

Appendix for TABULA: A Tabular Self-Supervised Foundation Model for Single-Cell Transcriptomics

A Pretraining Data Collection and Preprocessing

Data collection. To establish a robust foundation for TABULA self-supervised *pretraining*, we assemble an extensive single-cell RNA sequencing (scRNA-seq) dataset comprising transcriptomic data from approximately 15 million human cells, sourced from the CELLxGENE portal (<https://cellxgene.cziscience.com/>).

Given that the CELLxGENE portal is regularly updated, we utilized the release from July 25, 2023. The dataset encompasses cells from a range of major human tissues, including the intestine, pancreas, lung, heart, blood, kidney, and brain. Furthermore, cells from a variety of less-represented human tissues like the spinal cord and spleen were aggregated into a separate category called “others.” The compilation resulted in a total of 8 categories, namely “intestine”, “pancreas”, “lung”, “heart”, “blood”, “kidney”, “brain” and “others”, comprising **37.99 million human cells**. In our federated learning framework, we set the number of clients to be eight, each corresponding to a distinct tissue category aforementioned in our setting. To maintain data balance across categories and guarantee effective pretraining, a selection process was implemented, limiting each category to a maximum of 3 million cells.

We regrouped each category according to its original dataset identifiers. We then performed quality control and data preprocessing procedures for cells belonging to the same source dataset separately (detailed methodology described in the next subsection). Within each category, we randomly selected 3 million cells from the preprocessed datasets. For categories with fewer than 3 million total available cells, all cells were used. This process ultimately yields a pretraining dataset of 15 million cells. The number of pretraining cells for each tissue client is as follows:

Intestine: 80,060 cells; **Pancreas:** 220,436 cells; **Lung:** 3,013,840 cells; **Heart:** 1,768,184 cells; **Blood:** 3,057,298 cells; **Kidney:** 816,538 cells; **Brain:** 3,024,233 cells; **Others:** 3,046,184 cells.

Quality control & data preprocessing for federated learning. We implemented a dataset-specific strategy for quality control and data preprocessing. This strategy diverges from previous approaches of consolidating data prior to quality control and data preprocessing, allowing us to preserve the distinctive characteristics of each original dataset and reduce the impact of the batch effect. Specifically, we filtered cells with fewer than 250 detected genes to exclude potential empty droplets, non-viable cells, or severely degraded cells. Concurrently, we also removed genes detected in fewer than 250 cells to mitigate the influence of lowly expressed or rare transcripts. To capture biologically relevant variations, we identified 1,200 highly variable genes (HVGs) within each dataset after quality control. It is worth noting that HVGs were identified prior to the aggregation of datasets for each tissue. Consequently, HVGs may differ among datasets even within the same tissue. We chose this strategy because selecting HVGs from individual datasets or distinct contexts can improve the model’s ability to discern cell-specific characteristics while also enhancing the model’s capacity to learn generalizable features from varied biological signals. To standardize the data for pretraining and further mitigate batch effects, we implemented a value binning technique, discretizing the expression values into 50 bins for each cell. This transformation normalizes the scale of gene expression across batches, creating a more uniform input space for the pretraining process. The choice of 50 bins strikes a balance between preserving gene expression differences and reducing batch-specific technical variations.

Data preprocessing for tabular modeling. Our model employs a tabular learning approach for self-supervised *pretraining*, with learning objectives that include contrastive loss and reconstruction loss. The contrastive loss focuses on enhancing feature discrimination between cells, while the reconstruction loss aims to capture complex relationships among genes within each cell. To achieve these objectives, we need to corrupt the input samples to construct corrupted views from the original views. We followed Xtab [9] to construct corrupted views through random feature resampling. Specifically, we randomly selected a subset of genes in cells in a training batch and then resampled their values from the empirical marginal distribution [10] of these genes within the same training batch. We set the corrupted ratio at 60%. This means that for each cell sample’s original view x and its corrupted view \tilde{x} , 60% of the gene values are resampled while 40% remain unchanged.

This resampling strategy will provide sufficient perturbation to enhance the model’s robustness while retaining enough original information to maintain biological relevance.

Following the assembly of the pretraining dataset, Figure 7 summarizes its composition and cellular diversity across tissues. Figure 7a illustrates the spatial distribution and corresponding cell counts for each of the eight federated clients, representing major human tissues and collectively comprising approximately 15 million cells used for pretraining. Figure 7b presents UMAP visualizations of 10,000 randomly sampled cell embeddings from each tissue, generated by pretrained client-specific models. These visualizations reveal clearly separated, tissue-specific clustering patterns. Notably, the “Others” client aggregates cells from various less-represented tissues, yet still maintains distinguishable cellular structures. Collectively, this figure highlights the scale, diversity, and federated organization of the dataset, which provides the foundation for TABULA’s ability to learn robust and generalizable single-cell transcriptomic representations.

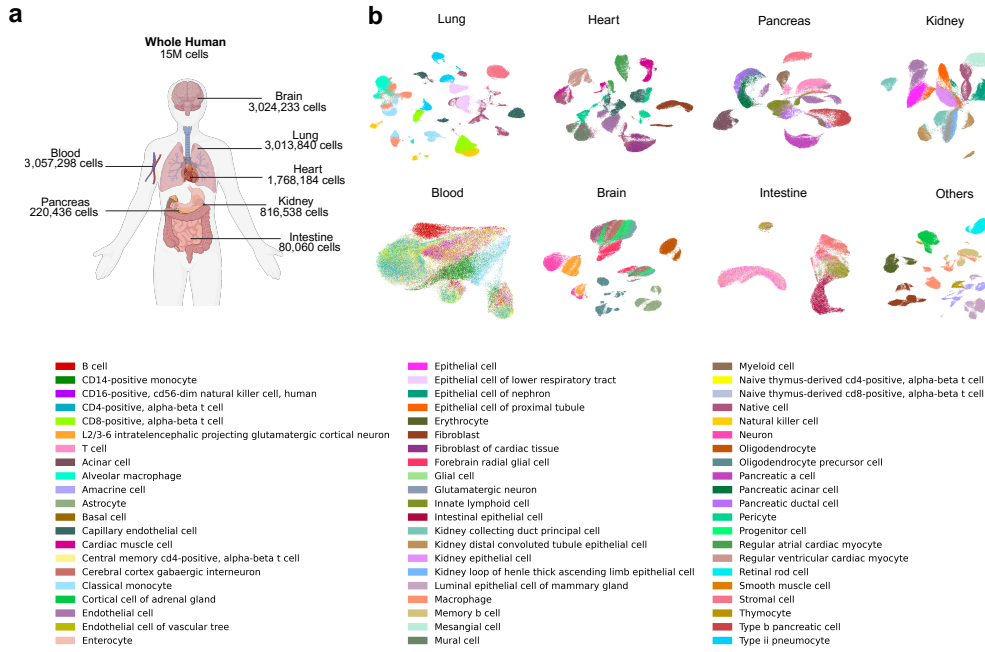


Figure 7: Overview of pretraining dataset in TABULA. (a) Summary of TABULA’s pretraining scRNA-seq dataset, which includes approximately 15 million cells from multiple human tissues. Each tissue is assigned to a distinct client to simulate federated learning. (b) UMAP visualization of cell embeddings from pretrained client models, with 10,000 cells randomly sampled per client. The “Others” client aggregates less-represented tissues.

B Collaborative Federated Learning Framework

TABULA is based on the federated learning framework. Unlike existing foundation models based on centralized learning, which require aggregating all data in a central location and thus raise privacy and ethical concerns, TABULA allows data to remain distributed across multiple clients. Each client trains a local model on its own client data, and only model parameters are shared with the central server for global model updates. To be specific, the key steps are as follows:

- **Local training.** Each client k (e.g., institution or hospital) trains a local copy of the model using its own data D_k . During this step, the model weights w_k are updated by minimizing the local loss function $\mathcal{L}_k(w)$ based on the client's local dataset. The local training process can be denoted as:

$$w_k^{T+1} = w_k^T - \eta \nabla \mathcal{L}_k(w_k^T) \quad (8)$$

where T is the current iteration, η is the learning rate, and $\nabla \mathcal{L}_k(w_k^T)$ is the gradient of the loss function at the current iteration weights w_k^T . The model weights contain two components: $w_k^{(t)}$ is the model weights for the tabular transformer, and $w_k^{(o)}$ is the model weights for the rest of the architecture, including embedders and project heads. w_k^T can be represented as $w_k^T = w_k^{(t)} + w_k^{(o)}$.

- **Sending local updates.** Once local training is complete, each client k sends its updated tabular transformer gradient $w_k^{T,(t)} - w_k^{T+1,(t)}$ to a central server. These gradients represent the local knowledge learned from the individual datasets. This can be summarized as:

$$\text{Client } k \rightarrow \text{Server} : w_k^{T,(t)} - w_k^{T+1,(t)} \quad (9)$$

- **Global aggregation.** The central server aggregates the local tabular transformer updates from all clients to create a new global tabular transformer. In TABULA, we adopt a common aggregation method, which is Federated Averaging (FedAvg) [23].

The global tabular transformer weights $w_g^{T+1,(t)}$ are updated by computing a weighted average of the local tabular transformer updates, where the gradients reflect the significance of each client. The importance of each client is quantified by a corresponding weight p_k . It can be summarized as:

$$P = \sum_{k=1}^K p_k \quad (10)$$

$$w_g^{T+1,(t)} = w_g^{T,(t)} + \frac{1}{P} \sum_{k=1}^K p_k \left(w_k^{T,(t)} - w_k^{T+1,(t)} \right) \quad (11)$$

- **Broadcast global update.** After aggregation, the global tabular transformer is updated with the newly averaged weight $w_g^{T+1,(t)}$. This updated global model is then broadcasted back to the clients, which subsequently update their local models based on the received global weights. This process can be expressed as:

$$\text{Server} \rightarrow \text{Clients } k : w_g^{T+1,(t)} \quad (12)$$

$$w_k^{T+1} = w_g^{T+1,(t)} + w_k^{T+1,(o)} \quad (13)$$

Those processes are repeated for several rounds, with clients continuing to train the model locally, send updates, and the server performing global aggregation, until the model converges or reaches the desired performance. In TABULA, we update the global model once every epoch and treat each client with the same importance weight.

