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

Jiayuan Ding[*†]
Stanford University

Jianhui Lin[*]
Central South University

Shiyu Jiang[*]
University of Southern California

Yixin Wang
Stanford University

Ziyang Miao
Central South University

Zhaoyu Fang
Central South University

Jiliang Tang[†]
Michigan State University

Min Li[†]
Central South University

Xiaojie Qiu[†]
Stanford University

Abstract

Foundation models (FMs) have shown great promise in single-cell genomics, yet current approaches, such as scGPT, Geneformer, and scFoundation, rely on centralized training and language modeling objectives that overlook the tabular nature of single-cell data and raise significant privacy concerns. We present TABULA, a foundation model designed for single-cell transcriptomics, which integrates a novel tabular modeling objective and federated learning framework to enable privacy-preserving pretraining across decentralized datasets. TABULA directly models the cell-by-gene expression matrix through column-wise gene reconstruction and row-wise cell contrastive learning, capturing both gene-level relationships and cell-level heterogeneity without imposing artificial gene sequence order. Extensive experiments demonstrate the effectiveness of TABULA: despite using only half the pretraining data, TABULA achieves state-of-the-art performance across key tasks, including gene imputation, perturbation prediction, cell type annotation, and multi-omics integration. It is important to note that as public single-cell datasets continue to grow, TABULA provides a scalable and privacy-aware foundation that not only validates the feasibility of federated tabular modeling, but also establishes a generalizable framework for training future models under similar privacy-preserving settings. All resources are openly available at `https://github.com/aristoteleo/tabula` to support broad community adoption and future methodological advances.

1 Introduction

Over the past year, we have witnessed the emergence of single-cell foundation models, notably Geneformer [1], scFoundation [2], CellPLM [3], scGPT [4], and NicheFormer [5]. However, current models fail to explicitly account for the tabular structure of the scRNA-seq data, often naively converting gene expression within a single cell to gene sequences to mimic word sequences used in NLP. While this adapts the NLP paradigm to single-cell data, it overlooks critical features of the underlying tabular structure of scRNA-seq. Moreover, it has been shown previously that scRNA-seq data is susceptible to privacy breaches through linking attacks [6, 7] (More details are provided in Appendix D), where information from one study can be linked to another to identify private data.

[*] Equal contribution. [†] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

This privacy concern becomes even more pronounced as we begin creating foundation models with tens of millions of single cells and datasets from thousands of individuals.

To address these critical gaps, we introduce TABULA, a foundation model specifically designed for single-cell data through the lens of tabular learning and federated training. TABULA introduces four key innovations:

First, TABULA formulates single-cell pretraining as a self-supervised tabular learning problem, directly modeling the cell-by-gene tabular structure. It is important to note that TABULA uses corrupted gene expression inputs instead of introducing artificial mask values such as -1 in traditional FMs to learn robust gene-level representations through column-wise reconstruction, and cell-level representations through row-wise contrastive learning. This tabular modeling strategy captures both gene dependencies and cell-level heterogeneity, without imposing artificial gene order.

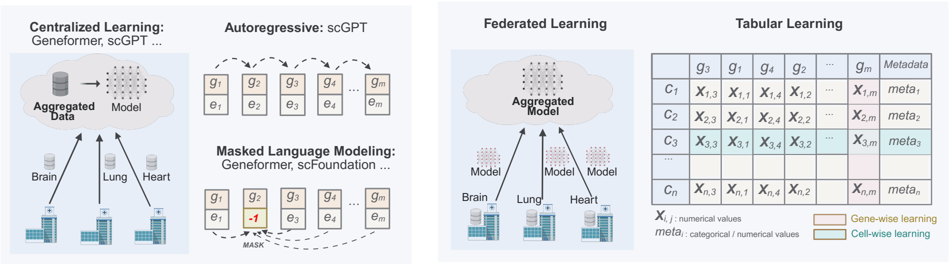
Second, TABULA is trained in a federated learning setting not only to preserve data privacy across institutions or tissue datasets, but also to enable tissue-specific embeddings by decoupling the training of a shared tabular transformer from client-specific embedders. This design allows TABULA to capture both globally shared biological patterns and unique transcriptional features of each client tissue.

Third, extensive experiments demonstrate the effectiveness of TABULA. TABULA achieves state-of-the-art performance across a wide range of benchmark tasks, including gene imputation, genetic perturbation prediction, cell type annotation, multi-omics integration, and batch correction while requiring only half the data for pretraining.

Fourth, as public single-cell datasets continue to grow, the need for scalable and privacy-preserving training becomes increasingly important. TABULA provides a generalizable framework that combines federated learning with tabular modeling, enabling collaborative model training without data sharing. This not only validates the feasibility of federated tabular modeling but also lays the foundation for training future models under similar privacy-preserving settings.

2 Design Principles

Despite recent progress in single-cell foundation models [8, 4, 1, 2], existing approaches face critical limitations that hinder their biological fidelity, scalability, and privacy. These limitations motivate the design of TABULA, which is guided by the following three principles.


Centralized Learning:
Geneformer, scGPT ...
Aggregated Data
Model
Brain
Lung
Heart
Autoregressive: scGPT
g_1 g_2 g_3 g_4 ... g_m
e_1 e_2 e_3 e_4 ... e_m
Masked Language Modeling:
Geneformer, scFoundation ...
g_1 g_2 g_3 g_4 ... g_m
e_1 -1 e_3 e_4 ... e_m
MASK
Federated Learning
Aggregated Model
Model
Model
Model
Brain
Lung
Heart
Tabular Learning

	g_3	g_1	g_4	g_2	...	g_m	Metadata
c_1	$x_{1,3}$	$x_{1,1}$	$x_{1,4}$	$x_{1,2}$	...	$x_{1,m}$	$meta_1$
c_2	$x_{2,3}$	$x_{2,1}$	$x_{2,4}$	$x_{2,2}$	...	$x_{2,m}$	$meta_2$
c_3	$x_{3,3}$	$x_{3,1}$	$x_{3,4}$	$x_{3,2}$	...	$x_{3,m}$	$meta_3$
...							
c_n	$x_{n,3}$	$x_{n,1}$	$x_{n,4}$	$x_{n,2}$		$x_{n,m}$	$meta_n$

$x_{i,j}$: numerical values
$meta_i$: categorical / numerical values
Gene-wise learning
Cell-wise learning

(a) Existing Foundation Models

(b) TABULA

Figure 1: An illustration of the difference between (a) existing foundation models and (b) TABULA for single-cell modeling.

Principle 1: Realistic Corruption Strategies. Existing FMs such as Geneformer [1] and scFoundation [2] often use artificial corruption strategies (e.g., masking gene expression values with -1) for training (Figure 1a, right). These unrealistic sentinel values do not appear during inference and result in a training-inference mismatch that degrades generalization. TABULA avoids this issue by generating corrupted views through resampling from the empirical marginal distribution of gene values (Figure 2), preserving statistical realism and improving downstream biological performance.

Principle 2: Tabular Modeling for Unordered Mixed-Type Single-Cell Data. Unlike natural language, where sequence order is meaningful, gene expression profiles lack an inherent ordering and consist of mixed discrete gene token IDs and continuous gene expression values. Most existing models artificially impose gene sequences via attention masking (scGPT [4]) or ranking (Geneformer [1]) (Figure 1a, right), which distorts the tabular nature of the data. In contrast, TABULA treats the cell-by-gene matrix as a table and learns both gene-level dependencies via column-wise reconstruction and cell-level variation via row-wise contrastive learning (Figure 1b, right). This dual-axis self-supervised learning respects the true tabular structure of the data.

Principle 3: Federated Learning for Privacy-Preserving Collaboration. Centralized training poses major privacy risks in genomics, where individual-level data is sensitive and protected. Prior work has shown that scRNA-seq data is vulnerable to linkage attacks [6, 7]. To mitigate these risks, TABULA is trained using a federated learning setup (Figure 1b, left). Each client (e.g., tissue or institution) retains its data locally and contributes to a shared global model by only exchanging weights, enabling collaborative training across heterogeneous datasets without data sharing.

These three design principles, realistic corruption, tabular structure awareness, and federated privacy, form the foundation of TABULA and are reflected in the framework described in Section 3.

3 The Proposed TABULA Framework

In this section, we introduce the proposed TABULA framework and each component in TABULA. As illustrated in Figure 2, the tabular modeling in TABULA requires corrupted gene expression inputs, which are detailed in Section 3.1. The process by which the tabular embedder transforms the cell-by-gene matrix into embeddings for the transformer, along with the modeling of cell-by-gene structure from a tabular learning perspective, is detailed in Section 3.2. At a high level, TABULA consists of two stages: pre-training and fine-tuning, which are demonstrated in Section 3.3.

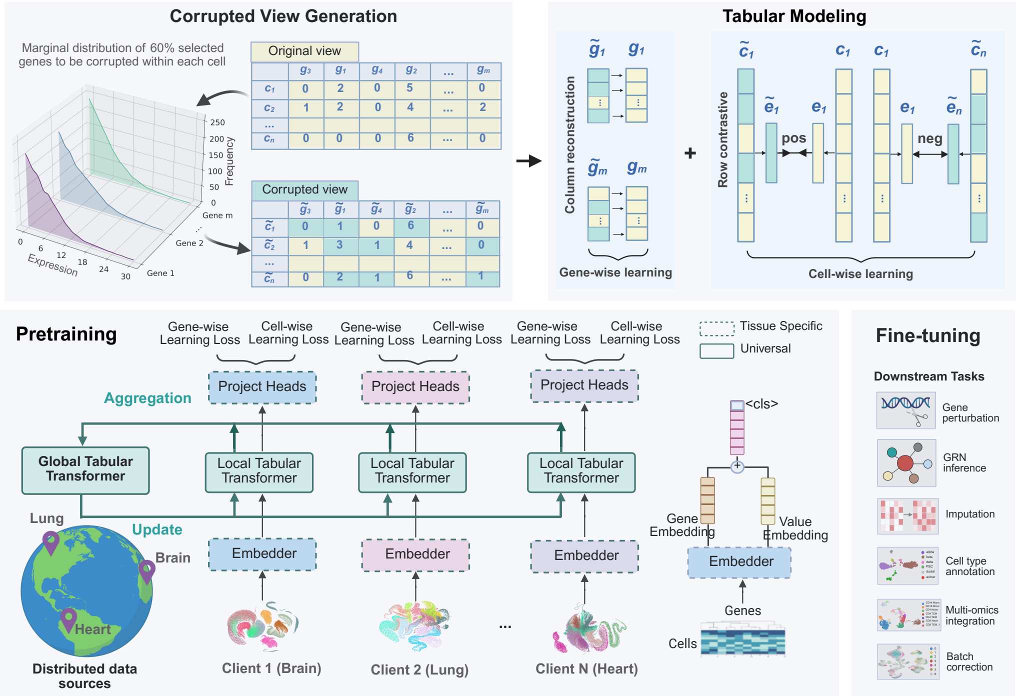
3.1 Corrupted View Generation for Tabular Modeling

To handle the intrinsic tabular structure of scRNA-seq dataset, TABULA introduces an innovative self-supervised tabular modeling framework that consists of both gene-wise reconstruction learning and cell-wise contrastive learning. Both training components start with generating a corrupted representation of the scRNA-seq data. We followed Xtab [9] to construct corrupted views through random feature resampling. As shown in Figure 2, 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 the original view of each cell sample c and its corrupted view $\tilde{c}$, 60% of the gene values are resampled while 40% remain unchanged. The original and corrupted views of the same cell are treated as a positive pair in contrastive learning, as they originate from the same underlying biological state but differ through controlled noise.

3.2 TABULA Model Architecture

TABULA innovatively treats single-cell data, represented as a cell-by-gene matrix, as tabular data, and models its intrinsic tabular structure through a dual-axis self-supervised learning framework. Additionally, TABULA addresses cross-tissue variation by partitioning the model into distinct clients in a federated setting, each tissue client equipped with tissue-specific embedders that capture tissue-specific variants, and a shared tabular transformer (global or local) that encodes global knowledge across tissues. In this section, we describe how TABULA encodes the raw cell-by-gene matrix into transformer-compatible embeddings in **Tabular Embedder**, and how it models the matrix from a tabular learning perspective in **Tabular Modeling**.

Tabular Embedder scRNA-seq data can be represented as a cell-by-gene matrix, $X \in \mathbb{R}^{N \times M}$, where each entry $X_{i,j} \in \mathbb{R}^{+}$ denotes the read count of RNA molecules for gene $j \in \{0, 1, \ldots, M\}$ in cell $i \in \{0, 1, \ldots, N\}$. Each row of the cell-by-gene table, X, is considered as an input cell sample in a tabular transformer, and each column is a gene feature token. The function of the embedder module is to convert each cell sample to feature embeddings $\mathbf{E} \in \mathbb{R}^{(M+1) \times d}$. Here, M denotes the number of columns, $+1$ indicates the special [CLS] token (see below), and d is the embedding dimension. TABULA treats each row of the cell-by-gene matrix as an unordered set of genes, where each gene is represented by a column feature name embedding (gene token) and a feature value embedding



Original view

	g_3	g_1	g_4	g_2	...	g_m
c_1	0	2	0	5	...	0
c_2	1	2	0	4	...	2
...						
c_n	0	0	0	6	...	0

Corrupted view

	$\tilde{g}_3$	$\tilde{g}_1$	$\tilde{g}_4$	$\tilde{g}_2$	...	$\tilde{g}_m$
$\tilde{c}_1$	0	1	0	6	...	0
$\tilde{c}_2$	1	3	1	4	...	0
...						
$\tilde{c}_n$	0	2	1	6	...	1

Figure 2: Overview of the TABULA framework. TABULA learns both gene- and cell-level representations from tabular single-cell data using dual-axis self-supervised tabular learning objectives: column-wise gene reconstruction and row-wise cell contrastive learning. It is trained in a federated setting with a shared global transformer and client-specific embedders, enabling privacy-preserving, and tissue-aware representation learning.

(expression level). While the current implementation models only gene tokens, the framework can naturally incorporate metadata (e.g., cell type, tissue, or platform) in future extensions.

- **Column feature name embedding.** The feature column in the cell-by-gene table for each client corresponds to individual genes. Our pretraining dataset includes 196 distinct studies, from which we select 1,200 HVGs per study, yielding a total of 23,156 HVGs across all datasets. Consequently, we have 23,156 unique feature columns to learn during the pretraining phase. We assign each column feature name, $feature_j$, a unique integer identifier, $id(feature_j)$. These identifiers collectively form the column feature name vocabulary utilized in TABULA. Additionally, a special token, [CLS], is used to aggregate all column features into a cell-level representation. The input column feature tokens for cell i are therefore represented as a vector $\mathbf{f}_{\text{token}}^{(i)} \in \mathbb{R}^M$, where M is a predetermined maximum sequence length in the tabular transformer:

$$\mathbf{f}_{\text{token}}^{(i)} = \left[id\left(feature_1^{(i)}\right), id\left(feature_2^{(i)}\right), \ldots, id\left(feature_M^{(i)}\right) \right] \tag{1}$$

  The embedding representation of input column feature tokens for cell i is denoted as $\mathbf{E}_{\text{token}}^{(i)} \in \mathbb{R}^{M \times d}$ where d represents the embedding dimension:

$$\mathbf{E}_{\text{token}}^{(i)} = \text{emb}_{\text{token}}\left(\mathbf{f}_{\text{token}}^{(i)}\right) \tag{2}$$

  where $\text{emb}_{\text{token}}$ represents an embedding layer to learn feature token embeddings.

- **Column feature value embedding.** Expression values in scRNA-seq data vary due to differences in sequencing protocols, sensitivity, and depth, leading to inconsistencies across datasets. To tackle this issue, we adopt scGPT's value binning technique [4] to convert expression counts into relative values. Unlike scGPT, which only focuses on non-zero expression counts in each cell, we select 1,200 HVGs for modeling in each dataset, including those with zero expression.

By incorporating zero-expressed genes, TABULA provides a more complete view of gene expression variability, capturing both active and inactive genes. This broader perspective allows for the detection of important biological patterns that could be overlooked when considering only non-zero genes, as zero expression can have biological significance [11]. For each HVG expression count in a cell, we compute the raw absolute values and partition them into B consecutive intervals $[Bin_t, Bin_{t+1})$, where $t \in \{1, 2, \ldots, B\}$. The binned expression value $x_j^{(i)}$ for cell i is designated as:

$$x_j^{(i)} = \begin{cases} t, & \text{if } X_{i,j} > 0 \text{ and } X_{i,j} \in [Bin_t, Bin_{t+1}) \\ 0, & \text{if } X_{i,j} = 0 \end{cases} \tag{3}$$

The column feature values for cell i are represented as a vector $\mathbf{f}_{\text{val}}^{(i)} = \left[x_1^{(i)}, x_2^{(i)}, \ldots, x_M^{(i)}\right] \in \mathbb{R}^M$, where M is the predefined maximum sequence length. These values are then embedded via $\mathbf{E}_{\text{val}}^{(i)} = \text{emb}_{\text{val}}(\mathbf{f}_{\text{val}}^{(i)})$, where $\mathbf{E}_{\text{val}}^{(i)} \in \mathbb{R}^{M \times d}$ and emb_{val} is a learnable embedding layer.

Consequently, the embedding $\mathbf{E}^{(i)} \in \mathbb{R}^{M \times d}$ for cell i is represented as:

$$\mathbf{E}^{(i)} = \mathbf{E}_{\text{token}}^{(i)} + \mathbf{E}_{\text{val}}^{(i)} \tag{4}$$

Finally, the embedding of the [CLS] token is appended to the feature embedding, resulting in a final feature embedding $\mathbf{E}^{(i)} \in \mathbb{R}^{(M+1) \times d}$. The final hidden state corresponding to the [CLS] token is used as row (cell) representation, aggregating from column feature embeddings. It is important to note that each client's embedder module is distinct and specialized for learning tissue-specific features.

Tabular Modeling. To effectively model the cell-by-gene table, we design two specialized pretraining losses for tabular modeling: reconstruction loss and contrastive learning to learn gene column features and row cell features, respectively shown in Figure 2. To calculate the reconstruction and contrastive losses, a corrupted view must be generated for each cell, which has been discussed in Section 3.1. To note that the original view c and corrupted view $\tilde{c}$ of the same cell are treated as a positive pair in contrastive learning, as they originate from the same underlying biological state but differ through controlled noise.

- **Reconstruction loss as gene-wise column learning.** Reconstruction loss is a self-supervised training objective. It is important to note that it aims to recover the original gene-wise view from a corrupted view of that gene across all cells in the batch. Taking gene j as an example, we take the corrupted view, $\tilde{\mathbf{c}}_j \in \mathbb{R}^N$ (N indicates the number of cells in a batch.), for gene j as input and aim to reconstruct its original binned gene expression $\mathbf{g}_j \in \mathbb{R}^N$ where $\mathbf{g}_j = \left[c_j^{(1)}, c_j^{(2)}, \ldots, c_j^{(N)}\right]$. The reconstructed view for gene j can be represented as $\hat{\mathbf{g}}_j \in \mathbb{R}^N$. In this study, we use Mean Squared Error (MSE) to measure the distance between the original view and the reconstructed view. The reconstruction loss is denoted as:

$$\mathcal{L}_{\text{rec}} = \frac{1}{M \cdot N} \sum_{j=1}^{M} \sum_{i=1}^{N} \left(\hat{c}_j^{(i)} - c_j^{(i)}\right)^2 \tag{5}$$

  The reconstruction loss $\mathcal{L}_{\text{rec}}$ is applied to all M gene values, enabling the model to learn gene-specific distributions and inter-gene relationships for richer feature representations. Unlike prior models [8, 2, 1, 4] that use an artificial mask value (e.g., -1), we replace corrupted entries by resampling from the empirical gene distribution, aligning training conditions with inference for improved consistency.

- **Contrastive loss as cell-wise row learning.** Similar to the reconstruction objective, we generate $\tilde{c}^{(i)}$ as a corrupted view of cell $c^{(i)}$. The original and corrupted views of the same cell form a positive pair, while all other cells ($\mathbf{c}^{(j)}$ and $\tilde{\mathbf{c}}^{(j)}$) within the batch are treated as negative pairs. Contrastive loss encourages the model to minimize the distance between positive pairs and maximize it between negatives. Specifically, as shown in Figure 2, the gene expression profiles of each cell, including the original view c_1 and corrupted view $\tilde{c}_1$, are embedded into latent representations $(\mathbf{e}_1, \tilde{\mathbf{e}}_1)$. The loss enforces that positive pairs (e.g., $\mathbf{e}_1$ and $\tilde{\mathbf{e}}_1$) are close in latent space, while negative pairs (e.g., $\mathbf{e}_1$ and $\tilde{\mathbf{e}}_n$) are far apart.

In this study, we employ the SimCLR [12] loss for contrastive pretraining, which is denoted as:

$$\mathcal{L}_{\text{contrast}} = -\frac{1}{2N} \sum_{i=1}^{N} \Bigg[\log \frac{\exp\left(\text{sim}(\mathbf{e}_i, \tilde{\mathbf{e}}_i)/\tau\right)}{\sum\limits_{j=1}^{N} \mathbb{I}_{[j \neq i]} \left(\exp\left(\text{sim}(\mathbf{e}_i, \tilde{\mathbf{e}}_j)/\tau\right) + \exp\left(\text{sim}(\mathbf{e}_i, \mathbf{e}_j)/\tau\right)\right)} + \log \frac{\exp\left(\text{sim}(\tilde{\mathbf{e}}_i, \mathbf{e}_i)/\tau\right)}{\sum\limits_{j=1}^{N} \mathbb{I}_{[j \neq i]} \left(\exp\left(\text{sim}(\tilde{\mathbf{e}}_i, \mathbf{e}_j)/\tau\right) + \exp\left(\text{sim}(\tilde{\mathbf{e}}_i, \tilde{\mathbf{e}}_j)/\tau\right)\right)} \Bigg] \tag{6}$$

where $\mathbf{e}^{(i)}$ and $\tilde{\mathbf{e}}^{(i)}$ represent the cell embedding of cell i from the original view and corrupted view respectively after being processed through the transformer block. $\mathbb{I}_{[k \neq i]} \in \{0, 1\}$ is an indicator function that takes the value of 1 if $k \neq i$; otherwise, it is 0. The similarity function $\text{sim}(\cdot, \cdot)$ represents cosine similarity here to measure the similarity between two cell embeddings, τ denotes the temperature parameter, and N is the number of cells in a batch.

The final tabular modeling objective in TABULA is a combination of the two loss functions:

$$\mathcal{L}_{\text{tab}} = \alpha \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{contrast}} \tag{7}$$

where the scaling factor α is used to balance the two loss terms to ensure comparable magnitudes. In this study, it is set to 0.03. The combined loss provides a holistic approach to learning both gene-level (column-wise) and cell-level (row-wise) features from the cell-by-gene matrix. While gene-wise reconstruction promotes robust gene-level representation learning, cell-wise contrastive learning enforces both intra-cell consistency and inter-cell variability. Together, these objectives enable TABULA to effectively model the tabular structure of single-cell data across both the gene and cell axes.

3.3 TABULA Pre-training & Fine-tuning

Pre-training. To protect data privacy in single-cell data, TABULA is pretrained using a federated learning framework shown in Figure 2. Each client, representing a hospital, research institute, or tissue type (e.g., lung, brain, heart), trains locally without sharing raw data. In this study, TABULA is pretrained on 15 million scRNA-seq profiles across eight clients: Intestine, Pancreas, Lung, Heart, Blood, Kidney, Brain, and Others (see Appendix A for pretraining data collection and preprocessing). Each client includes a tissue-specific embedder that encodes the cell-by-gene matrix into embeddings. These embeddings are passed through a tabular transformer to compute gene-wise and cell-wise self-supervised losses (shown in Equation 7). Training proceeds in alternating rounds of local and global updates: local transformer weights from client-side are uploaded to a central server, aggregated into a global model, and then broadcast back to clients (see Appendix B for the federated learning framework). This iterative communication enables knowledge sharing across heterogeneous datasets while preserving data confidentiality through strict decentralization. Note that this cross-tissue client setup demonstrates federated training of TABULA, the framework is generally applicable to cross-institutional or other settings, enabling collaborative pretraining without sharing raw data and thereby addressing privacy concerns. More pertaining details can be found in Appendix C.

Task-specific Fine-tuning. TABULA can be fine-tuned for a variety of downstream tasks, including gene perturbation prediction, cell type annotation, multi-omics integration, batch correction, and gene expression imputation shown in Figure 2. During fine-tuning, it first loads the tissue-specific pretrained embedder, followed by the pretrained global tabular transformer. A task-specific head or decoder is then attached based on the nature of the task, enabling the model to adapt effectively through task-specific optimization.

4 Experiments

4.1 Preliminary Study: Federated vs. Centralized and Tabular vs. MLM

To demonstrate the advantage of TABULA 's tabular learning and its adaptability to federated learning, we compare four pretraining strategies: centralized vs. federated learning, each paired with either tabular modeling or masked language modeling (MLM) (Figure 3 (left)). The pretraining dataset includes 1M cells from CELLxGENE (250K per tissue: pancreas, blood, brain, lung). Centralized training uses pooled data; federated training distributes data across four tissue clients with model weights aggregation. MLM uses a 15% masking ratio. We assess performance on two downstream tasks: cell type annotation (hPancreas dataset [13]) and genetic perturbation prediction (Adamson [14] and Norman [15] datasets), measuring accuracy and Pearson correlation coefficients. As shown in Figure 3, the results show that tabular modeling consistently outperforms MLM in both training schemes. Notably, federated-tabular pretraining achieved the highest overall performance, matching or exceeding centralized models despite data heterogeneity. More details of the implementation can be found in the Appendix C.

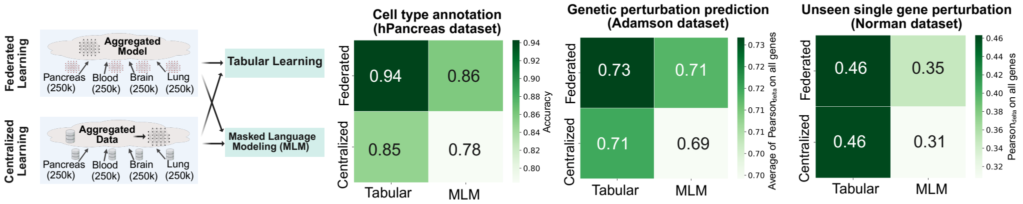


Figure 3: Benchmarking federated vs. centralized pretraining and tabular learning vs. masked language modeling (MLM) on 1 million cells.

4.2 Pretraining Data Scaling Laws in TABULA

To investigate how pretraining dataset size influences downstream performance, we identify a clear scaling law exhibited by TABULA. As depicted in Figure 4, increasing the number of pretraining cells from 300,000 to 3 million and then 15 million progressively enhances cell type annotation accuracy in the hPancreas dataset [13]. UMAP visualizations (Figure 4a) reveal more coherent clustering of annotated cell populations as data scale increases, especially notable in the alpha and beta cell clusters (highlighted in red). Quantitative analysis (Figure 4b) further confirms consistent improvements across accuracy, precision, recall, and macro F1 score, highlighting that larger-scale pretraining substantially strengthens the generalizability and robustness of learned representations. Importantly, these results also demonstrate that the benefits of scaling laws hold even under a federated learning framework, underscoring TABULA 's ability to leverage large-scale, decentralized datasets while preserving data privacy.

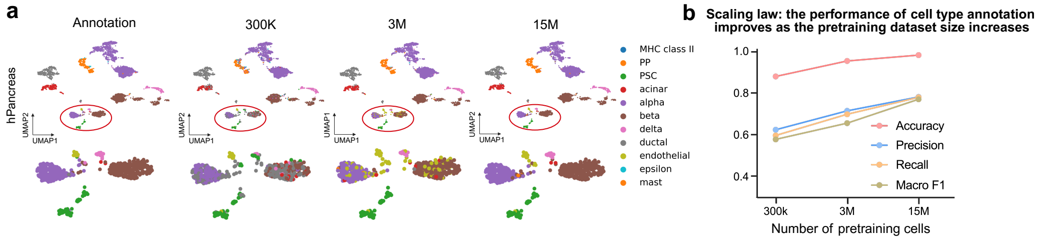
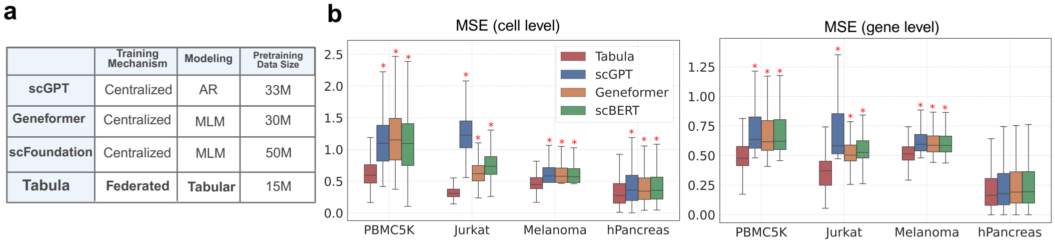


Figure 4: **Scaling law in TABULA. (a)** Pancreas-specific cell type annotation performance on the hPancreas dataset [13], visualized via UMAP. **(b)** Scaling law showing improved cell type annotation performance (accuracy, precision, recall, and macro F1) as the pretraining dataset increases from 300K to 3M and 15M cells.

4.3 Evaluation on Gene-Level Downstream Tasks

Task 1: Gene Imputation We first summarize key differences among TABULA and state-of-the-art foundation models from training paradigm, modeling objective, and pretraining data scale as shown in Figure 5a. Unlike existing centralized models that rely on language modeling objectives (AR or MLM), TABULA adopts a federated training mechanism and explicitly models the tabular structure of single-cell data with only half the pertaining data. We evaluate gene-level imputation performance by masking non-zero expression values in scRNA-seq data and predicting them using a fine-tuned decoder. TABULA is benchmarked against scGPT, scBERT, and Geneformer across four datasets (PBMC5K, Jurkat, Melanoma [16], hPancreas [13]). As shown in Figure 5b, TABULA consistently achieves lower mean squared error (MSE) at both the cell and gene levels, with statistically significant improvements (Wilcoxon $P < 0.05$, marked by *). See Appendix E for additional results and implementation details.



	Training Mechanism	Modeling	Pretraining Data Size
scGPT	Centralized	AR	33M
Geneformer	Centralized	MLM	30M
scFoundation	Centralized	MLM	50M
Tabula	Federated	Tabular	15M

Figure 5: TABULA improves imputation fidelity and accuracy. (a) Differences between TABULA and state-of-the-art foundation models. (b) MSE comparisons across four datasets at cell and gene levels; TABULA consistently achieves lower errors (Wilcoxon $P < 0.05$, marked by *).

Task 2: Genetic Perturbation Prediction We evaluate genetic perturbation prediction on three benchmark datasets: Adamson [14], Norman [15], and Replogle [17], covering both single- and double-gene perturbations. Performance is measured using Pearson$_{\text{delta}}$ between predicted and observed gene expression changes, on all genes and the top 20 differentially expressed (DE) genes per condition. We compare TABULA against both task-specific baselines, GEARS [18], CPA [19], and linear regression (LR), as well as foundation models including scGPT [4], Geneformer [1], and scFoundation [2]. As shown in Table 1, on Adamson, TABULA achieves the highest score on all genes (0.695) and performs comparably on DE genes (0.787 vs. 0.789 for scGPT). On Norman, it achieves the best DE score (0.778) and remains competitive overall. On Replogle, TABULA outperforms scGPT greatly by +0.069 (all genes) and +0.012 (DE genes). To assess generalization, we test on unseen Norman perturbations. TABULA outperforms scGPT in Pearson$_{\text{delta}}$ for both single- and double-gene settings, especially on DE genes (Figure 6 left).

We further assess reverse perturbation prediction following the setup of scGPT [4], where the task is to recover the causal gene perturbation(s) that led to an observed expression profile. We evaluate on a 20-gene benchmark comprising 210 possible single- and double-gene combinations, reporting top-K hit rates. As shown in Figure 6 (middle), TABULA achieves a 2/2 hit rate of 48.57% and a 1/2 hit rate of 94.29% at top-1, substantially outperforming scGPT. The heatmap in Figure 6 (right) visualizes the perturbation pairs correctly identified in the top-1 (red) and top-3 (blue), demonstrating TABULA 's effectiveness in pinpointing causal gene combinations. These results highlight TABULA 's strong performance in both predictive and reverse perturbation tasks, especially under distribution shifts and combinatorial settings. See Appendix E for additional results and implementation details.

Table 1: *(Task 2)* The results of genetic perturbation prediction across Adamson, Norman, and Replogle datasets, evaluated by Pearson$_{\text{delta}}$ on all genes and differentially expressed (DE) genes. OOT indicates out-of-time.

Model	Adamson		Norman		Replogle	
	All Genes	DE Genes	All Genes	DE Genes	All Genes	DE Genes
GEARS	0.531	0.678	0.547	0.715	0.154	0.277
CPA	0.145	0.309	–	–	–	–
LR	0.387	0.620	0.532	0.697	0.141	0.298
Geneformer	0.199	0.666	0.316	0.326	0.091	0.362
scFoundation	OOT	OOT	OOT	OOT	OOT	OOT
scGPT	0.615	**0.789**	**0.583**	0.742	0.242	0.464
TABULA	**0.695**	0.787	0.577	**0.778**	**0.311**	**0.476**

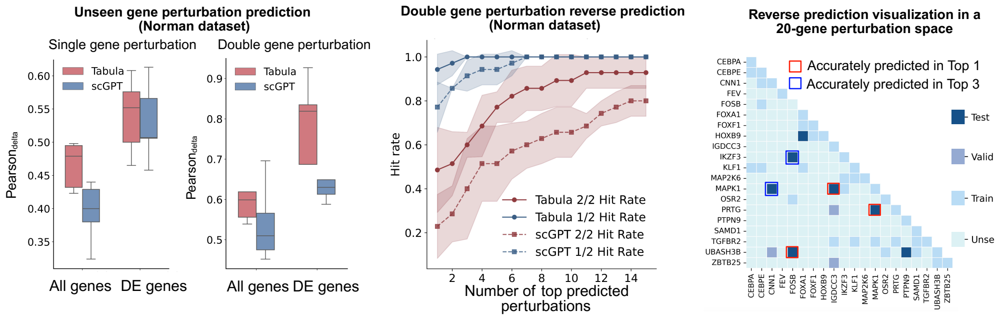


Figure 6: TABULA outperforms scGPT on unseen gene perturbation and reverse perturbation. Left: Pearson correlations for unseen single and double perturbation prediction on all and DE genes. Middle: Top-K hit rates for reverse prediction of double perturbations (1/2 and 2/2 correct). Right: Heatmap of 20-gene pairwise space with top-1 (red) and top-3 (blue) accurate reverse predictions.

4.4 Evaluation on Cell-Level Downstream Tasks

Task 3: Cell Type Annotation Cell type annotation assigns biological labels to cells based on gene expression profiles. To evaluate TABULA, we fine-tune it with a supervised cross-entropy loss and test on three datasets: hPancreas [13], Myeloid [4], and Cell Lines [20]. We compare against scGPT, Geneformer, and scFoundation, each fine-tuned with their default settings. Performance is measured using accuracy, precision, recall, and F1 score. TABULA achieves the best overall performance on hPancreas and Cell Lines, and while slightly behind scFoundation in Myeloid accuracy, it outperforms others in remaining metrics—demonstrating strong generalization (Table 2). Notably, TABULA achieves this using only half the pretraining data while ensuring privacy protection through federated training. See Appendix E for additional results and implementation details. We also report **Task 4&5 Multi-Omics and Multi-Batch Integration** results with implementation details in Appendix E.

Table 2: *(Task 3)* The results of cell type annotation across hPancreas, Myeloid, and Cell Lines datasets, evaluation by test accuracy, precision, recall, and F1 score.

Model	hPancreas				Myeloid				Cell Lines			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
geneFormer	0.9395	0.6534	0.6491	0.6425	0.5953	0.3704	0.3300	0.3441	0.9911	0.9918	0.9904	0.9910
scFoundation	0.9492	0.6140	0.6032	0.6003	0.6352	**0.4070**	**0.3764**	**0.3804**	0.9931	0.9932	0.9930	0.9931
scGPT	0.9680	0.7350	0.7250	0.7180	**0.6420**	0.3660	0.3470	0.3460	0.9930	0.9930	0.9920	0.9930
TABULA	**0.9810**	**0.7814**	**0.7795**	**0.7708**	0.6213	0.3909	0.3634	0.3728	**0.9935**	**0.9939**	**0.9931**	**0.9935**

In addition to these main results, we provide additional analyses in the appendix to further demonstrate the efficacy of TABULA. Specifically, we demonstrate that tissue-specific embedders can effectively capture biologically meaningful variation across different tissue types, as detailed in Appendix F. Furthermore, ablation studies presented in Appendix G confirm the importance and effectiveness of our proposed tabular modeling objectives, highlighting their critical role in enhancing downstream performance.

5 Related Work

Recent advances in single-cell foundation models (FMs) have adapted techniques from natural language processing to learn transferable representations from large-scale transcriptomic data. Notable models include Geneformer [1], scFoundation [2], scGPT [4], and CellPLM [3]. These models predominantly employ centralized training and language modeling objectives such as autoregressive (AR) [21] or masked language modeling (MLM) [22], which ignore the intrinsic tabular structure of single-cell data. Moreover, those models reply on centralized training thus raising concerns about data privacy and ethics from public single-cell data. Prior work has shown that scRNA-seq data is vulnerable to linkage attacks [6, 7].

To address these limitations, we propose TABULA, a foundation model that captures the tabular structure of single-cell data through tabular modeling while preserving privacy through federated learning. TABULA employs a dual-axis pretraining objective, gene-wise reconstruction and cell-wise contrastive learning, to model the cell-by-gene matrix without imposing artificial gene order. By decoupling global updates from client-specific embedders, it supports privacy-preserving learning across institutions. Extensive experiments demonstrate the effectiveness of TABULA.

6 Conclusion

In this work, we present TABULA, the first privacy-preserving foundation model for single-cell transcriptomics that explicitly models the cell-by-gene matrix using a novel tabular pretraining objective. To our knowledge, TABULA is the first to propose a tabular modeling framework tailored for single-cell data, moving beyond sequence-based approaches that impose artificial gene sequence order. It achieves strong performance across diverse benchmark tasks using only half the pretraining data and captures complex biological signals across gene- and cell-level representations. It is important to note that as public single-cell datasets continue to grow, TABULA offers a scalable and privacy-aware foundation that not only validates the feasibility of federated tabular modeling, but also establishes a generalizable framework for training future models under similar privacy-preserving settings.

References

[1] 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*, 618(7965):616–624, 2023.

[2] Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature methods*, 21(8):1481–1491, 2024.

[3] Hongzhi Wen, Wenzhuo Tang, Xinnan Dai, Jiayuan Ding, Wei Jin, Yuying Xie, and Jiliang Tang. CellPLM: Pre-training of cell language model beyond single cells. In *The Twelfth International Conference on Learning Representations*, 2024.

[4] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21(8):1470–1480, 2024.

[5] Anna C Schaar, Alejandro Tejada-Lapuerta, Giovanni Palla, Robert Gutgesell, Lennard Halle, Mariia Minaeva, Larsen Vornholz, Leander Dony, Francesca Drummer, Mojtaba Bahrami, et al. Nicheformer: a foundation model for single-cell and spatial omics. *bioRxiv*, pages 2024–04, 2024.

[6] Conor R Walker, Xiaoting Li, Manav Chakravarthy, William Lounsbery-Scaife, Yoolim A Choi, Ritambhara Singh, and Gamze Gürsoy. Private information leakage from single-cell count matrices. *Cell*, 187(23):6537–6549, 2024.

[7] D'Nae Ferguson. A privacy concern: Bioinformatics and storing biodata. In *The ADMI 2021 Symposium*, 2021.

[8] 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.

[9] Bingzhao Zhu, Xingjian Shi, Nick Erickson, Mu Li, George Karypis, and Mahsa Shoaran. Xtab: Cross-table pretraining for tabular transformers. *arXiv preprint arXiv:2305.06090*, 2023.

[10] Vinit Kumar Mishra, Karthik Natarajan, Dhanesh Padmanabhan, Chung-Piaw Teo, and Xiaobo Li. On theoretical and empirical aspects of marginal distribution choice models. *Management Science*, 60(6):1511–1531, 2014.

[11] Ruochen Jiang, Tianyi Sun, Dongyuan Song, and Jingyi Jessica Li. Statistics or biology: the zero-inflation controversy about scrna-seq data. *Genome biology*, 23(1):31, 2022.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[13] 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.

[14] Britt Adamson, Thomas M Norman, Marco Jost, Min Y Cho, James K Nuñez, Yuwen Chen, Jacqueline E Villalta, Luke A Gilbert, Max A Horlbeck, Marco Y Hein, et al. A multiplexed single-cell crispr screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7):1867–1882, 2016.

[15] 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.

[16] Eduardo Torre, Hannah Dueck, Sydney Shaffer, Janko Gospocic, