared the predicted expressions with the corresponding ground-truth post-perturb bulk expression profiles from KnockTF. Since the gene expression values are on different scales from single-cell to bulk experiments, we reported $Spearman_{delta}$ correlation scores on all gene expression changes, as well as the 20 highest-ranked DE genes for each TF perturbation condition. We benchmarked with the GEARS model that was trained on Replogle dataset and tested in the same settings as above.

In Supplementary Table 8, scGPT showcased competitive performance in predicting gene expression changes captured by the $Spearman_{delta}$ and $Spearman_{delta+de}$ scores. This endorsed scGPT's ability to encode important changes in the gene network elicited by perturbations and to generalize to unseen datasets. GEARS on the other hand, tend to predict small expression changes post-perturbation similar to the predicted values on single-cell data that fail to generalize to the bulk gene expressions. Furthermore, scGPT offers the unique capability of predicting perturbation results for TFs that are not included in gene expression profiles originally (i.e. the unseen group), which is primarily thanks to the advantage of the transformer architecture, so the new gene tokens

of the unseen TFs can be easily appended to the input.

Test Group	Model	Correlation Metrics	
		$Spearman_{delta}$	$Spearman_{delta+de}$
TFs seen in Replogle dataset	scGPT (fine-tuned)	**0.340**	**0.310**
	GEARS	-0.148	-0.270
TFs unseen in Replogle dataset	scGPT (fine-tuned)	**0.417**	**0.256**
	GEARS	-	-

Supplementary Table 8: scGPT's Perturbation Prediction Results Validated with KnockTF [26]. scGPT fine-tuned on the Replogle dataset was benchmarked against GEARS [18] by comparing with bulk perturbation expression profiles from the KnockTF database. We present correlation evaluation metrics $Spearman_{delta}$, and $Spearman_{delta+de}$. Notably, GEARS does not allow addition of new genes at inference time and thus do not support the prediction on the unseen group. See metric details in Supplementary Note 12.

9 Revisiting evaluation metrics for perturbation response prediction

The field of perturbation response prediction represents an emergent yet highly promising application. Existing studies commonly employ correlation-based metrics for evaluation. Specifically, correlations between predicted post-perturbation gene expression values as well as gene expression changes ("delta"), have been reported in existing methods of GEARs [18] and CPA [27].

In this work, we specifically examined these correlational metrics and opted for an evaluation paradigm focused on expression changes ("delta"): We found that the evaluations by $Pearson_{delta}$ and its variants, as reported in Supplementary Table 6, can better capture the difference between methods and faithfully indicate the prediction quality. Contrary to this, we delineate the following reasons to argue that metrics based on raw expression levels may introduce bias:

(1) Bias Toward Natural Transcriptional Variance: Utilizing Pearson correlation metrics on expression levels introduces bias as it predominantly captures natural variances in transcriptional abundance across genes. For instance, housekeeping genes, which are sufficiently expressed across varying conditions, can contribute to inflated correlation scores. To demonstrate this phenomenon, we report a naive baseline that employs the mean expression level of control cells as the prediction for perturbation responses. Remarkably, despite its inability to predict expression changes, this baseline yielded high $Pearson$ scores across all evaluated datasets, as showcased in Supplementary Table 9. This pinpoints the issue that the correlation on raw expression is mostly evaluating the background expression variances among genes, rather than the differences specific to the perturbed conditions.

(2) Inflation of Performance Metrics: Metrics based on raw expression levels frequently present inflated scores, often surpassing the 0.95 threshold, as indicated in Supplementary Table 9. This overestimation can be misleading, as it may imply that the challenge of predicting perturbation responses has been adequately addressed. In contrast, as reported in Supplementary Table 6, metrics on the delta-change reveal that the predictions are far from perfect, especially on the most challenging genome-wide Perturb-seq Replogle dataset. Although the comparative trends among different methodologies remain consistent under both metric categories, delta-change metrics provide a more stringent and informative evaluation.

In summary, our findings advocate for the adoption of delta-change metrics in evaluating per-

turbation response predictions. These metrics not only circumvent the biases inherent in raw-expression-based evaluations but also offer a more granular understanding of performance differences among competing methodologies.

Dataset	Model	$Pearson$	$Pearson_{de}$
Adamson	scGPT (fine-tuned)	0.990	0.976
	GEARS	0.984	0.966
	LR	0.987	0.939
	Control	0.987	0.919
Norman	scGPT (fine-tuned)	0.989	0.940
	GEARS	0.986	0.940
	LR	0.988	0.930
	Control	0.981	0.850
Replogle (All Perturbations)	scGPT (fine-tuned)	0.991	0.962
	GEARS	0.989	0.958
	LR	0.988	0.958
	Control	0.991	0.956
Replogle (TF Only)	scGPT (fine-tuned)	0.993	0.970
	GEARS	0.992	0.961
	LR	0.990	0.964
	Control	0.994	0.956

Supplementary Table 9: Evidence for the inadequacy of $Pearson$ correlation between expression abundance as an evaluation metric for perturbation response prediction. The naive baseline method of using mean control expression as prediction can generate high correlations on all datasets, although the method has no ability to predict gene expression change post-perturbation. This indicates that the correlation of expression magnitude can be inflated by the natural difference between genes, and thus it is a reliable indicator to evaluate perturbation response predictions.

10 Discussion about future prospects and limitations

We have presented scGPT as a promising attempt to build single-cell foundation models. Although at a relatively early stage, we expect foundation models can leverage the knowledge from large-scale cell atlases and transfer them for contribution to the various analyses for specific research questions on custom data, potentially providing new biological insights. In this section, we identify some of the important prospects and limitations for this direction based on our observations and considerations.

Zero-shot applications In Supplementary Notes 1 and 4, we examined the zero-shot applications for data integration and reference mapping. The pre-trained scGPT model demonstrates considerable abilities for these tasks off-the-shell. Shown in Supplementary Table 1, the zero-shot scGPT model displays noteworthy performance that is on par with other specialized methods. Shown in Supplementary Figure 12, zero-shot embeddings provide better integration performance than principal component-based approaches for highly variable genes. The advantage is demonstrated by the higher AvgBIO scores on two out of three datasets and AvgBATCH scores on all three datasets. On the COVID-19 dataset, zero-shot performance closely approximates that of the fine-tuned model and surpasses both Seurat V3 and Harmony, as evidenced by comparisons between Supplementary Figures 12 and 6A. In the context of the Lung-Kim dataset, the zero-shot embeddings distinctly segregate NK cell populations from T cells within the UMAP visualization. Moreover, within the Multiple Sclerosis dataset, zero-shot embeddings effectively isolate neurons of cortical layers 5-6 from other layers, which is not accomplished by utilizing the UMAP of highly variable genes (HVGs). Based on these findings, we believe the pre-training introduced in this work demonstrates promising embeddings that align with biological features. With the growing model capacity and pre-training data diversity in future development, we envision the zero-shot embeddings from single-cell foundation models can serve as a fast and context-rich method for various applications such as visualizing, clustering, and reference mapping, etc.

While the demonstrated zero-shot capabilities of our model are indeed promising, we find it imperative to highlight two potential limitations: (1) The range and proportions of cell populations present in the pre-training dataset could limit the efficacy of zero-shot applications for underrepresented cell types. As depicted in Figure 1D,E, cell atlases are commonly populated with specific cell types, such as neural and immune cells. Rarely occurring cell types or those from unique conditions are correspondingly underrepresented, which could curtail performance in subsequent, similar analytical contexts. We anticipate that future pre-training with more diverse and balanced cell types may ameliorate this limitation. (2) The pre-training protocol in its current form does not explicitly account for the removal of batch effects. Consequently, zero-shot embeddings generated directly from the pre-trained model may retain "consistent" variations introduced by differing sequencing protocols or devices. The following paragraph delves into this subject in greater detail.

Pre-training design and its implications for batch correction The current pre-training is a purely data-driven and self-supervised procedure that optimizes the model to reconstruct gene expressions in sequenced samples. This procedure offers the benefits of avoiding potential human bias in the meta-labels and treating all raw signals in the data "equally". However, it inherently omits specific requirements that may be critical for downstream analyses. We draw particular attention to the fact that our pre-training strategy does not inherently include the mitigation of the technical variations introduced by sequencing protocols and devices.

In single-cell data analysis, technical batch effects indicate data distribution shifts that are driven by the sequencing procedure, rather than biological variations [28]. The mitigation of such

effects is instrumental for more accurate visualization and clustering of cell populations across multiple samples. Numerous elements contribute to the introduction of batch effects in the collation of cell atlases, including but not limited to, capturing times, personnel involved, reagent batches, types of equipment, sequencing protocols, and technological platforms [29]. Considering the pretraining process reconstructs binned gene expressions from other observed genes, we can classify these factors into two broad types based on their influence on pretraining: (1) The type I factors are the ones that may largely have global effects on the capturing of all gene expression, such as capturing times or sequencing depth. The pre-training may mitigate these factors through repeated training on large data examples. (2) The type II factors have gene-specific influences. As reported by existing literature [29], some of the technical batch effects (i.e. sequencing protocols and machines) can heavily shape the gene expression distributions. Rather than being ameliorated, these effects could be faithfully encapsulated during the pre-training process. From the standpoint of our pre-training objectives, these factors are not considered noise but important signals to the gene expression reconstruction task.

We would argue this limitation on the mitigating of type II technical effects is a necessary trade-off for maintaining an "unbiased" pre-training. While it may be plausible to incorporate a batch effect mitigation objective within the pre-training scheme, the added objective could compromise the original pre-training goals aimed at reconstructing biological gene expression because of the complex, often non-linear relationships between technical and biological factors. Consequently, we advocate maintaining an "unbiased" data-driven approach during pre-training, thereby offering the flexibility to prioritize specific analytical objectives in subsequent fine-tuning stages.

Challenge of evaluation The evaluation for single-cell foundation models can be extremely challenging, primarily due to the absence of definitive "ground truth" for a range of application tasks. Unlike in Natural Language Processing, where human judgment often serves as a reliable standard, biological data lack an unequivocal benchmark for performance evaluation. Furthermore, obtaining or validating expert annotations is both time-consuming and resource-intensive.

One specific challenge we encountered was evaluating the model performance on tasks that require human annotations as labels, such as data integration and cell type identification. The original annotations for cell types were usually derived from clustering analyses and linear separations based on marker gene expressions. Therefore, when evaluating the performance of models by matching with human annotations, linear and simple models may be favored in the assessment. This phenomenon has also been observed in the benchmarking study for cell type identification [30], where conventional methods demonstrated superior performance than existing specialized approaches for single-cell analysis.

Expression prediction tasks such as perturbation response prediction offer potentially more authentic evaluations without the need for human annotations. However, the issue of data quality becomes prominent. In our analysis of three Perturb-seq datasets used for perturbation response prediction, we found the ratios of uncaptured gene expression to be notably high: 84% for the Adamson dataset, 92% for the Norman dataset, and 66% for the Replogle dataset. In the case of the Norman dataset, the extreme data sparsity not only reduced the preservation of genuine biological signals but also complicated both the training and evaluation processes. Conversely, the more recent Replogle dataset, with a higher capturing rate, provided a more rigorous test of methodological differences and revealed the intrinsic challenges of the prediction task, as indicated in Supplementary Table 6. Moving forward, we anticipate that the continual improvement in the quality and scale of available datasets will enhance the utility of expression prediction tasks for evaluating single-cell foundation models.

11 Comparison to existing approaches

Transformers for modelling single-cell sequencing data. Transformer models employing self-attention mechanisms [31] have demonstrated remarkable success in the field of natural language processing (NLP) [32], computer vision [33], and protein folding prediction [34]. However, there have been few efforts to incorporate the transformer architecture into single-cell biology and its related applications. Shen et al. [35] utilize a transformer decoder setup to learn the sequence of the names of highly expressed genes, but they do not take into account the actual sequenced expression abundance, resulting in the loss of crucial biological information. scBERT [7] and TOSICA [6] used BERT-like architectures [32] to train cell embeddings but only applied the model on the supervised task of cell annotation. Similarly, Connell et al. [36] utilized transformer encoders mainly to predict genetic perturbation responses. To our best knowledge, scGPT is one of the first methods to provide a generative pre-trained transformer foundation model for multiple single-cell analysis tasks.

Pre-training on large cell atlases Although the idea of employing pre-training and fine-tuning on a wide range of downstream tasks as a unified framework remains largely unexplored, several works have attempted to use transfer learning on specific tasks. Enformer [37] pre-trained a transformer model on 60 thousand human and mouse DNA sequences, and fine-tuned on experimental data for variant effect predictions on gene expressions and inference on enhancer–promoter interactions. scArches [1] devised a transfer-learning-based approach for reference mapping by pre-training a conditional variational autoencoder (VAE) on the reference datasets. However, the scale of the reference datasets remains limited, and the VAE-based architecture does not incorporate attention computation. On the other hand, scBERT [7] is pre-trained on 1 million cells with a BERT architecture. However, the downstream application focuses on cell type annotation only, thus restricting the generalizability of the pre-training and fine-tuning strategy. A recent work, Geneformer [38], has extended pre-trained transformers beyond cell type annotation to gene network analysis and in-silico probing of cell states, but the work didn't demonstrate abilities for perturbation response prediction or multi-omic integration. scGPT has compiled an unprecedented scale of pre-training data and evaluated on a diverse range of downstream tasks, which presents the pre-training and fine-tuning strategy as a unified framework for versatile single-cell analysis.

Learning cell and gene representation for single-cell omic data on downstream tasks. Cell representation learning facilitates a variety of downstream tasks such as cell type annotation, multi-omic integration, and perturbation prediction. A popular framework Seurat [9, 39] employs nearest-neighbor-based alignment to remove batch effect via linear transformation in the embedding space. LIGER [40] and OCAT [41] use matrix factorization to extract latent cell embeddings. Recently, Deep Learning methods and especially VAE-based generative models have gained increasing popularity, as they generate deep embeddings via non-linear transformations through neural networks [42]. scVI [8] learns latent cell representations by reconstructing original gene expression via variational inference. TotalVI [43], scGen [44], CPA [27] and MultiCPA [45] utilized similar models and extended the application to multi-omics and perturbation prediction. For scATAC-seq data, due to the high dimensionality and sparsity of the features space, scBasset [46] leveraged convolutional neural networks to extract compact cell representations. On the other hand, gene representation learning also supports many downstream tasks including gene regulatory network and functional pathway analysis. As an example, GeneVector [47] detects gene-gene functional relations by factorizing the co-expression and mutual information matrix of the sequencing readout. VEGA [48] utilizes a sparse VAE architecture to encode gene network activities for added interpretability. DeepMAPS [49] utilizes graph neural networks to encode cell and gene nodes for related tasks. Extending to multi-omic analyses, SIMBA [50] jointly embeds cell and feature rep-

resentations from scRNA-seq, scATAC-seq and DNA sequences into shared latent space through a graph embedding approach. CellSpace [51] also employed graph embedding techniques to project scATAC-seq profiles of cells and k-mers in DNA sequences into the same space. Despite the importance of the two branches of research in cell and gene embedding learning, few approaches have worked on jointly learning both. scGPT stands out as an approach to effectively learn both embeddings of cells and genes jointly in a shared architecture.

12 Evaluation Metric Calculations

12.1 Cell Type Assignment

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

The *Accuracy*, *Precision* and *Recall* scores are calculated as follows:

$$Accuracy = \frac{\sum_{c \in C} tp_c}{\sum_{c \in C} N_c}, \quad Precision = \frac{1}{|C|} \sum_{c \in C} \frac{tp_c}{tp_c + fp_c}, \quad Recall = \frac{1}{|C|} \sum_{c \in C} \frac{tp_c}{tp_c + fn_c} \quad .$$

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

$$MacroF1 = \frac{1}{|C|} \sum_{c \in C} F1_c, \quad \text{where} \quad F1_c = \frac{2 \times Precision_c \times Recall_c}{Precision_c + Recall_c}.$$

The above metrics are calculated using `scikit-learn`'s implementations [52].

12.2 Perturbation Expression Prediction

We employed the correlation-based metrics to assess perturbation expression prediction performance, akin to the approach by Roohani et al. [18]. The original Pearson correlation metrics were calculated on average expression profiles between ground-truth and prediction per perturbation condition. These Pearson scores were then averaged across conditions as the final $Pearson$ metric. We also reported variations of this metric including $Pearson_{delta}$, $Pearson_{de}$, and $Pearson_{delta+de}$, focusing on the change of expression profiles post-perturbation and/or differentially expressed genes across conditions respectively.

For each perturbation k, given the predicted gene expression matrix X^k_{pred} containing N_{pred} cells and M genes, the average predicted profile $\boldsymbol{x}^k_{pred}$ is calculated as:

$$\boldsymbol{x}^k_{pred,m} = \frac{\sum_{i=0}^{N_{pred}} X_{i,m}}{N_{pred}},$$

where the value at gene m is the mean expression value of that gene across all predictions. The dimension of the average predicted gene expression profile $\boldsymbol{x}^k_{pred}$ is $1 \times M$. Similarly, we calculate the average gene expression profile for target expressions denoted as $\boldsymbol{x}^k_{truth}$, and that for control cells denoted as $\boldsymbol{x}_{control}$.

The Pearson correlation score for perturbation condition k is calculated as:

$$Pearson^k = \frac{\sum_{m=0}^{M}(\boldsymbol{x}^k_{pred,m} - \bar{\boldsymbol{x}}^k_{pred})(\boldsymbol{x}^k_{target,m} - \bar{\boldsymbol{x}}^k_{target})}{\sqrt{\sum_{m=0}^{M}(\boldsymbol{x}^k_{pred,m} - \bar{\boldsymbol{x}}^k_{pred})^2 \sum_{m=0}^{M}(\boldsymbol{x}^k_{target,m} - \bar{\boldsymbol{x}}^k_{target})^2}}.$$

The final $Pearson$ score is the average across K perturbation conditions:

$$Pearson = \frac{\sum_{k=0}^{K} Pearson^k}{K}.$$

In the $Pearson_{delta}$ variation, we calculated the correlation scores between $\boldsymbol{x}^k_{pred} - \boldsymbol{x}_{control}$ and $\boldsymbol{x}^k_{truth} - \boldsymbol{x}_{control}$ instead of $\boldsymbol{x}^k_{pred}$ and $\boldsymbol{x}^k_{truth}$, to highlight the post-perturbational changes in the gene expression profiles. In the $Pearson_{de}$ variation, we calculated the correlation scores on the top 20 most differentially expression genes for each condition. In the $Pearson_{delta+de}$ variation, we applied both of these changes above. For the extended comparison with bulk profiles from KnockTF, we computed the rank-based $Spearman$ correlation scores instead for these variations. The Pearson and Spearman metrics were implemented using `scipy.stats` package.

12.3 Reverse Perturbation - Predicting driving gene perturbations with TopK Retrieval

We employed two modified topK retrieval accuracy metrics to assess the reverse perturbation prediction performance. The hit rate of correct predictions calculates the proportion of test cases where the topK retrieved experiments contain the target (i.e., query) condition. For example, for each test case of target condition X+Y, if the topK retrieved experiments contain X+Y, we count this test case as a hit. We also reported a relaxed topK accuracy metric for relevant retrievals with one-gene overlap. The percentage of test cases with relevant predictions calculates the proportion of test cases where the topK retrieved experiments contain any cases with a one-gene overlap with the target condition. For example, for the same test case X+Y, if the topK retrieved experiments contain X, Y, X+A, A+X, Y+A, or A+Y, we count this test case as a hit. This relaxed metric aims to provide added interpretability for scGPT's choices of retrieval.

12.4 Single-cell integration

We adopted the evaluation metric calculations outlined by Luecken et al. [28] in their benchmark study. Each metric is described below.

Normalized Mutual Information

To quantify the concurrence between the cell type labels based on ground truth and the Louvain cluster labels obtained from integrated cell embeddings, we computed the normalized mutual information (NMI) score. The Louvain clustering was conducted across resolutions ranging from

0.1 to 2, with increments of 0.1. The best score will be selected. The NMI score for cell types, referred to as $\boldsymbol{NMI_{cell}}$, ranges between 0 and 1, where a higher score indicates a better match of cell types.

Adjusted Rand Index

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

Average Silhouette Width

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

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

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

Here, C represents the cell types.

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

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

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

Graph Connectivity

The graph connectivity metric quantifies the average proportion of cells within each cell type that are connected through a kNN (k-nearest neighbors) graph. For every cell identity c in the set C, we compute the size of the largest connected component using kNN among cells exclusively belonging to identity c. This value is divided by the total number of cells with identity c to obtain a normalized measure. The $\boldsymbol{GraphConn}$ score is then reported as the average across all cell types:

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

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

Aggregated Metrics

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

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

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

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

In accordance with the convention established in [28], an ***Overall*** metric is derived as the weighted average of $0.6 * AvgBIO + 0.4 * AvgBATCH$.

References

[1] Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, et al. Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology*, 40 (1):121–130, 2022.

[2] Mohammad Lotfollahi, Sergei Rybakov, Karin Hrovatin, Soroor Hediyeh-Zadeh, Carlos Talavera-López, Alexander V Misharin, and Fabian J Theis. Biologically informed deep learning to query gene programs in single-cell atlases. *Nature Cell Biology*, 25(2):337–350, 2023.

[3] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. scmap: projection of single-cell rna-seq data across data sets. *Nature methods*, 15(5):359–362, 2018.

[4] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck III, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

[5] Nayoung Kim, Hong Kwan Kim, Kyungjong Lee, Yourae Hong, Jong Ho Cho, Jung Won Choi, Jung-Il Lee, Yeon-Lim Suh, Bo Mi Ku, Hye Hyeon Eum, et al. Single-cell rna sequencing demonstrates the molecular and cellular reprogramming of metastatic lung adenocarcinoma. *Nature communications*, 11(1):2285, 2020.

[6] Jiawei Chen, Hao Xu, Wanyu Tao, Zhaoxiong Chen, Yuxuan Zhao, and Jing-Dong J Han. Transformer for one stop interpretable cell type annotation. *Nature Communications*, 14(1): 223, 2023.

[7] Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022.

[8] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

[9] Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502, 2015.

[10] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

[11] Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, et al. A python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology*, 40 (2):163–166, 2022.

[12] Kimberly Siletti, Rebecca Hodge, Alejandro Mossi Albiach, Ka Wai Lee, Song-Lin Ding, Lijuan Hu, Peter Lönnerberg, Trygve Bakken, Tamara Casper, Michael Clark, et al. Transcriptomic diversity of cell types across the adult human brain. *Science*, 382(6667):eadd7046, 2023.

[13] Darren A Cusanovich, Riza Daza, Andrew Adey, Hannah A Pliner, Lena Christiansen, Kevin L Gunderson, Frank J Steemers, Cole Trapnell, and Jay Shendure. Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing. *Science*, 348(6237):910–914, 2015.

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

[15] Malte Luecken, Daniel Burkhardt, Robrecht Cannoodt, Christopher Lance, Aditi Agrawal, Hananeh Aliee, Ann Chen, Louise Deconinck, Angela Detweiler, Alejandro Granados, Shelly Huynh, Laura Isacco, Yang Kim, Dominik Klein, BONY DE KUMAR, Sunil Kuppasani, Heiko Lickert, Aaron McGeever, Joaquin Melgarejo, Honey Mekonen, Maurizio Morri, Michaela Müller, Norma Neff, Sheryl Paul, Bastian Rieck, Kaylie Schneider, Scott Steelman, Michael Sterr, Daniel Treacy, Alexander Tong, Alexandra-Chloe Villani, Guilin Wang, Jia Yan, Ce Zhang, Angela Pisco, Smita Krishnaswamy, Fabian Theis, and Jonathan M Bloom. A sandbox for prediction and integration of dna, rna, and proteins in single cells. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 1:13, 2021.

[16] Eleni P Mimitou, Caleb A Lareau, Kelvin Y Chen, Andre L Zorzetto-Fernandes, Yuhan Hao, Yusuke Takeshima, Wendy Luo, Tse-Shun Huang, Bertrand Z Yeung, Efthymia Papalexi, et al. Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. *Nature biotechnology*, 39(10):1246–1258, 2021.

[17] Ziqi Zhang, Haoran Sun, Ragunathan Mariappan, Xi Chen, Xinyu Chen, Mika S Jain, Mirjana Efremova, Sarah A Teichmann, Vaibhav Rajan, and Xiuwei Zhang. scmomat jointly performs single cell mosaic integration and multi-modal bio-marker detection. *Nature Communications*, 14(1):384, 2023.

[18] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, pages 1–9, 2023.

[19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[20] Joseph M Replogle, Reuben A Saunders, Angela N Pogson, Jeffrey A Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J Wagner, Karen Adelman, Gila Lithwick-Yanai, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

[21] Miglena Koprinarova, Michael Schnekenburger, and Marc Diederich. Role of histone acetylation in cell cycle regulation. *Current topics in medicinal chemistry*, 16(7):732–744, 2016.

[22] Kazuhiro Ito and Ian M Adcock. Histone acetylation and histone deacetylation. *Molecular biotechnology*, 20:99–106, 2002.

[23] Gary Felsenfeld and Mark Groudine. Controlling the double helix. *Nature*, 421(6921):448–453, 2003.

[24] Pablo De Ioannes, Victor A Leon, Zheng Kuang, Miao Wang, Jef D Boeke, Andreas Hochwagen, and Karim-Jean Armache. Structure and function of the orc1 bah-nucleosome complex. *Nature communications*, 10(1):2894, 2019.

[25] Ashwin Unnikrishnan, Philip R Gafken, and Toshio Tsukiyama. Dynamic changes in histone acetylation regulate origins of dna replication. *Nature structural & molecular biology*, 17(4): 430, 2010.

[26] Chenchen Feng, Chao Song, Yuejuan Liu, Fengcui Qian, Yu Gao, Ziyu Ning, Qiuyu Wang, Yong Jiang, Yanyu Li, Meng Li, et al. Knocktf: a comprehensive human gene expression profile database with knockdown/knockout of transcription factors. *Nucleic acids research*, 48(D1):D93–D100, 2020.

[27] Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, page e11517, 2023.

[28] Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19 (1):41–50, 2022.

[29] Hoa Thi Nhu Tran, Kok Siong Ang, Marion Chevrier, Xiaomeng Zhang, Nicole Yee Shin Lee, Michelle Goh, and Jinmiao Chen. A benchmark of batch-effect correction methods for single-cell rna sequencing data. *Genome biology*, 21(1):1–32, 2020.

[30] Tamim Abdelaal, Lieke Michielsen, Davy Cats, Dylan Hoogduin, Hailiang Mei, Marcel JT Reinders, and Ahmed Mahfouz. A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome biology*, 20:1–19, 2019.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

[33] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[34] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[35] Hongru Shen, Xilin Shen, Jiani Hu, Jilei Liu, Chao Zhang, Dan Wu, Mengyao Feng, Meng Yang, Yang Li, Yichen Yang, et al. Generative pretraining from large-scale transcriptomes: Implications for single-cell deciphering and clinical translation. *bioRxiv*, 2022.

[36] William Connell, Umair Khan, and Michael J Keiser. A single-cell gene expression language model. *arXiv preprint arXiv:2210.14330*, 2022.

[37] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

[38] Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, pages 1–9, 2023.

[39] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019.

[40] Jialin Liu, Chao Gao, Joshua Sodicoff, Velina Kozareva, Evan Z Macosko, and Joshua D Welch. Jointly defining cell types from multiple single-cell datasets using liger. *Nature protocols*, 15 (11):3632–3662, 2020.

[41] Chloe X Wang, Lin Zhang, and Bo Wang. One cell at a time (ocat): a unified framework to integrate and analyze single-cell rna-seq data. *Genome biology*, 23(1):1–25, 2022.

[42] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[43] Adam Gayoso, Zoë Steier, Romain Lopez, Jeffrey Regier, Kristopher L Nazor, Aaron Streets, and Nir Yosef. Joint probabilistic modeling of single-cell multi-omic data with totalvi. *Nature methods*, 18(3):272–282, 2021.

[44] Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature methods*, 16(8):715–721, 2019.

[45] Kemal Inecik, Andreas Uhlmann, Mohammad Lotfollahi, and Fabian J Theis. Multicpa: Multimodal compositional perturbation autoencoder. *bioRxiv*, pages 2022–07, 2022.

[46] Han Yuan and David R Kelley. scbasset: sequence-based modeling of single-cell atac-seq using convolutional neural networks. *Nature Methods*, 19(9):1088–1096, 2022.

[47] Nicholas Ceglia, Zachary Sethna, Samuel S Freeman, Florian Uhlitz, Viktoria Bojilova, Nicole Rusk, Bharat Burman, Andrew Chow, Sohrab Salehi, Farhia Kabeer, et al. Identification of transcriptional programs using dense vector representations defined by mutual information with genevector. *Nature Communications*, 14(1):4400, 2023.

[48] Lucas Seninge, Ioannis Anastopoulos, Hongxu Ding, and Joshua Stuart. Vega is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nature communications*, 12(1):1–9, 2021.

[49] Anjun Ma, Xiaoying Wang, Cankun Wang, Jingxian Li, Tong Xiao, Juexing Wang, Yang Li, Yuntao Liu, Yuzhou Chang, Duolin Wang, et al. Deepmaps: Single-cell biological network inference using heterogeneous graph transformer. *bioRxiv*, 2021.

[50] Huidong Chen, Jayoung Ryu, Michael E Vinyard, Adam Lerer, and Luca Pinello. Simba: Single-cell embedding along with features. *Nature Methods*, pages 1–11, 2023.

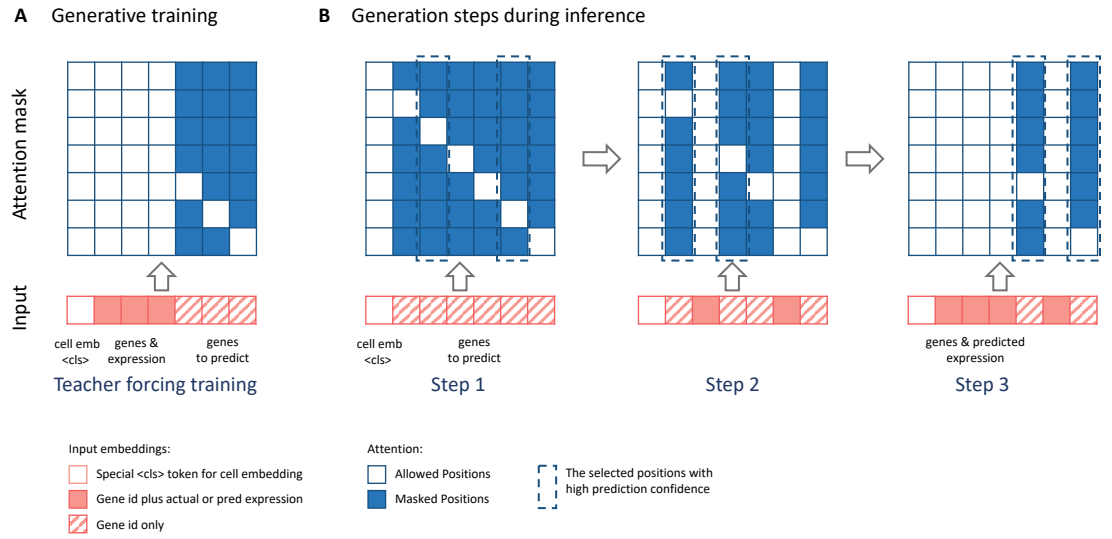
[51] Zakieh Tayyebi, Allison R Pine, and Christina S Leslie. Scalable sequence-informed embedding of single-cell atac-seq data with cellspace. *bioRxiv*, pages 2022–05, 2022.

[52] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. Api design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.

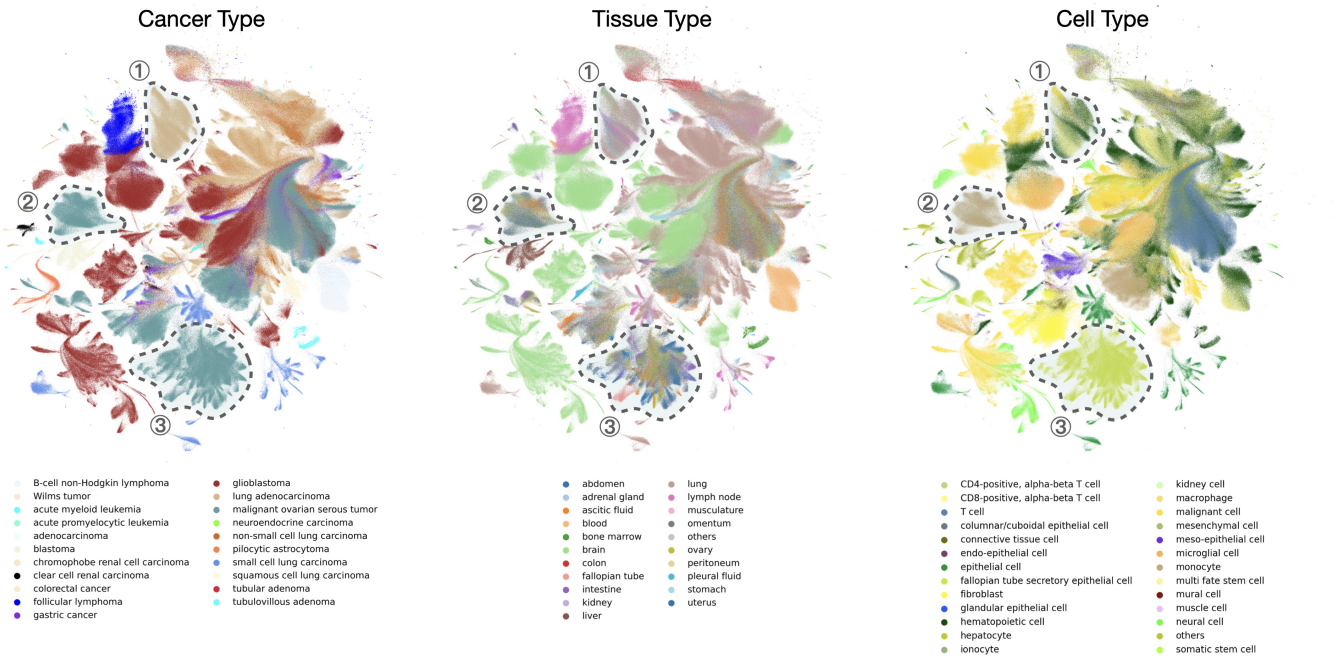
[53] Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

[54] Lucas Schirmer, Dmitry Velmeshev, Staffan Holmqvist, Max Kaufmann, Sebastian Werneburg, Diane Jung, Stephanie Vistnes, John H Stockley, Adam Young, Maike Steindel, et al. Neuronal vulnerability and multilineage diversity in multiple sclerosis. *Nature*, 573(7772): 75–82, 2019.

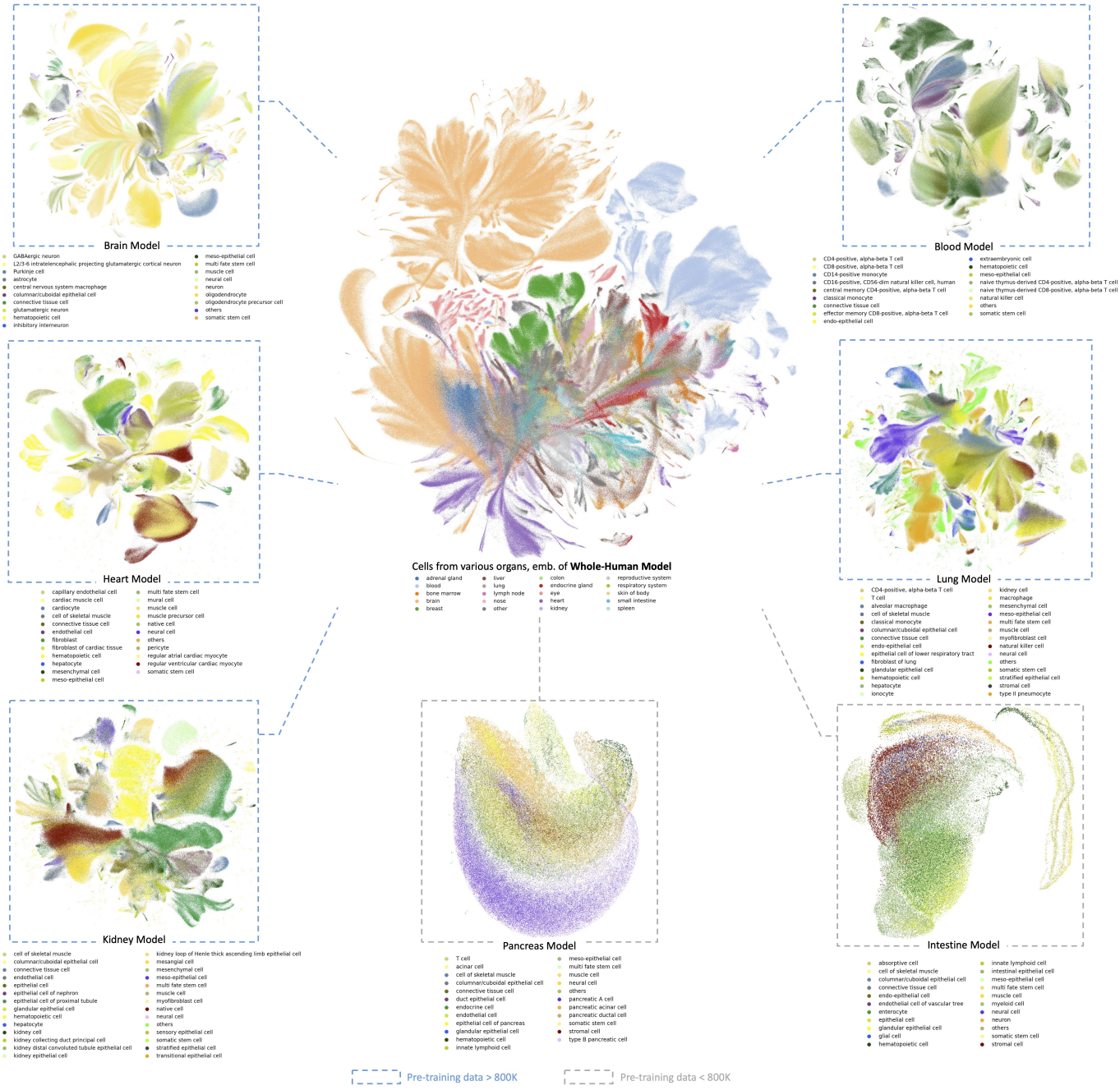
Supplementary Figures



Supplementary Figure 1: The scGPT Attention Mask. The masked positions are colored in blue, and the unmasked positions in white. These masked and unmasked positions correspond to the $M \times M$ attention map for M input tokens. The rows correspond to queries (Q) and columns correspond to keys (K). In the self-attention computation of transformers, the attention scores on the masked positions will be removed. The token identity associated with each column is annotated below, namely "cell emb $< cls >$" for cell embedding, "genes & expression" 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)* After training, the attention mask at each step during the iterative process of scGPT cell-prompt generation.

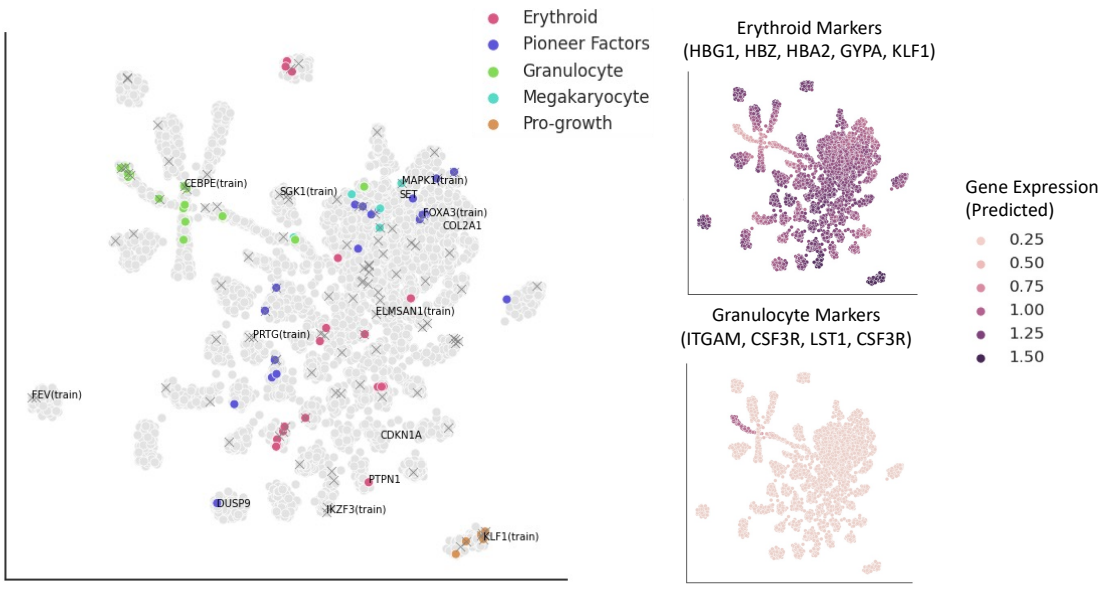


Supplementary Figure 2: UMAP of 3 million cancer cells using the cell embeddings from the pre-trained pan-cancer model. From left to right, the colors indicate the cancer types, tissue types, and cell types. We observed that the model is able to generate cell embeddings revealing the difference in cancer and cell types primarily, exemplified by the outlined three regions.

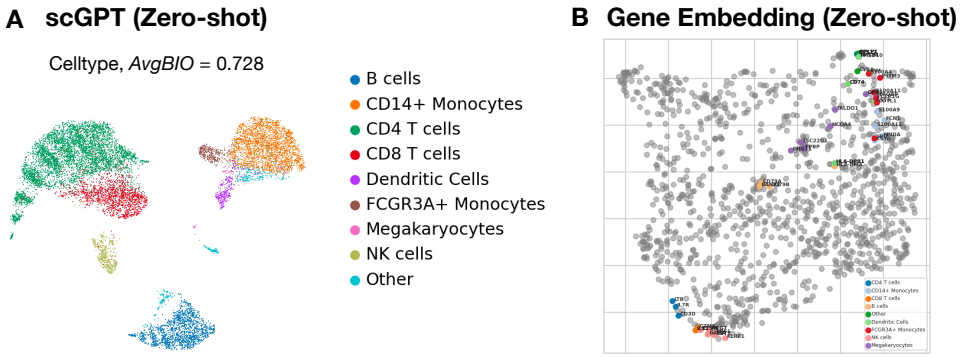


Seven UMAP Plots of Cells from Individual Organs, Using The Cell Emb. of Corresponding Organ-Specific Models

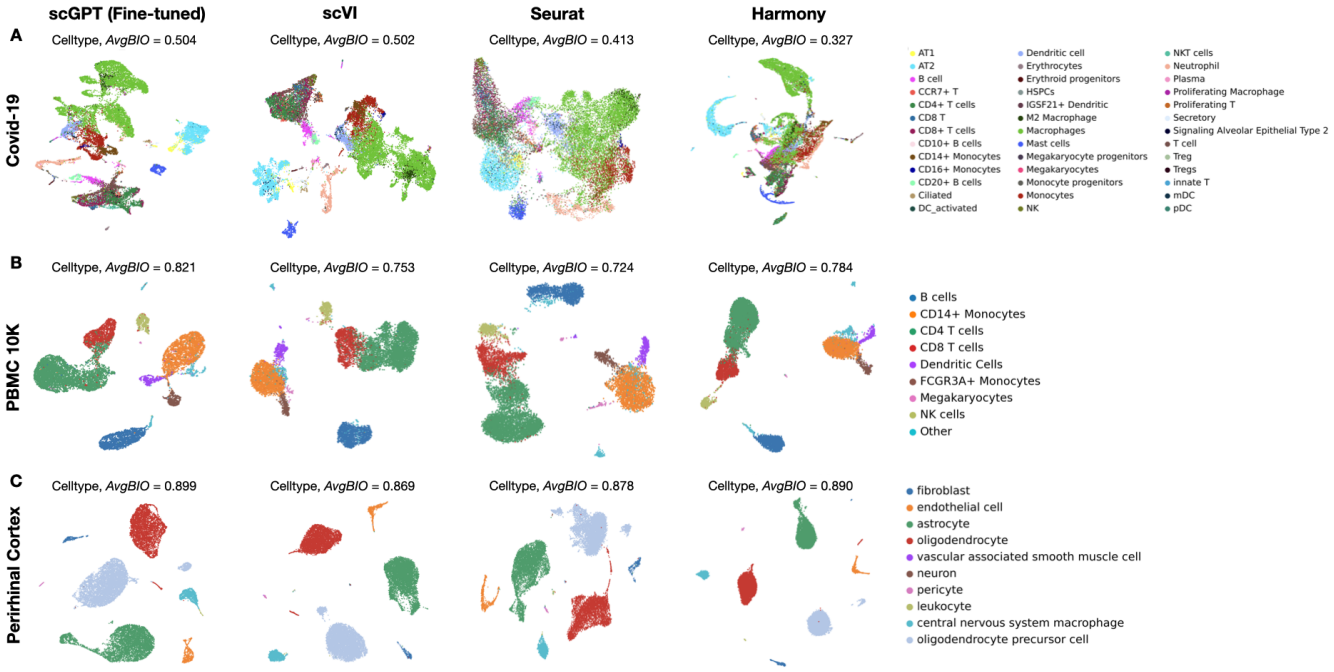
Supplementary Figure 3: Organ-specific models. (*Center*) The UMAP visualization of selected 3 million collected normal human cells using the cell embeddings from the pre-trained scGPT whole-human model. Cells are colored by the organs of origin. (*Around*) The UMAP visualization of cells from each organ using the cell embeddings from the corresponding organ-specific models. The colors in each image indicate major cell types. For example, the top left UMAP visualizes brain cell embeddings from the scGPT model that was specifically pre-trained on brain cells. The outline color of each UMAP plot indicates whether the size of the organ-specific training data is larger than 800,000 cells (blue) or not (grey). We observed that models trained on sufficient data (i.e., $>$ 800,000 cells) could generate decent cell embeddings that can separate major cell types.



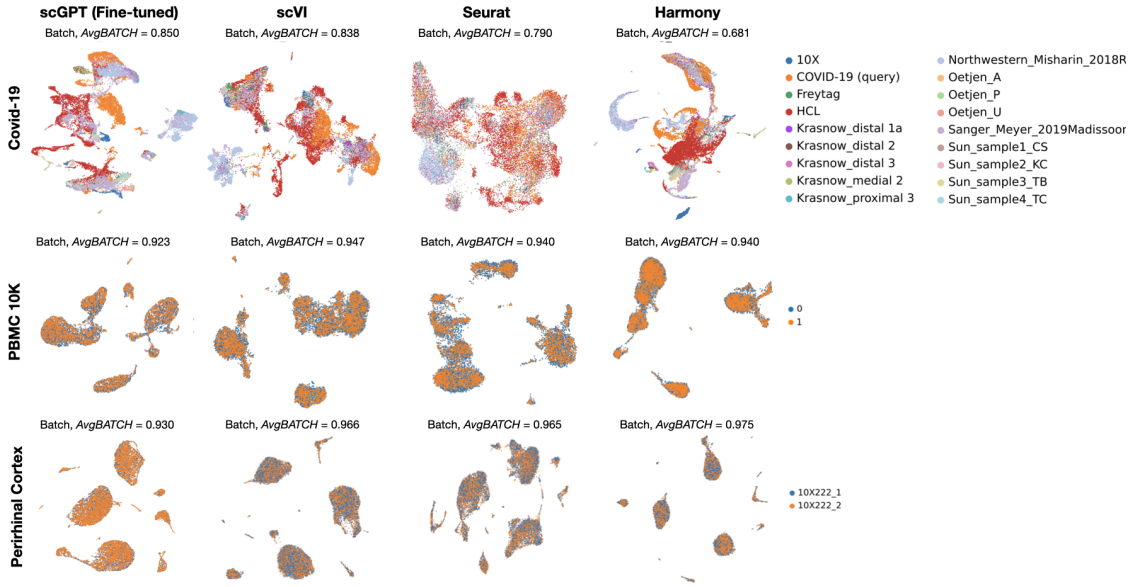
Supplementary Figure 4: Visualization of Annotated Functional Groups by Norman et al. [53]. *(Left)* UMAP visualization of perturbation condition embeddings colored by *functional groups* on the left. Crosses indicate perturbations that have been tested experimentally in the original study. Colored dots indicate perturbation conditions with annotations. *(Right)* UMAP visualization of perturbation condition embeddings colored by average predicted marker gene expression on the right for the Erythroid and Granulocyte cell groups.



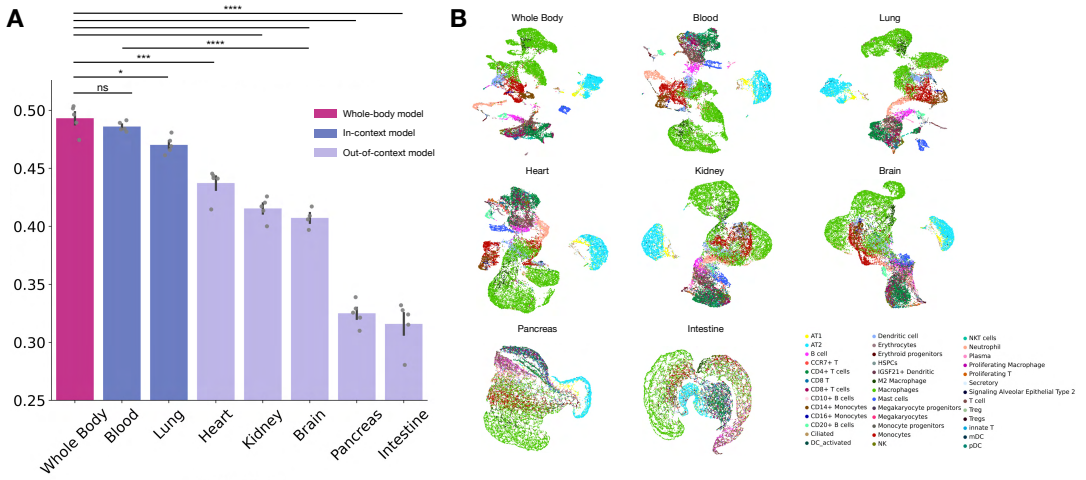
Supplementary Figure 5: Visualization of the scGPT cell and gene embeddings on the PBMC 10K dataset, using the pre-trained model without fine-tuning (i.e., zero-shot). *(A)* UMAP visualization of cell embeddings colored by *cell types*. *(B)* UMAP visualization of gene embeddings. The highly variable genes corresponding to major *celltype* were colored accordingly.



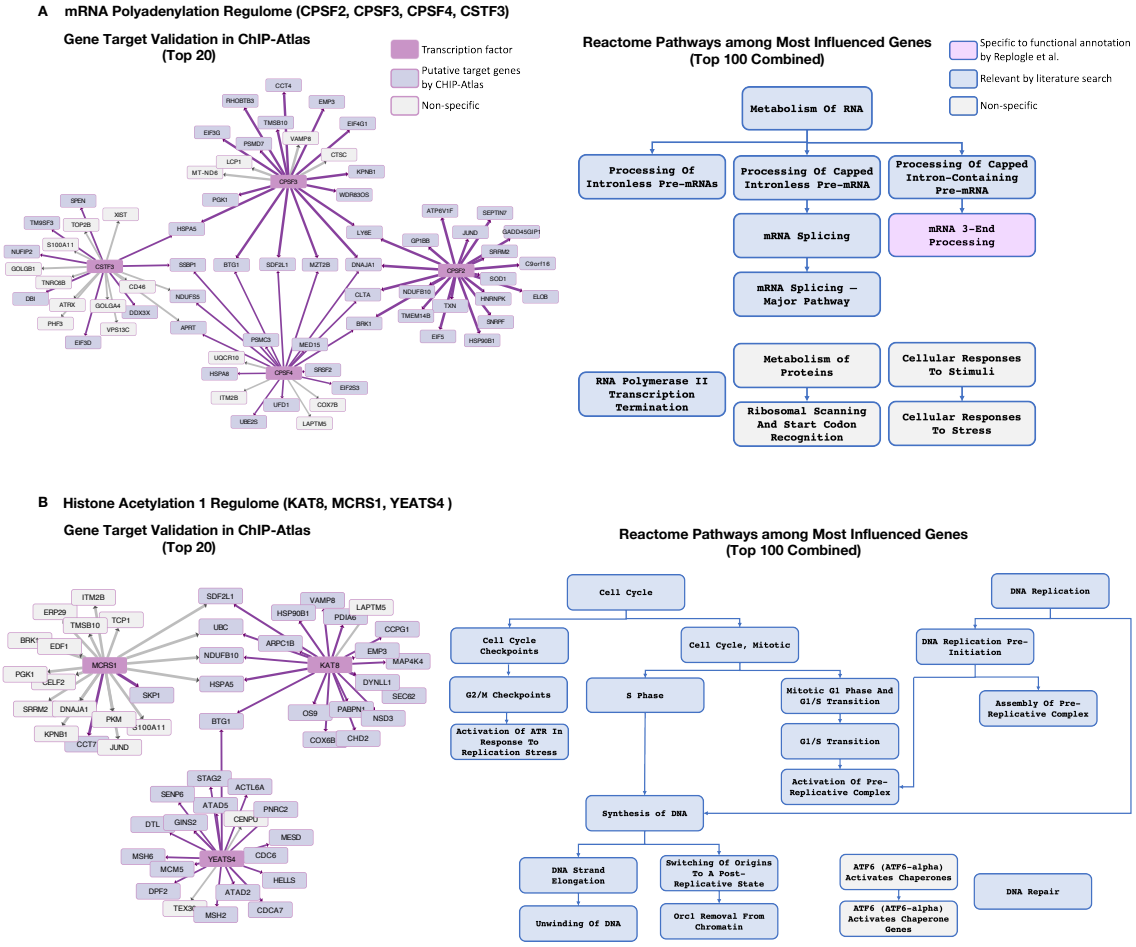
Supplementary Figure 6: *(A,B,C)* Benchmark of the fine-tuned scGPT model with scVI [8], Seurat Seurat [9], and Harmony Harmony [10] on the COVID-19 (18 batches) [1], PBMC 10K (2 batches) [11], and Perirhinal Cortex (2 batches) [12] datasets for cell type clustering performance upon batch integration. The UMAP plot of learned cell embeddings was colored by *cell types*.



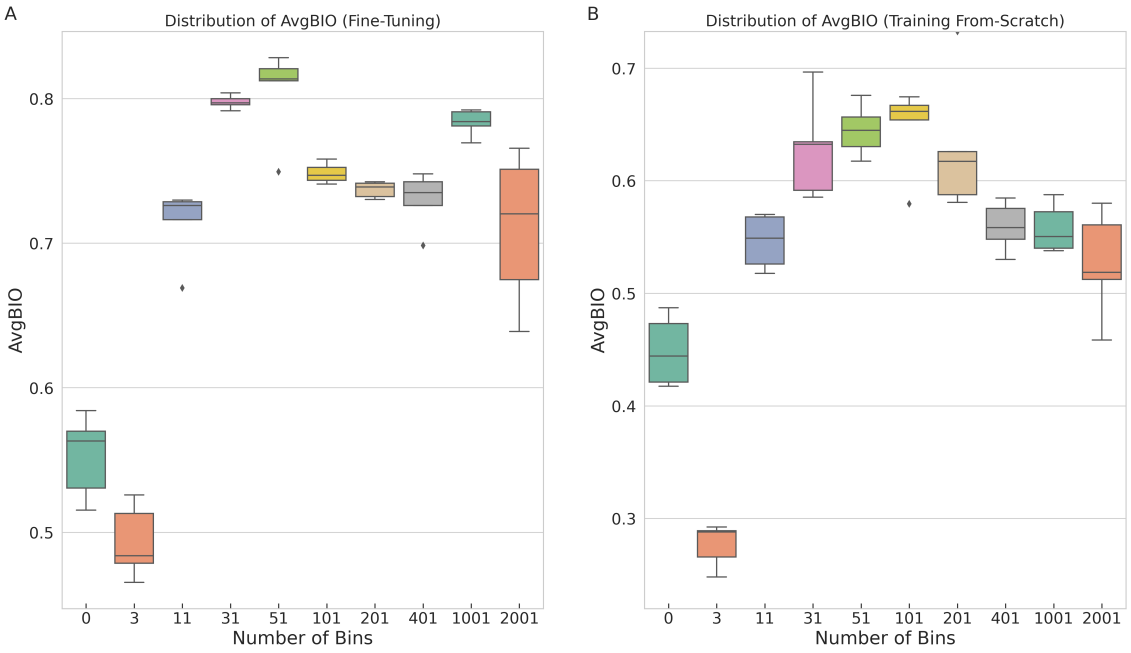
Supplementary Figure 7: Benchmark of scGPT with scVI [8], Seurat [9], and Harmony [10] on the COVID-19 (18 batches) [1], PBMC 10K (2 batches) [11], and Perirhinal Cortex (2 batches) [12] Datasets for Batch Correction. UMAP visualization of cell embeddings colored by *sequencing batches*.



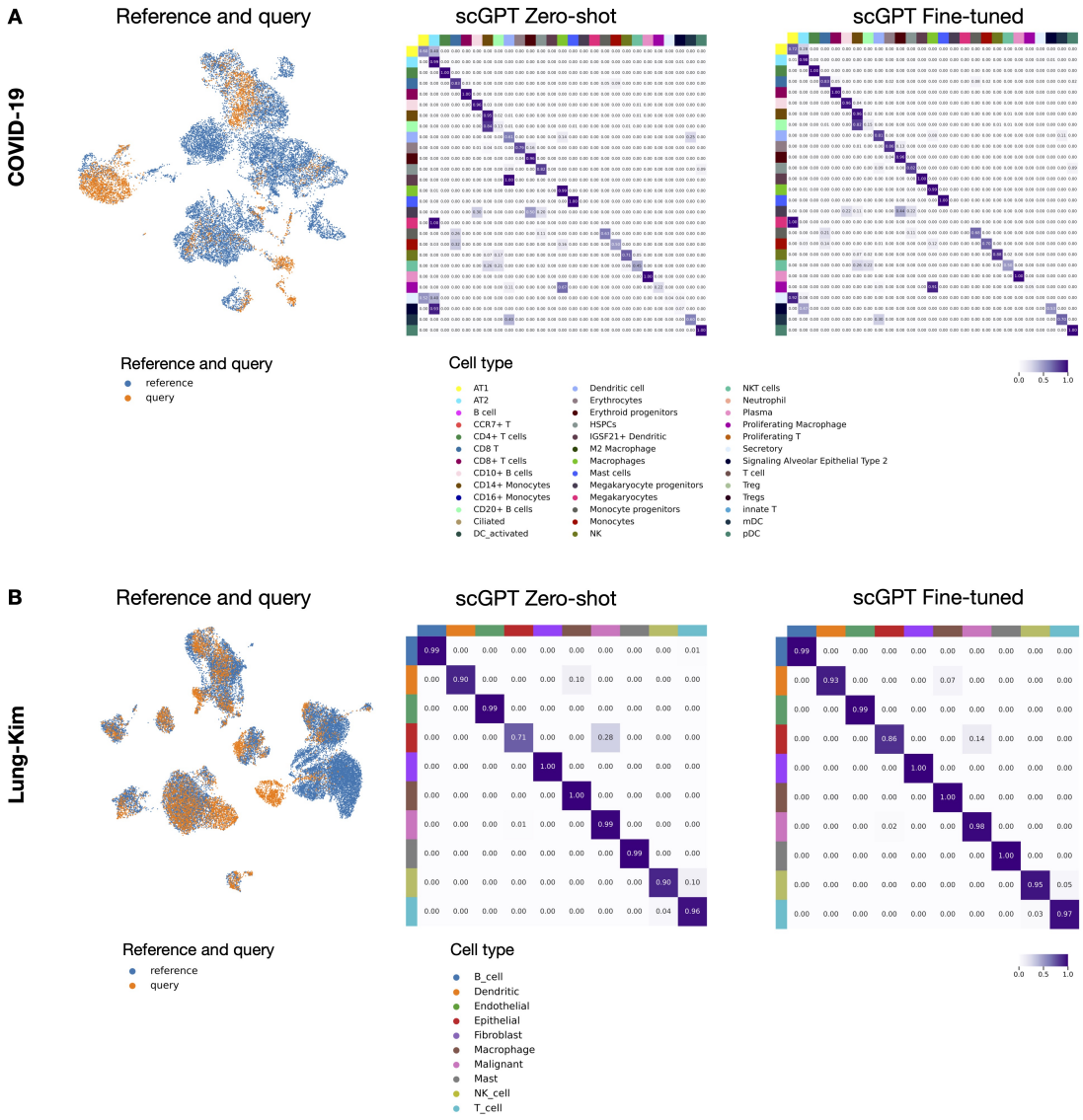
Supplementary Figure 8: scRNA-seq batch integration results on the COVID-19 dataset (comprising 18 batches) [1], with fine-tuning applied on each of our tissue-specific pre-training models. The COVID-19 dataset comprises a range of samples, including those derived from COVID-19, lung, bone marrow, and PBMC data, reflecting its cellular context. *(A)* Illustration of the mean *AvgBIO* score with ± standard error derived from ($n = 5$) scRNA-seq integration experiments for each pre-trained model. $^{*}p = 0.0487$, $^{***}p = 0.0008$, $^{****}p < 0.0001$ (independent samples t-test). The *AvgBIO* score provides a quantitative measure of the efficacy of each pre-trained model in integrating batches. The models are ordered by their average score, highlighting the potential of tissue-specific pre-training in improving the performance of integration. In-context models (Blood and Lung), which align with the cell types in the target dataset, are separated from out-of-context models (Heart, Brain, Kidney, Pancreas, Intestine) by coloring, highlighting the influence of cellular context on the integration performance. *(B)* UMAP visualization of the cell embeddings, colored by *cell types*, achieved by fine-tuning each of the tissue-specific pre-training models. The plots depict the quality of learned representations in the context of cell type diversity, offering visual confirmation of the models' integration capabilities.



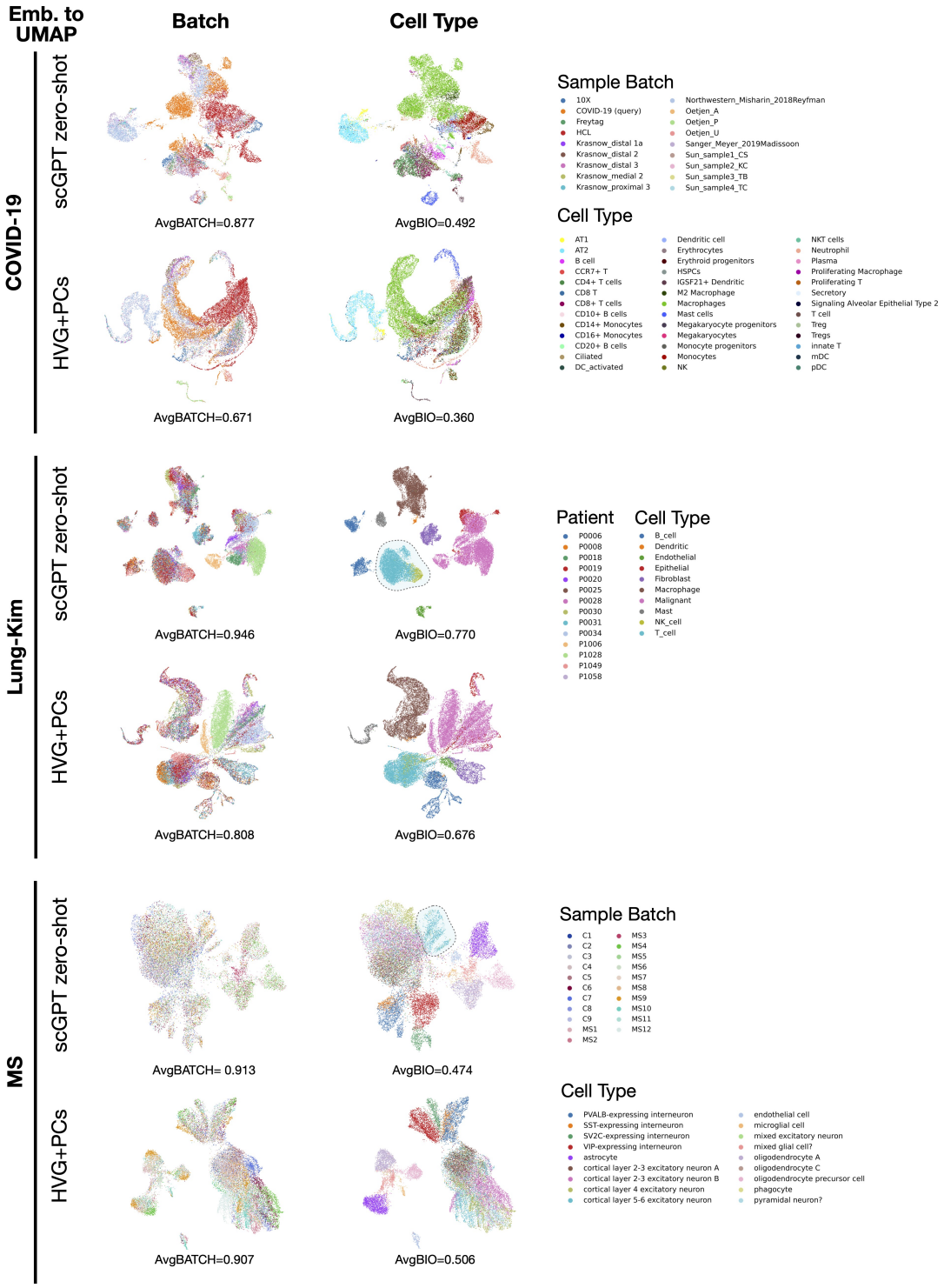
Supplementary Figure 9: scGPT's attention-based GRN analyses predict gene interactions for functionally related transcription factors from the Replogle CRISPRi perturbation dataset [20]. *(A)* GRN analysis for transcription factors CPSF2, CPSF3, CPSF4, and CSTF3 involved in mRNA polyadenylation regulation. The gene target network graph showcases the top 20 most influenced genes identified by scGPT for each transcription factor. Verified gene targets by the CHIP-Atlas database are highlighted in purple. The edge strengths reflect importance scores from scGPT. The pathway diagram showcases grouping of enriched functional pathways among the top 100 most influenced genes from these transcription factors from the Reactome database. The term specific to Replogle et al.'s annotation (i.e., mRNA polyadenylation) is highlighted in purple. The terms related to these TFs or mRNA polyadenylation by literature search are highlighted in blue. The non-spefic pathways are colored in grey. *(B)* GRN analysis for transcription factors KAT8, MCRS1, and YEATS4 involved in histone acetylation regulation.



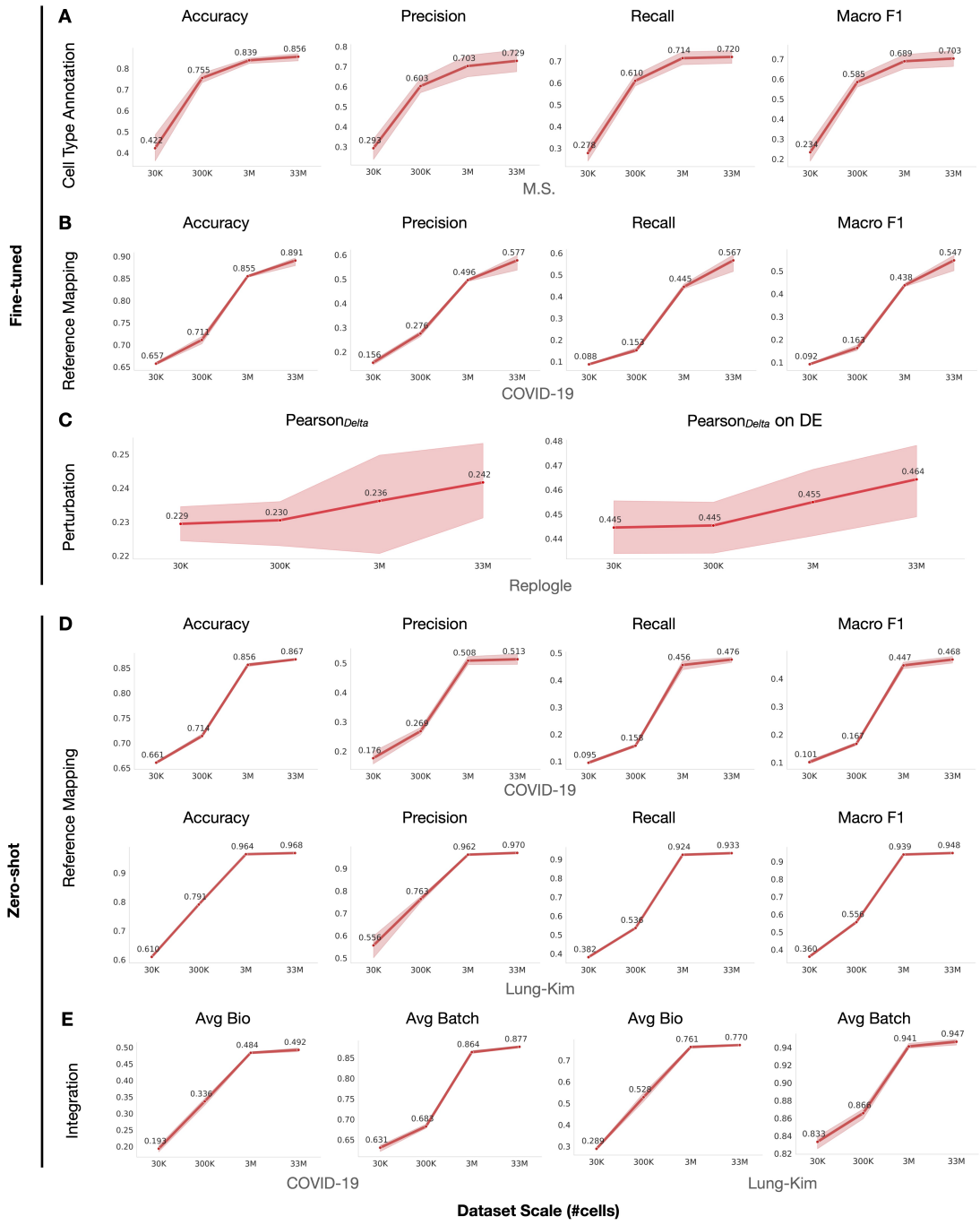
Supplementary Figure 10: Distribution of integration performance across different binning strategies. The model is trained on the PBMC 10K dataset for five runs per setting. Varying numbers of bins were used for the data processing. The option of zero bins indicates using $log1p$ transformed data as input without binning. The box denotes the interquartile range of avgBIO. The median is marked by the central line within each box. Whiskers extend to 1.5 times the interquartile range. *(A)* The avgBIO scores reported when fine-tuning the pre-trained whole-human model. *(B)* The avgBIO scores reported when training a smaller model of four-layer transformer from scratch.



Supplementary Figure 11: Reference mapping results. *(A)* On COVID-19 [1] dataset, UMAP of cell embeddings for zero-shot setting, and the confusion matrix computed by zero-shot and fine-tuned predictions, respectively. *(C)* On Lung-Kim [5] dataset, UMAP of cell embeddings for zero-shot setting, and the confusion matrix computed by zero-shot and fine-tuned predictions, respectively.



Supplementary Figure 12: Multi-batch integration on three datasets using the zero-shot cell embeddings. UMAPs of the embeddings are visualized and compared with the UMAPs of the principle components (PCs) of highly variable gene (HVG) expression. Colors indicate the batch and cell types in each dataset. The *AvgBATCH* and *AvgBio* metrics are reported. We outlined the cell types that are uniquely separable using scGPT cell embeddings on the Lung-kim and multiple sclerosis (MS) datasets.



Supplementary Figure 13: Influence of pre-training dataset size to various applications in both fine-tuning and zero-shot settings. The COVID-19 [1], Lung-Kim [5], M.S. [54], and Replogle [20] datasets were used in the experiments. Mean metrics on the test set, with $n = 5$ random train-validation splits, were presented per task per pre-training setting. Shaded regions denote $\pm$ one standard error. *(A,B,C)* Performance of *fine-tuned* scGPT for cell type annotation, reference mapping, and perturbation response prediction, respectively. *(D,E)* Performance of *zero-shot* scGPT for reference mapping and multi-batch integration, respectively.