Our federated pretraining framework offers three key advantages. First, it ensures data privacy, as sensitive information remains localized on the client side, preventing unnecessary data transfers. Second, it distributes the computational burden across multiple clients, alleviating the demand for resources on any single entity and enhancing scalability. Third, the federated architecture naturally

enables the development of tissue-specific embedders and project heads, which are designed to capture tissue-specific features. In contrast, conventional foundation models rely on a single, generalized embedder, making them less effective at capturing the nuances of tissue-specific variations. It is important to note that in our federated training framework, both the local model and the tissue-specific embedder, along with the project head, can be shared. This allows for greater flexibility and collaboration across clients while preserving the benefits of tissue-specific feature learning.

C Pretraining Implementation Details

TABULA was implemented in Python and utilized PyTorch Lightning to construct its overall framework. Within the federated learning setup, message passing interface (MPI) was employed in the cluster to allocate computing nodes to each client. The NVIDIA Collective Communications Library (NCCL) framework was adopted for efficient communication among clients over the InfiniBand (IB) network. Additionally, *torch.distributed* was implemented to realize the message queues of the parameter sending, receiving, broadcasting, and managing process.

- **TABULA Pertaining on 15 Million Cells.** TABULA employs FlashAttention-2 architecture [24], comprising 3 transformer blocks, each containing 8 attention heads, with an output embedding size of 192. The maximum input gene length was set to 1,200, corresponding to 1,200 highly variable genes (HVGs) selected from each dataset. TABULA was trained using 8 NVIDIA A100 40GB GPUs, with each GPU serving as a distinct client which has tissue-specific datasets. The pretraining dataset consisted of 15 million cells, which were randomly sampled from tissue-specific subsets within the CELLxGENE whole-human dataset. During the pretraining stage, an AdamW optimizer [25] was utilized with a 0.0001 initial learning rate, which was decreased by a factor of 0.95 after each epoch. The model was trained for 8 epochs with a batch size of 32. For the tissue-specific client models, 90% of the data was used for training and the remaining 10% for validation. The corruption rate was set to 0.6 and the temperature coefficient of $\mathcal{L}_{\text{contrast}}$ to 0.07. To achieve a relative balance between the two learning objectives, the weight of $\mathcal{L}_{\text{contrast}}$ and \mathcal{L}_{rec} was set to 1:0.03. For the federated learning settings, local tabular transformer weight aggregation and update were performed at the end of every training epoch for all clients. Since all clients are considered equally important, the global tabular transformer weight was averaged across all clients during aggregation. After pretraining, we extract the client model weights following 8 epochs as the final tissue-specific embedders and project heads weights, and we use the aggregated weight of the transformer as the final global tabular transformer weight.
- **Benchmarking Federated vs. Centralized Pretraining on 1 Million Cells** We benchmarked two pretraining schemes: federated learning and centralized learning. The federated learning scheme followed the same training methodology as TABULA. In contrast, the centralized learning scheme adopted a traditional approach, where all data was consolidated, and model pretraining was performed on a single centralized device using all acquired data. This approach avoids maintaining multiple models for individual clients and performing model weight aggregation and updates across clients.
- **Benchmarking Tabular Learning vs. Masked Language Modeling on 1 Million Cells** We benchmarked two single-cell modeling strategies: tabular learning and masked language modeling (MLM). The tabular learning strategy used the same learning objectives as TABULA, while the MLM strategy aimed to reconstruct these masked expression values through contextual inference. Specifically, 15% of the non-zero gene expression values in the input sequence were randomly masked by setting their expression levels to -1, following scBERT [8]. The mean squared error (MSE) loss is computed exclusively for the masked positions.

D Privacy Leakage in Single-Cell Expression Matrices

Anonymized cell-by-gene count matrices, which are routinely shared in single-cell RNA sequencing (scRNA-seq) studies, pose a significant privacy risk due to the genetic basis of transcriptional regulation. These matrices are vulnerable to *linking attacks* [6], a class of privacy breaches wherein gene expression profiles are used to reconstruct genotypes, which are then matched to reference genomic databases to re-identify individuals and infer sensitive phenotypes.

We outline the key steps of this attack pipeline below:

1. **Input: Raw Cell-by-Gene Count Matrices**

The attack begins with matrices representing transcript counts for thousands of genes across individual cells from a donor. Although such matrices are typically de-identified, they retain gene expression signatures that are genetically regulated and potentially identifying.

2. **Aggregation into Cell-Type-Specific Pseudobulk Profiles**

To enhance signal and suppress noise, cells are grouped by type and expression values are aggregated to form pseudobulk profiles. Each resulting matrix contains gene expression levels per individual for a given cell type, preserving genotype-dependent expression patterns.

3. **Genotype Inference via Public eQTL Databases** QTL Expression expression loci (eQTL) [26] from public resources (e.g. GTEx or single-cell eQTL consortia) are used to reverse-engineer genotypes from pseudobulk data. This step exploits established eQTL relationships to estimate an individual’s genotype at loci known to influence expression.

4. **Construction of Genotype Vectors**

For each individual, predicted genotypes at eQTL loci are assembled into vectors. Despite being noisy estimates, these vectors capture sufficient genetic information to allow re-identification.

5. **Linkage to Reference Genomic Databases**

The inferred genotype vectors are then compared to a reference panel (e.g., 1000 Genomes or publicly leaked datasets). Using similarity metrics, individuals are matched, enabling the attacker to link expression profiles to real identities.

6. **Phenotype Inference via Re-Identification**

Once re-identification occurs, any associated metadata from the expression dataset, such as disease status (e.g., HIV, cancer, psychiatric conditions), can be attributed to the matched individual. Thus, sensitive traits can be inferred even if only expression data was released.

Existing single-cell foundation models are predominantly trained using centralized learning frameworks on publicly available, anonymized cell-by-gene count matrices. However, these matrices have been shown to be privacy-sensitive, as they can be exploited to infer genotypes and reveal sensitive phenotypes through linking attacks. As scRNA-seq datasets continue to expand in scale and resolution, the risk of individual re-identification and phenotype leakage escalates. This highlights the urgent need for federated learning-based foundation models that can preserve data locality and confidentiality, while still enabling the extraction of transferable biological representations across institutions.

E Benchmarking on Downstream Tasks

E.1 Task 1: Gene Imputation

E.1.1 Fine-tuning Objective

We assessed the imputation performance using several quantitative metrics. The primary evaluation involves the Pearson correlation coefficient (Pearson), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) to measure the accuracy of imputed values compared to the true values. To demonstrate the model’s robustness, we applied different masking ratios (10%, 20% and 30%) to the dataset, with Pearson correlation as one of the metrics. Additionally, the model’s performance is analyzed through Mean Squared Error (MSE) at both the cell-wise and gene-wise levels. To assess the quality of imputation, cosine similarity is calculated between different cell types. This helps evaluate how well the imputed data preserves the original structure and relationships of the data. Specifically, it provides insights into the accuracy of the imputation and ensures that the natural diversity and variability between cell types remain consistent before and after the imputation process.

Since the true dropout values are unknown, we evaluated the method’s performance by randomly masking the expression matrix in the scRNA-seq dataset, replacing these values with zeros. To more accurately simulate the distribution of dropout values, we adopted the settings from DeepImpute [27] and utilized an exponential kernel [3]. For each cell, we identify genes with non-zero expression values, excluding cells with few expressed genes. For these non-zero genes, we compute sampling probabilities using an exponential distribution and normalize them to sum to 1. Using these probabilities as weights, we randomly sample a subset of the non-zero genes without replacement. The expression values of the selected genes are then masked to zero to simulate dropout genes, while the remaining genes retain their original expression values. The model is then used to reconstruct the true expression values of these masked genes.

During the fine-tuning process, the masked data is divided into three sets: the training set, the validation set, and the test set. For each set, predictions are made exclusively for the positions within that specific set. Subsequently, an imputation decoder (MLP) is introduced for each gene, with the objective of predicting the expression levels at the respective positions. $\hat{x}^{(i)}$ is computed by passing the integrated embedding to the transformer layers and the imputation decoder. We calculate the imputation loss \mathcal{L}_{imp} in masked positions. The process is outlined in the formula below:

$$\hat{x}^{(i)} = \text{MLP} \left(\text{transformer_block}(\text{addition}(\mathbf{E}_{\text{token}}^{(i)}, \mathbf{E}_{\text{val}}^{(i)})) \right) \quad (14)$$

$$\mathcal{L}_{\text{imp}} = \frac{1}{|\mathcal{M}_{\text{mask}}|} \sum_{j \in \mathcal{M}_{\text{mask}}} \left(\hat{x}_j^{(i)} - x_j^{(i)} \right)^2 \quad (15)$$

Here, $\mathbf{E}_{\text{token}}^{(i)}$ and $\mathbf{E}_{\text{val}}^{(i)}$ are representation embeddings of gene token and gene expression value, respectively.

We assessed our model using four datasets derived from PBMC5K, Jurkat, Melanoma, and hPancreas (see Appendix H Downstream Task Datasets). We benchmarked TABULA against scGPT [4], scBERT [8], and Geneformer [1]. This comparison was conducted in a consistent fine-tuning framework, utilizing gene-level embeddings for the prediction of dropout expression profiles.

E.1.2 Implementation Details

For the imputation task, the initial learning rate was set to 0.0001 with a weight decay of 0.95. The batch size was set to 32, and fine-tuning was performed over a fixed 10 epochs. The maximum sequence length was set to 2,000. We masked the data to simulate dropout values. This process masked non-zero gene expression data for each cell, splitting it into training, validation, and test sets.

For each cell, non-zero values were identified. If there were fewer than 5 non-zero values, no masking was applied. Otherwise, based on an exponential probability distribution (scale 20), 10% of the data was masked for validation and another 10% for testing. The masked values were set to 0, and the remaining data formed the training set.

The data preprocessing steps included TPM normalization and log1p transformation. We only conducted a value binning preprocess for the gene expressions in the training and validation sets, and the test sets were used with log1p values. The model weights exhibiting the lowest loss on the validation set were selected as the optimal weights for evaluating performance on the test set.

E.1.3 Evaluation Metrics

To evaluate the performance of imputation tasks, we use multiple metrics, including the Pearson correlation coefficient, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). The Pearson correlation coefficient quantifies the linear relationship between imputed and true gene expression values, while RMSE and MAE measure the differences between them. All metrics are computed across all imputed gene expression values and are formulated as follows:

$$\text{Pearson} = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \quad (16)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2} \quad (17)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |x_i - y_i| \quad (18)$$

Here, x_i and y_i represent the imputed and true gene expression values for masked gene i , respectively. \bar{x} and \bar{y} are the mean values of the imputed and true gene expression values across all masked genes. N denotes the total number of masked genes across all cells.

To evaluate the consistency between the original and imputed data, we calculate the Mean Squared Error (MSE) at two levels: cell-cell and gene-gene. The definitions are as follows:

Cell-cell MSE:

$$\text{MSE}_{\text{cell-cell}} = \frac{1}{m} \sum_{j=1}^m (x_j - y_j)^2 \quad (19)$$

Here, x_j and y_j represent the imputed and true gene expression values for masked gene j in a given cell. m is the total number of masked genes in this cell.

Gene-gene MSE:

$$\text{MSE}_{\text{gene-gene}} = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (20)$$

Here, x_i and y_i represent the imputed and true gene expression values for masked cell i in a given gene. n is the total number of masked cells in this gene.

E.1.4 Additional Results

The gene imputation task utilizes dropout probabilities derived from an exponential distribution to mask gene expression values to zero, simulating dropouts in single-cell data, after which the fine-tuned TABULA recovers these masked values, as shown in Figure 8a. TABULA exhibits robust performance across a range of masking ratios (10%, 20%, 30%) on the PBMC5K dataset (Figure 8b). This performance is evaluated by Pearson correlation for both the entire dataset and the top five clusters. A comparison was also made between the ground truth and imputed values on the PBMC5K dataset across multiple benchmark models, including scGPT[4], Geneformer[1], and scBERT[8], using log-transformed scatterplots with density coloring, accompanied by quantitative metrics (Figure 8c). A heatmap visualisation on the hPancreas dataset further illustrates the fidelity and consistency of cell type similarity patterns before and after imputation (Figure 8d), with colored squares at the edges representing original and imputed data and varying internal colors showing similarities between cell types. It is noteworthy that TABULA effectively preserves similarities within the same cell type while

maintaining clear distinctions between different cell types, thus validating its biological relevance and accuracy in preserving cellular identity.

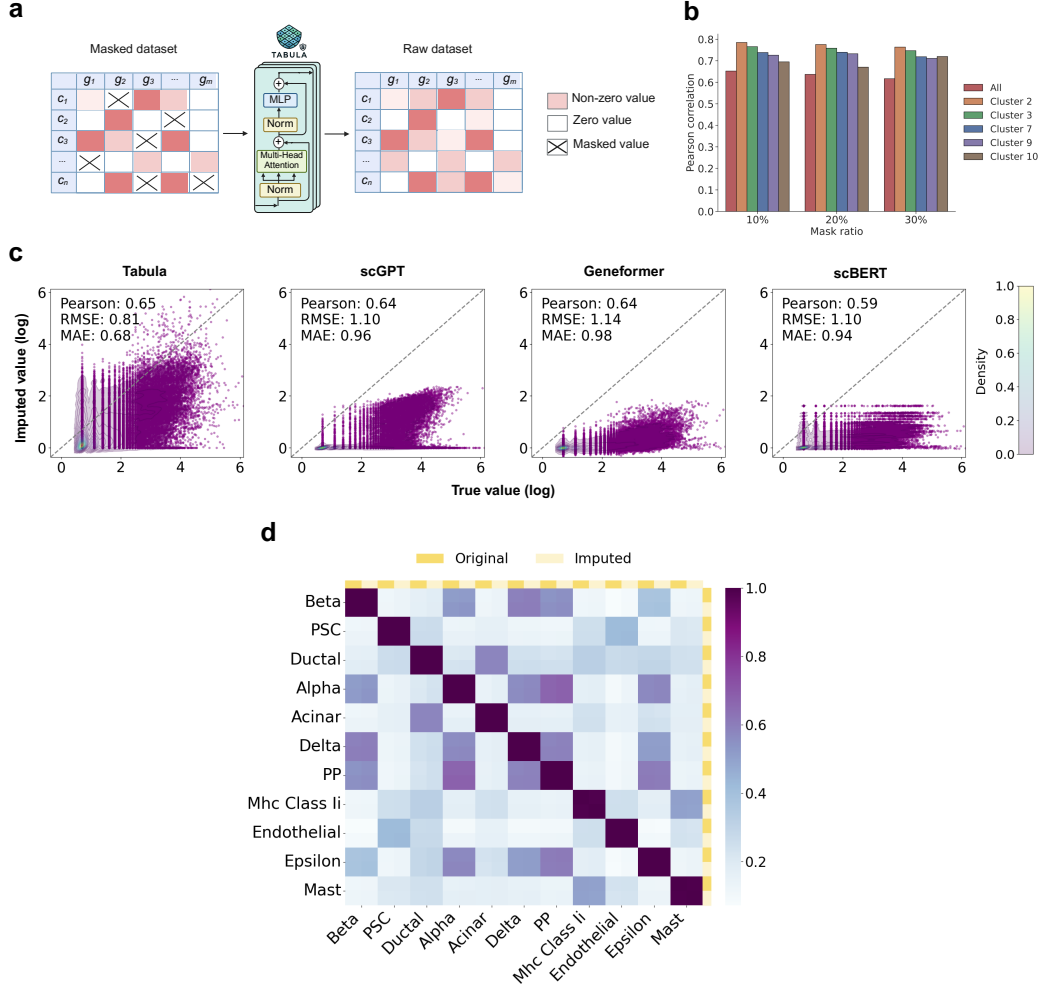


Figure 8: (Task 2: Gene Imputation) xxxx Overview of the gene expression imputation task and evaluation of TABULA's accuracy, robustness, and cellular identity preservation.

E.2 Task 2: Genetic Perturbation Prediction

E.2.1 Fine-tuning Objective

We assessed the performance on genetic perturbation prediction using two key metrics: $Pearson_{\Delta}$, which quantifies the correlation between predicted and actual expression changes after perturbation, and $DEG\ Pearson_{\Delta}$, which specifically evaluates this correlation for the top 20 most differentially expressed genes.

During model testing, we consider four distinct gene perturbation visibility scenarios defined in GEARS [18]: (i) 1/1 Unseen: single-gene perturbations where the gene is not observed during training; (ii) 2/2 Unseen: two-gene perturbations where neither gene is observed during training; (iii) 1/2 Unseen: two-gene perturbations where one gene is observed and one is not; (iv) 0/2 Unseen: two-gene perturbations where both genes have been seen during training.

These evaluations help elucidate the model's predictive robustness under varying levels of experimental uncertainty.

During the fine-tuning stage, a perturbation encoder (MLP) is initialized to incorporate perturbation information by adding a perturbation embedding $emb_p(t_p^{(i)})$ to the input. The predicted expression $\hat{x}^{(i)}$ is computed by passing the integrated embedding through the transformer layers and a reconstruction head. The perturbation loss $\mathcal{L}_{\text{pert}}$ is calculated as MSE loss across all positions, defined as:

$$\hat{x}^{(i)} = MLP \left(\text{transformer_block} \left(\text{addition}(\mathbf{E}_{\text{token}}^{(i)}, \mathbf{E}_{\text{val}}^{(i)}, emb_p(t_p^{(i)})) \right) \right) \quad (21)$$

$$\mathcal{L}_{\text{pert}} = \frac{1}{M} \sum_{j=1}^M \left(\hat{x}_j^{(i)} - x_j^{(i)} \right)^2 \quad (22)$$

Here, $\mathbf{E}_{\text{token}}^{(i)}$ and $\mathbf{E}_{\text{val}}^{(i)}$ are the representation embeddings of gene tokens and expression values, respectively.

We evaluated our model on three Perturb-seq datasets derived from leukemia cell lines, including Adamson, Norman, and Replogle (see Appendix H Downstream Task Datasets). Adhering to preprocessing protocols defined by GEARS [18], we benchmarked TABULA against scGPT [4], scBERT [8], and Geneformer [1]. This comparison was conducted within a consistent fine-tuning framework, utilizing gene-level embeddings and perturbation signals to predict perturbed expression profiles.

E.2.2 Implementation Details

For the gene perturbation prediction task, in order to reconstruct all genes to enable a comprehensive understanding of inter-gene relationships, for the datasets exceeding the maximum input length of 1,200 genes, we randomly initialized the parts that could not be accommodated by the pretrained model weights. The model optimization adopted a learning rate of 0.0001 and a weight decay of 0.95. The batch size was set to 64, while the training was conducted with an early stopping strategy with a patience of 5 epochs.

E.2.3 Evaluation Metrics

To evaluate the performance of genetic perturbation prediction tasks, we use Pearson correlation coefficient, which quantifies the linear relationship between predicted and actual expression changes following perturbation. The Pearson coefficient computed across all predicted gene expression changes for a cell is formulated as follows:

$$Pearson_{\text{delta}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (23)$$

Here, x_i and y_i represent the predicted and actual gene expression changes for gene i , respectively. \bar{x} and \bar{y} are the mean values of the predicted and actual expression changes across all genes. n is the total number of genes.

Similar to $Pearson_{\text{delta}}$, the Pearson coefficient computed across the top 20 differentially expressed gene expression changes for a cell, $DEG\ Pearson_{\text{delta}}$, is formulated as follows:

$$DEG\ Pearson_{\text{delta}} = \frac{\sum_{i=1}^{20} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{20} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{20} (y_i - \bar{y})^2}} \quad (24)$$

Here, x_i and y_i represent the predicted and actual gene expression changes for the top 20 differentially expressed genes (DEGs), respectively. \bar{x} and \bar{y} denote the mean values of the predicted and actual expression changes for these DEGs.

E.2.4 Additional Results

We conducted more experiments to assess the robustness of TABULA’s perturbation prediction across genes with varying expression levels and perturbation complexities (Figure 9). In the Adamson dataset, we perturbed AMIGO3 and evaluated the downstream effects on three representative target genes—HSPA8 (high expression), BACH1 (medium), and HAP1 (low)—with UMAP visualizations showing strong agreement between true and predicted expression changes (Figure 9a). In the Norman dataset, we introduced a double-gene perturbation (CBL and PTPN9) and examined the effects on HIST1H4C, EPHB6, and LLNLR-246C6.1, again spanning a range of baseline expression levels and showing accurate recovery of spatial response patterns (Figure 9b). To further evaluate prediction fidelity, we compared the distribution of expression changes for the top 20 differentially expressed genes following perturbations of KCD16, SYVN1, TTI1, and DAD1, observing close alignment between model predictions and ground-truth measurements (Figure 9c).

E.3 Task 3: Cell Type Annotation

E.3.1 Fine-tuning Objective

The performance of cell type classification was evaluated through metrics including accuracy, precision, recall, and F1 score, offering a well-rounded view of TABULA’s predictive accuracy against other models. Additionally, the confusion matrix was built to reveal patterns of predictions across cell types.

To fine-tune the pretrained TABULA for the cell type annotation task, an MLP classifier is introduced. This classifier takes cell embeddings generated by the Tabular Transformer as input and outputs categorical predictions for cell types. The entire model is optimized using cross-entropy loss \mathcal{L}_{AN} , as depicted below:

$$a_j^{(i)} = \text{MLP}(\mathbf{E}_{\text{cls}}^{(i)}) \quad (25)$$

$$\mathcal{L}_{\text{AN}} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(a_{j,c}^{(i)}) \quad (26)$$

Here, the MLP layer outputs the predicted probabilities of cell type labels $a_j^{(i)}$, taking the cell representation $\mathbf{E}_{\text{cls}}^{(i)}$ as input. The cross-entropy loss, \mathcal{L}_{AN} , is computed where N denotes the total number of samples in a batch, C is the number of cell type classes, $y_{i,c}$ represents the true label for the i -th sample for class c , encoded in one-hot format, and $a_{j,c}^{(i)}$ is the predicted probability for the i -th sample that belongs to cell type class c .

We assessed our model using four datasets including Human pancreas (hPancreas), Myeloid, Cell Lines, and Liver datasets (see Appendix H Downstream Task Datasets). Benchmark comparisons were made against scGPT [4], Geneformer [1], and scFoundation [2]. The fine-tuning settings of these benchmark models follow their default settings.

E.3.2 Implementation Details

For the cell type annotation task, we adopted an initial learning rate of 0.0001 and a weight decay of 0.1. The batch size was set to 32, while fine-tuning was performed with an early stopping strategy. For the cell line dataset, we randomly split 33% of the data as the test set, with the remaining 67% used for training, where 10% was randomly chosen as the validation set. In other datasets, the training and test sets were pre-defined (see Appendix H Downstream Task Datasets), and 10% of the training set was similarly reserved as the validation set. The optimal weights were selected based on the lowest loss on the validation set and used for evaluating the model’s performance on the test set.

The data preprocessing steps included transcripts-per-million (TPM) normalization, log1p transformation, and value binning. In addition, we selected 3,000 highly variable genes and prioritized the genes with non-zero expression in each cell during data loading until the maximum length requirement of 1,200 genes was reached. If fewer than 1,200 genes were selected, genes with zero expression were used to fill the remaining positions. This approach ensures that the model senses more genes during the fine-tuning process while meeting the maximum length requirement.

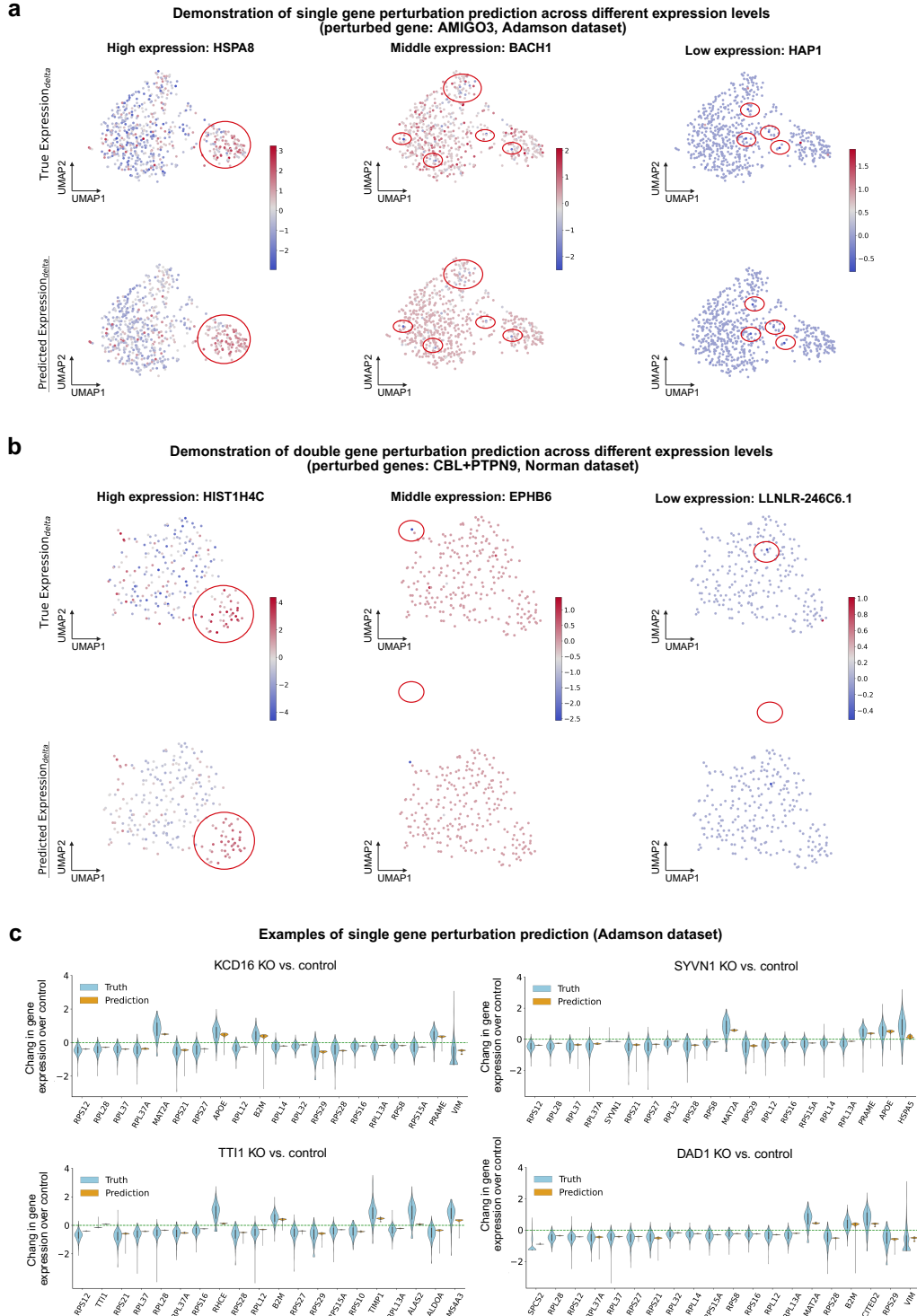


Figure 9: (Task 2: Genetic Perturbation Prediction) Scatterplot of target genes after single- or double-gene perturbation and examples of gene expression changes in response to single-gene perturbation.

E.3.3 Evaluation Metrics

To evaluate the performance of cell type annotation tasks, we use standard classification metrics: accuracy, precision, recall, and the F1 score. These metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (27)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (28)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (29)$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (30)$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

Precision quantifies the proportion of correctly predicted positive cell types among all predicted positives, while recall measures the ability to identify all true positive instances of a cell type. The F1 score provides a balanced measure that accounts for both precision and recall, especially useful for imbalanced datasets.

E.3.4 Additional Results

To further evaluate the capability of TABULA in annotating cells, we benchmarked on three publicly available datasets: hPancreas, myeloid, and 293T & Jurkat. As shown in Figure 10(a-c), Tabula successfully transfers labels from reference to query cells, accurately recovering both coarse-grained and fine-grained cell identities across diverse tissue types and conditions. The UMAP visualizations demonstrate strong alignment between predicted and ground truth cell types, while the corresponding confusion matrices (Figure 10d) highlight TABULA’s robust per-class accuracy, with high diagonal dominance indicating consistent annotation performance.

E.4 Task 4&5: Multi-Omics & Multi-Batch Integration

E.4.1 Fine-tuning Objective

Multi-Omics Integration To assess TABULA’s performance in integrating cell embeddings of multi-omics data after fine-tuning, we employ biological conservation metrics [28], including Normalized Mutual Information (NMI_{CELL}), Adjusted Rand Index (ARI_{CELL}), and Average Silhouette Width (ASW_{CELL}). These metrics quantify the alignment between cluster formations derived from the embeddings and the validated ground truth cell type annotations. Consistent with methodologies used in previous benchmarking studies [4], we calculated the average of these three metrics (AVGBIO) to provide a composite score reflecting overall performance in the task.

During the fine-tuning phase, the model weights are refined using multiple loss functions to enhance the learning of aligning different omics modalities. Specifically, the TABULA framework employs three fine-tuning losses: the reconstruction loss \mathcal{L}_{rec} , masked gene modeling (MGM), and cell context-aware masked gene modeling (CMGM). We detail these losses below:

- **Masked Gene Modeling (MGM).** To elucidate gene interrelationships, TABULA employs MGM loss as a fine-tuning objective. This approach parallels the reconstruction loss employed during pretraining but modifies the handling of genes at the masked positions. Specifically, whereas the reconstruction loss utilizes a corrupted view for masking, MGM involves the random masking of a subset of gene expression values e^i in each input cell. TABULA is then optimized to predict these masked expression values accurately. This setup enables the model to infer gene expression patterns from other genes within cells, thereby enhancing inter-gene awareness. The training objective is quantified using the mean squared error (MSE) at the masked positions, denoted as $\mathcal{M}_{\text{mask}}$. The formula is:

$$\hat{x}^{(i)} = \text{MLP} \left(\text{concat}(\mathbf{E}_k^{(i)}, \text{emb}_m(t_m^{(i)})) \right) \quad (31)$$

$$\mathcal{L}_{\text{MGM}} = \frac{1}{|\mathcal{M}_{\text{mask}}|} \sum_{j \in \mathcal{M}_{\text{mask}}} (\hat{x}_j^{(i)} - x_j^{(i)})^2 \quad (32)$$

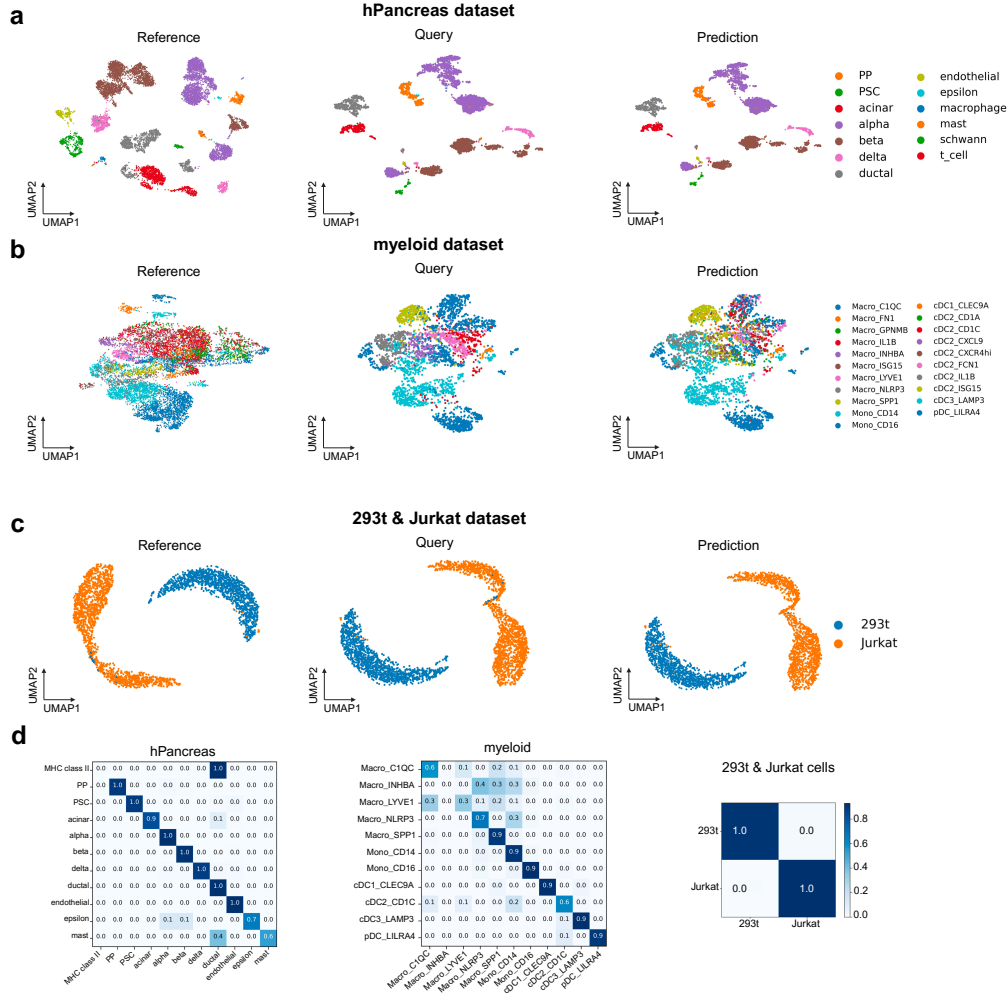


Figure 10: (Task 3: Cell Type Annotation) (a–c) UMAPs of reference, query, and predicted labels for (a) hPancreas, (b) myeloid, and (c) 293T & Jurkat datasets. The model accurately transfers labels to query cells, recovering fine-grained subtypes and maintaining global structure. (d) Confusion matrices showing per-class annotation accuracy, with strong diagonal patterns indicating high predictive performance.

Here, $E_k^{(i)}$ denotes the gene embeddings from the Tabular Transformer layers. By concatenating the embedding of modality information $emb_m(t_m^{(i)})$, generated by a modality encoder given modality label $t_m^{(i)}$, the integrated embedding is passed to an MLP to predict the gene expression value $\hat{x}^{(i)}$ for cell i .

- **Cell Context-Aware Masked Gene Modeling (CMGM).** Following prior work [4], the CMGM loss improves optimization by predicting gene expression values through incorporating cell representation. This promotes contextual awareness between cells and genes, enhancing the efficacy of cell representation learning. Specifically, a gene-specific query vector q_j is created using the gene token embedding $E_{\text{token}}^{(i)}$ and calculates the predicted expression value using the

parameterized inner product between q_j and the cell representation $\mathbf{E}_{\text{cls}}^{(i)}$:

$$q_j = \text{MLP}(\mathbf{E}_{\text{token}}^{(i)}) \quad (33)$$

$$\hat{x}_j^{(i)} = q_j \cdot W \mathbf{E}_{\text{cls}}^{(i)} \quad (34)$$

$$\mathcal{L}_{\text{CMGM}} = \frac{1}{|\mathcal{M}_{\text{mask}}|} \sum_{j \in \mathcal{M}_{\text{mask}}} \left(\hat{x}_j^{(i)} - x_j^{(i)} \right)^2 \quad (35)$$

Here, $\mathbf{E}_{\text{cls}}^{(i)}$ denotes the cell-level embedding extracted from the [CLS] token. $\mathcal{L}_{\text{CMGM}}$ is computed as MSE loss between the predicted expression value and the true value, similar to \mathcal{L}_{MGM} .

Multi-Batch Integration To assess modeling performance on multi-batch integration, biological conservation metrics [28], especially the inverse of the average silhouette width for batch clustering ($\text{ASW}_{\text{batch}}$) and graph connectivity (Graph_Conn) are employed. The overall score, $\text{Avg}_{\text{batch}}$ is computed by averaging the two metrics.

During the fine-tuning phase, a domain adaptation (DA) loss is utilized to enable the model to optimize batch correction by reversing the backpropagation of learned batch labels. Following the methodology used by scGPT [4], we initiate an MLP classifier that predicts the batch label based on the cell representation $\mathbf{E}_{\text{cls}}^{(i)}$. Through a reverse back-propagation procedure, the model will aggregate batch information by integrating the reversed gradients, thereby enhancing the robustness of the batch correction process. Cross-entropy loss is employed for model finetuning. The formula is illustrated as follows:

$$b_j^{(i)} = \text{MLP}(\mathbf{E}_{\text{cls}}^{(i)}) \quad (36)$$

$$\mathcal{L}_{\text{DA}} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(b_{j,c}^{(i)}) \quad (37)$$

Here, the MLP layer outputs the predicted probabilities of batch labels $b_j^{(i)}$ using the cell representation $\mathbf{E}_{\text{cls}}^{(i)}$ as the input. The domain adaptation loss, \mathcal{L}_{DA} , is computed as the negative categorical cross-entropy loss, where N is the total number of samples in a batch, C is the total number of batch classes, $y_{i,c}$ is the true label for i -th sample for class c , in one-hot encoded form, and $b_{j,c}^{(i)}$ is the predicted probability for the i -th sample that belongs to batch class c .

We conducted a comprehensive evaluation of multi-omics and multi-batch integration tasks within a unified framework, with four losses: \mathcal{L}_{rec} , \mathcal{L}_{MGM} , $\mathcal{L}_{\text{CMGM}}$, and \mathcal{L}_{DA} , optimized together according to the batch and omic features in the dataset.

E.4.2 Implementation Details

For multi-omics and multi-batch integration tasks, we selected 2,000 HVGs from the dataset and randomly chose 1,200 genes, which is the maximum input length of the pretrained TABULA, for each training epoch. The data preprocess steps included TPM normalization, log1p transformation, HVG selection and value binning. During finetuning, we used an initial learning rate of 0.0001 and a weight decay of 0.95. For datasets with multi-omics, the model was extended to accommodate new tokens for new omics modality similar to gene embeddings, the mask ratio of \mathcal{L}_{MGM} and $\mathcal{L}_{\text{CMGM}}$ was set to 0.4, and the corruption rate of $\mathcal{L}_{\text{reconstruction}}$ was set to 0.6. For datasets with multi-batch, \mathcal{L}_{DA} was adopted with the weight of 1. The four losses were applied to both tasks if the dataset has multiple batches and multiple omics. Gradient clipping with a value of 0.5 was enabled to avoid exploding gradients. The model was trained with an early stopping strategy. The best performance of each dataset is reported based on the best-combined validation loss.

E.4.3 Evaluation Metrics

To evaluate the performance of integration tasks, we adopt the widely used evaluation pipeline in the field [28] and selectively choose five metrics as conducted by scGPT [4]. For multi-omics integration, metrics that measure conservation of biological variance of cell type, including NMI_{cell} , ARI_{cell} , and ASW_{cell} , are considered. For multi-batch integration, graph connectivity (Graph_Conn) and $\text{ASW}_{\text{batch}}$ are used for evaluation.

NMI (Normalized Mutual Information) measures the agreement between predicted and true cluster labels, providing an indication of how well cell types are preserved post-integration. It ranges from 0 (no agreement) to 1 (perfect agreement).

ARI (Adjusted Rand Index) evaluates the similarity between clustering results and true cell type annotations, adjusted for random chance. A higher ARI indicates better preservation of biological groupings.

ASW (Average Silhouette Width) quantifies the separation of cells within their assigned clusters compared to other clusters. A higher ASW indicates better-defined and more biologically meaningful clusters.

The batch-specific ASW evaluates how well batch effects are removed while maintaining meaningful biological groupings. A lower batch ASW indicates better integration across different batches.

Graph connectivity measures the continuity of the integrated data in a shared embedding space. Higher connectivity reflects improved integration, ensuring that cells from the same biological group form a cohesive cluster regardless of batch origin.

For systematically assessing the performance of integration tasks, *AvgBIO* and *Avgbatch* are employed to evaluate multi-omics integration and multi-batch integration, respectively. Additionally, *Overall* is used to measure the comprehensive performance of both tasks when the dataset is used to simultaneously evaluate the two tasks. The formulas are defined as follows:

$$AvgBIO = \frac{ARI_{cell} + NMI_{cell} + ASW_{cell}}{3} \quad (38)$$

$$Avgbatch = \frac{ASW_{batch} + Graph_Conn}{2} \quad (39)$$

$$Overall = 0.6 \cdot AvgBIO + 0.4 \cdot Avgbatch \quad (40)$$

E.4.4 Additional Results

We further benchmarked TABULA against state-of-the-art foundation models—scGPT, Geneformer, and scBERT—on both multi-omics integration and multi-batch correction tasks across diverse datasets. As shown in Figure 11, TABULA consistently outperforms or matches competing models in both multi-omics integration and batch effect removal.

Figure 11a–c presents UMAP visualizations on three representative datasets: 10x Multiome PBMC, BMBC, and DC. TABULA achieves higher AvgBIO scores, reflecting better preservation of biological identity while maintaining clear boundaries across fine-grained cell types. In particular, TABULA maintains high fidelity in distinguishing closely related populations. Figure 11d compares batch integration performance on the BMBC dataset. TABULA shows improved batch mixing while preserving biological structure, achieving an AvgBATCH score of 0.86, outperforming Geneformer. Figure 11e summarizes quantitative evaluations across four benchmark datasets using six commonly used metrics: AvgBATCH, AvgBIO, Graph Connectivity (GraphConn), Normalized Mutual Information (NMI), Average Silhouette Width (ASW), and Adjusted Rand Index (ARI). TABULA consistently achieves the highest or comparable performance across all metrics, demonstrating its robustness in complex multi-omics and multi-batch scenarios.

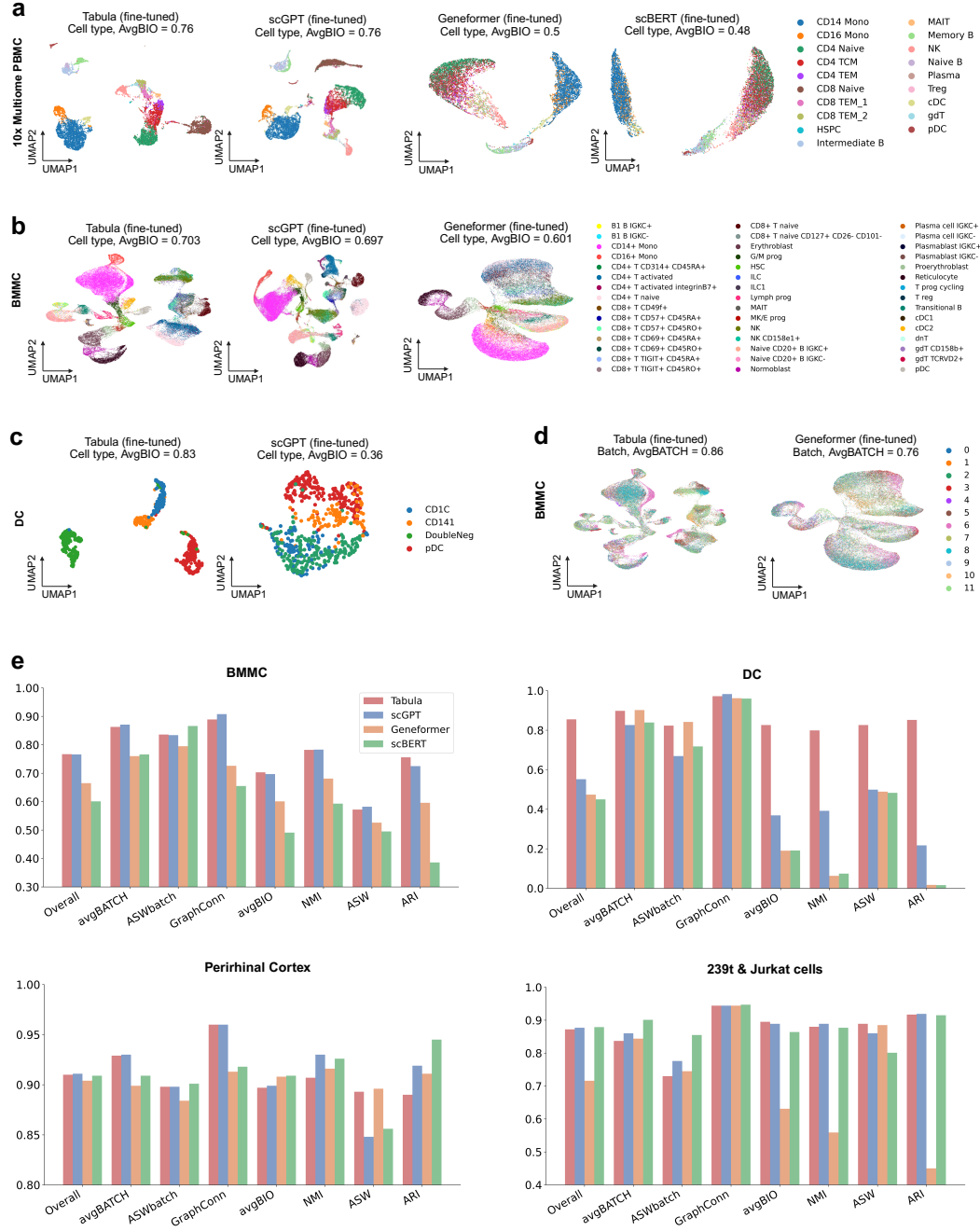


Figure 11: (Task 4&5: Multi-Omics & Multi-Batch Integration) (a–c) UMAPs of cell type integration across diverse datasets: (a) 10x Multiome PBMC, (b) BMNC, and (c) DC. (d) UMAPs of batch integration performance on BMNC. (e) Quantitative comparison across four datasets using standard integration metrics, including avgBIO, avgBATCH, GraphConn, NMI, ASW, and ARI. TABULA consistently achieves competitive or superior performance across both biological and batch-effect-related metrics.

F Tissue-Specific Embedders Encode Distinctive Tissue Features.

To investigate whether each client of the TABULA captures tissue-specific context, we compared embeddings generated using different tissue-trained encoders. As shown in Figure 12, tissue-specific embedders produce systematically distinct representations even for the same cell type or gene. Figure 12a presents cosine similarity distributions between cell embeddings from the same cell type, either generated by the same tissue-specific embedder (blue) or by different tissue embedders (orange). Across multiple cell types, within-tissue embeddings consistently show statistically higher similarity, indicating that the embedder captures tissue-specific structure. Figure 12b extends this analysis to the gene level. For each gene, we measure the cosine similarity between its embeddings in cells of the same type but derived from different tissue embedders. Genes such as AGMO, COBLL1, PWRN1, LINC00051, and ARHGAP24 exhibit statistically significant shifts in representation, demonstrating that the gene embeddings are also modulated by tissue-specific context.

Figure 12c compares multi-omics integration performance on brain tissue using three types of embedders: a random baseline, a lung-specific model, and a brain-specific model. UMAP projections reveal that the brain-specific embedder achieves the most biologically coherent clustering, with the highest AvgBIO score (0.924), demonstrating the importance of contextual pretraining for downstream accuracy. Together, these results suggest that TABULA's embedders encode meaningful tissue-specific variations, enhancing biological fidelity in both cell- and gene-level representations.

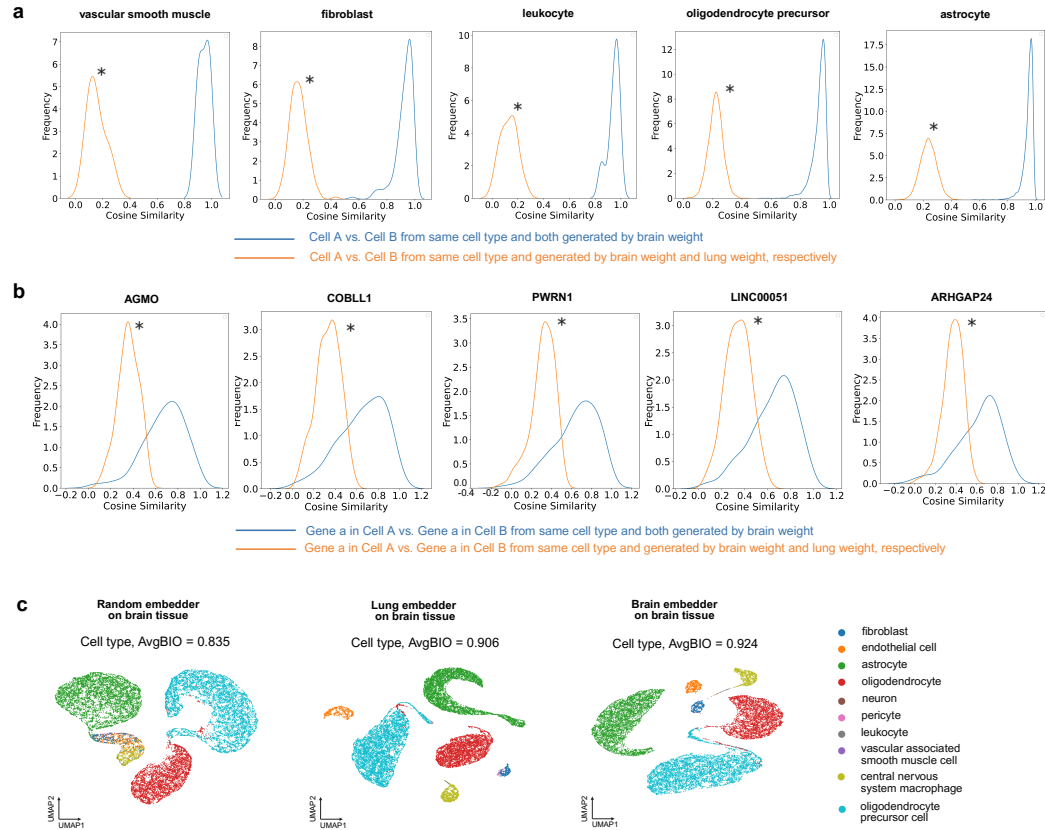


Figure 12: *Tissue-specific embedders encode distinctive tissue features.* (a) Cosine similarity distributions between cell embeddings within the same cell type generated by the same tissue embedder (blue) versus different tissue embedders (orange). The distribution from the blue lines is significantly shifted to the right (* indicates $p < 0.05$ by Wilcoxon test, FDR-corrected) highlights the tissue-specific TABULA models learn unique cellular molecular profiles of tissues. (b) Same as panel a but for comparing the cosine similarity of gene representations. (c) Multi-omics integration performance finetuned using three embedders: a random baseline, a lung-specific embedder, and a brain-specific embedder.

G Ablation Studies on Tabular Modeling Losses

Our foundation model, TABULA, is pretrained using two core loss functions: a reconstruction loss and a contrastive loss. To assess the individual contributions of these objectives, we conducted an ablation study in which three versions of TABULA are pretrained using: (1) contrastive loss only, (2) reconstruction loss only, and (3) both losses combined.

Pretraining is performed in a federated learning setting using 1 million cells sampled from the CELLxGENE collection, comprising 250,000 cells each from pancreas, blood, brain, and lung tissues. The federated infrastructure mirrors the setup used in prior experiments described in Appendix C.

To evaluate the downstream effectiveness of each loss configuration, we fine-tune the pretrained models on both gene-level and cell-level tasks. For gene-level evaluation, we use the gene imputation task across three benchmark datasets, PBMC5K, hPancreas, and Jurkat. For cell-level evaluation, we apply the models to a cell type annotation task using the Mye dataset.

Across all gene-level datasets (Table 3), the model trained with the combined contrastive and reconstruction loss consistently achieve the best or near-best performance, with the lowest RMSE and MAE, and the highest Pearson correlation. These results indicate that the combined objective not only improves reconstruction accuracy but also enhances alignment with biologically meaningful expression patterns. In particular, on the PBMC5K and Jurkat datasets, the combined loss substantially outperforms the single-loss models across all metrics, highlighting its ability to improve generalization and representation quality. **Corruption strategy.** Using *corrupted* inputs (resampling from the empirical marginal) is consistently stronger than *masking* for PBMC5K and hPancreas across all metrics, and on Jurkat it yields better RMSE/MAE. **Corruption rate.** At the same ratio, *corrupted* + *contrastive* consistently outperforms *mask* across all datasets. For a fixed ratio, using *corrupted* is more effective than *mask*.

In the cell-level classification task (Table 4), the contrastive-only model achieves the highest precision, reflecting its strength in learning discriminative characteristics. However, the combined-loss model yielded the best F1 score, along with the highest precision and recall, suggesting greater robustness and improved performance on imbalanced classes. This supports the complementary roles of the two losses: cell-wise contrastive loss enhances class separability, while gene-wise reconstruction loss promotes fidelity to underlying gene features.

Taken together, these findings justify the integration of both contrastive and reconstruction losses in TABULA’s pretraining strategy, resulting in a unified model that consistently achieves strong performance across both gene-level tasks (e.g., gene imputation) and cell-level tasks (e.g., cell type annotation).

Table 3: (*Gene-level: Gene Imputation*) Ablation results of TABULA on the gene imputation task across three datasets using different loss configurations. Lower RMSE↓ and MAE↓ indicate better accuracy, while higher Pearson correlation↑ reflects better alignment with ground truth expression profiles. “corrupted” denotes resampling selected gene values from the empirical marginal distribution used in TABULA, whereas “mask” denotes setting selected gene values to -1 (traditional masking). “corrupted 0.6” and “mask 0.6” indicate a corruption/masking ratio of 0.6.

Ablation	PBMC5K			hPancreas			Jurkat		
	RMSE↓	MAE↓	Pearson↑	RMSE↓	MAE↓	Pearson↑	RMSE↓	MAE↓	Pearson↑
Loss Ablation									
TABULA (only contrastive)	1.1431	0.9715	0.4872	0.7870	0.5328	0.0108	0.9542	0.8314	0.6043
TABULA (only reconstruction)	0.9856	0.8457	0.5647	0.5748	0.3355	0.3386	0.7403	0.6096	0.5848
TABULA (contrastive+reconstruction)	0.9298	0.8016	0.5934	0.5730	0.3226	0.3255	0.6295	0.5138	0.6511
Corruption Strategy Ablation									
TABULA (corrupted 0.6)	0.9856	0.8457	0.5647	0.5748	0.3355	0.3386	0.7403	0.6096	0.5848
TABULA (mask 0.6)	1.1978	1.0469	0.4244	0.6778	0.3985	0.2154	1.0268	0.9274	0.6970
Corruption Rate Ablation									
TABULA (corrupted 0.6 + contrastive)	0.9298	0.8016	0.5934	0.5730	0.3226	0.3255	0.6295	0.5138	0.6511
TABULA (mask 0.6)	1.1978	1.0469	0.4244	0.6778	0.3985	0.2154	1.0268	0.9274	0.6970
TABULA (corrupted 0.15 + contrastive)	0.9913	0.8587	0.5728	0.5881	0.3264	0.2791	0.7593	0.6088	0.5313
TABULA (mask 0.15)	1.2498	1.0930	0.3569	0.6893	0.3965	0.1821	1.0530	0.9571	0.6960

Table 4: (*Cell-level: Cell Type Annotation*) Ablation results of TABULA on the cell type annotation task on the Mye dataset using different loss configurations.

Mye dataset	Accuracy \uparrow	Precision \uparrow	Recall \uparrow	F1 \uparrow
TABULA (only contrastive)	0.6163	0.355	0.3249	0.3315
TABULA (only reconstruction)	0.5924	0.3331	0.3123	0.3188
TABULA (contrastive+reconstruction)	0.6093	0.3794	0.349	0.3583

H Downstream Task Datasets

- **CELLxGENE**

The CELLxGENE dataset is a publicly accessible resource developed and maintained by the Chan Zuckerberg Initiative, designed for storing, exploring, and analyzing scRNA-seq data. It aggregates data from research projects across the globe. Through the CELLxGENE portal, we collect scRNA-seq data from approximately 40 million human cells, and the release version from July 25, 2023. After quality control, preprocessing, and filtering, about 15 million human scRNA-seq data are obtained for TABULA pretraining. This comprehensive dataset includes a diverse range of 23,156 genes and 422 cell types from 133 tissues and 196 studies.

- **PBMC 5K**

We use a high-throughput scRNA-seq dataset, PBMC 5K from 10x Genomics, containing samples of peripheral blood mononuclear cells (PBMCs) from healthy donors. These cells are tagged with a panel of TotalSeq-B antibodies (v3 chemistry). The dataset comprises a total of 5,247 cells and includes 33,538 genes.

- **Jurkat**

The Jurkat cells are human T-lymphocyte cell lines derived from patients with acute T-cell leukemia and are widely used in immunology research. This dataset provides scRNA-seq data for Jurkat cells generated by the 10x Genomics. The dataset comprises a total of 3,258 cells and includes 32,738 genes.

- **Melanoma**

The Melanoma dataset centers on scRNA-seq of WM989-A6-G3 melanoma cells using two technologies: DropSeq and Fluidigm’s C1 mRNA Seq HT chip [16]. These methods allow a detailed analysis of the transcriptomic characteristics of melanoma cells. We select data sequenced using DropSeq technology. The dataset comprises a total of 8,640 cells and includes 32,287 genes.

- **Adamson**

The Adamson perturbation is a scRNA-seq dataset derived from K562 human cell lines [14], designed to study the effects of CRISPR interference on gene expression. The dataset contains RNA expression profiles for 65,337 raw cells, with 43,550 cells retained after quality filtering. It includes measurements for 32,738 genes across 87 genetic perturbations, following filtering from 90 initial perturbations. The dataset is preprocessed by GEARS data processing function [18] for further finetuning.

- **Norman**

The Norman perturbation dataset is also a scRNA-seq dataset generated from K562 human cell lines [15], designed to explore genetic interactions through CRISPR activation. The dataset captures RNA expression profiles from 111,668 raw cells, with 82,081 cells retained after quality filtering. It contains measurements for 33,694 genes across 105 single-gene perturbations and 131 dual-gene perturbations. The dataset is preprocessed by GEARS data processing function [18] for further finetuning.

- **Replogle**

The Replogle perturbation dataset uses a multiplexed CRISPR interference with single-cell transcriptome sequencing to analyze the effects of gene knockdowns across cell types, creating a comprehensive human cell genotype-phenotype atlas [17]. The dataset has been meticulously processed based on the curated protocols of scGPT [4], which includes a total of 171,542 samples, derived from 1,823 distinct one-gene perturbations, of which 99 target transcription factors. Additionally, the dataset features a test set consisting of 456

perturbations, with 25 specifically affecting transcription factors. The dataset is preprocessed by GEARS data processing function [18] for further finetuning.

- **Human Pancreas**

The Human pancreas dataset comprises data from five scRNA-seq studies on human pancreas cells, reprocessed by [13] for cell type annotation. In total, there are 14 cell types including alpha, beta, ductal, acinar, delta, pancreatic stellate, pancreatic polypeptide, endothelial, macrophage, mast, epsilon, Schwann, T cell, and MHC class III. The reference set includes data from two sources containing 10,600 cells across 13 cell types, while the query data covers the remaining three sources consisting of 4,218 cells spanning 11 cell types. The query set includes MHC class II, which is absent in the reference set, but lacks macrophage, Schwann, pancreatic polypeptide, and T cell, which are present in the reference set.

- **Myeloid**

The Myeloid dataset provides a comprehensive analysis of subpopulation profiling of tumour-infiltrating myeloid cells at the pan-cancer level, by integrating single-cell transcriptomic data from nine different cancer types [29]. The dataset was randomly subsampled by [4]. (2024). The reference set contains 9,748 cells from six cancer types including UCEC, PAAD, THCA, LYM, cDC2, and kidney, while the query set contains 3,430 cells from four cancer types including MYE, OV-FTC, and ESCA.

- **293T & Jurkat cells**

The 293T & Jurkat cells dataset, generated using the 10x Genomics platform, profiles gene expression across 16,602 genes in two distinct cell lines: 293T cells, originating from human embryonic kidney and commonly used in gene expression and viral studies, and Jurkat cells, a T lymphocyte line derived from a leukemia patient and frequently employed in T cell signaling research. As reported by [20], the dataset comprises three experimental batches: Batch 1 consisting of 2,885 293T cells, Batch 2 containing 3,258 Jurkat cells, and Batch 3 featuring a balanced mixture of both cell types with 3,388 cells equally distributed between 293T and Jurkat cell lines.

- **Liver**

The Liver dataset contains a variety of human liver cell types from two primary sources. The reference dataset, processed by [30], contains 8,444 parenchymal and non-parenchymal cells from five human livers. It includes 14 different cell types, including central venous sinusoidal endothelial cells, erythroid cells, hepatocytes, inflammatory and non-inflammatory macrophages, mature B cells, natural killer (NK) cells, periportal sinusoidal endothelial cells, plasma cells, portal endothelial cells, satellite cells, alpha-beta T cells, cholangiocytes, and gamma-delta T cells. While the query dataset provided by [31], consists of 9,162 cells from normal human liver tissue, it overlaps with 7 of the cell types found in the reference dataset.

- **10x Multiome PBMC**

The 10x Multiome PBMC dataset captures dual-modality single-cell profiles from human peripheral blood mononuclear cells through paired RNA and ATAC sequencing analysis. Processed by [32], this dataset contains 9,631 cells derived from a healthy 25-year-old female donor, with granulocytes removed via cell sorting before nuclear isolation and sequencing. The dataset spans 29,095 genes for expression analysis and 107,194 regions for chromatin accessibility measurements. The cells are classified into 19 distinct immune populations, encompassing various T cell subtypes ($CD4^+$ naive, $CD4^+$ TCM, $CD4^+$ TEM, $CD8^+$ naive, $CD8^+$ TEM 1, $CD8^+$ TEM 2, MAIT, Treg, gdT), B cell subtypes (naive, intermediate, memory, plasma), myeloid cells ($CD14^+$ monocytes, $CD16^+$ monocytes, cDC, pDC), as well as NK cells and HSPCs.

- **BMMC**

The BMMC dataset is a comprehensive multimodal single-cell profiling of bone marrow mononuclear cells from 12 healthy human donors [33], originally designed for the Multimodal Single-Cell Data Integration Challenge at NeurIPS 2021. It includes paired single-cell RNA and protein abundance measurements across 12 batches. Processed by [4], the final dataset consists of 90,261 cells, with data on 13,953 genes and 134 surface proteins, along with detailed annotations for 45 immune cell subtypes.

- **Perirhinal Cortex**

The Perirhinal Cortex dataset is a subset of a larger study by [34], which created a compre-

hensive transcriptomic atlas of the human brain using over three million nuclei from around 100 dissections across various brain regions. From this dataset, two separate batches from the perirhinal cortex were selected for further analysis, following the methodology outlined by scGPT [4]. The first batch contains 8,465 cells, while the second batch includes 9,070 cells. Together, the datasets comprise a total of 59,357 genes, with the ten unique cell types identified in the original study being used for analysis.

- **DC**

This DC dataset contains scRNA-seq data from human blood dendritic cells (DCs), originally collected by [35] and processed by [20]. It includes two batches, each with four distinct cell types: CD1C DC, CD141 DC, plasmacytoid DC (pDC), and double negative cells. To ensure non-identical cell compositions across batches, CD1C DCs are excluded from batch 1, and CD141 DCs are excluded from batch 2. Each batch contains 288 cells and 16,594 genes, with 96 pDCs and 96 double negative cells shared between the batches. Batch 1 includes 96 CD141 cells, while batch 2 includes 96 CD1C cells.

- **Hematopoiesis**

The Hematopoiesis dataset is derived from [36]. The dataset consists of 1,947 cells representing 8 cell types across five developmental nodes. These cell types include: hematopoietic stem cells (HSC), megakaryocyte-erythrocyte progenitors (MEP), granulocyte-monocyte progenitors (GMP), monocytes (MON), neutrophils (Neu), megakaryocytes (Meg), erythrocytes (Ery), and basophils (Bas). Based on the known cell differentiation [37], HSC first differentiates into MEP and GMP, with GMP further developing into Mon and Neu, while MEP diverts into Meg, Ery, and Bas.

- **Cardiogenesis**

The Cardiogenesis dataset is obtained from the cardiogenic region of mouse embryos [38]. The cardiogenic mesoderm differentiates from a shared population of cardiovascular progenitor cells into two regions with distinct gene expression: the first heart field (FHF) and the second heart field (SHF). We focus on the gene regulatory relationships within the FHF and SHF. A total of 1,165 cells are selected from this dataset for analysis, comprising cells from both the posterior second heart field (pSHF) and FHF.

- **Neurogenesis**

The Neurogenesis dataset is a comprehensive single-cell transcriptomic atlas of the embryonic mouse brain between gastrulation and birth [39]. We focus on the transformation of glial cells into astrocytic and oligodendrocytic lineages within the central nervous system. We select a total of 4,704 cells from the atlas, which include 2,516 mixed region astrocytes, 1,990 oligodendrocyte precursor cells, 162 committed oligodendrocyte precursor cells, and 36 mature oligodendrocytes.

- **Pancreatic Endogenous**

The Pancreatic Endogenous dataset is based on the work of [40]. We aim to investigate the master transcription factors that govern pancreatic cell fate during the development of the pancreas. A total of 14,445 cells are selected from the dataset across three time points: 4,631 cells at E12, 5,179 cells at E14, and 4,635 cells at E17.

I Limitations in TABULA

Despite the promising results and design innovations introduced by TABULA, several limitations remain:

1. **Federated Fine-Tuning Is Not Fully Demonstrated**

While TABULA adopts federated learning during pretraining to preserve data locality and privacy, all downstream fine-tuning experiments are conducted in a centralized manner. This creates a gap in understanding how well TABULA adapts to real-world deployment scenarios that require end-to-end federated workflows, such as fine-tuning on private datasets from individual hospitals or research institutions. Future work should explore fine-tuning protocols under federated settings to assess performance consistency and adaptability.

2. **Limited Support for Spatial or Temporal Single-Cell Modalities**

The current version of TABULA focuses exclusively on transcriptomic profiles from scRNA-seq data. However, emerging single-cell technologies increasingly include spatial context (e.g., spatial transcriptomics) [41] and temporal resolution (e.g., time-course or lineage-traced data) [42]. It remains unclear how tabular modeling can be extended or adapted to effectively pretrain on such multimodal data. Incorporating spatial coordinates or temporal dependencies within the tabular framework presents both architectural and algorithmic challenges that warrant further investigation.

3. **Vulnerability to Adversarial Attacks in Federated Training**

Although federated learning reduces the risk of direct data leakage, the current TABULA implementation does not incorporate formal defenses against adversarial threats [43, 44]. Malicious clients in the federated setup could perform poisoning or backdoor attacks [45]. Future versions of TABULA should consider integrating adversarial robustness techniques, such as secure aggregation, differential privacy [46], or robust aggregation schemes, to mitigate these risks in high-stakes biomedical applications.

J Broader Impact of TABULA

We propose TABULA, a foundation model for single-cell transcriptomics built upon three key innovations, each with significant societal and scientific implications:

1. Pioneering Self-Supervised Tabular Modeling for Single-Cell Data

Existing single-cell foundation models [1, 2, 3, 4, 5] are largely inspired by natural language processing, treating gene expression profiles as sequences of gene tokens. However, unlike natural language, gene expression data lacks meaningful order and combines discrete gene identities with continuous expression values. To the best of our knowledge, TABULA is the first foundation model to treat the cell-by-gene matrix as a table and apply self-supervised tabular modeling, capturing both gene-wise and cell-wise structure through dual-axis learning. We hope this work initiates a new direction for modeling single-cell data as tabular input. Despite using only half the pretraining data, TABULA achieves state-of-the-art performance across a wide range of gene-level tasks (e.g., gene imputation, perturbation prediction) and cell-level tasks (e.g., cell type annotation, multi-omics integration, and batch correction).

2. Improved Biological Realism Through Statistically Grounded Corruption

Prior models, such as Geneformer [1] and scFoundation [2], use artificial sentinel values (e.g., masking with -1) to corrupt input data during training, values that never appear during inference, thus creating a mismatch that harms generalization. TABULA addresses this by corrupting data through resampling from the empirical marginal distribution of gene expression values, leading to biologically realistic training and improved downstream performance.

3. Enabling Privacy-Preserving Foundation Modeling via Federated Learning

The cell-by-gene matrix has been shown to leak private information through linking attacks [6] (More details are provided in Appendix D), posing serious risks as datasets grow in size and resolution. TABULA introduces a federated learning framework that enables institutions or tissues to collaboratively train models without exposing raw data. This not only addresses pressing privacy concerns in single-cell research but also demonstrates the feasibility and effectiveness of training large-scale foundation models in federated, decentralized environments.

Together, these contributions highlight TABULA’s potential to advance the scientific utility, privacy safeguards, and ethical deployment of foundation models in the field of single-cell.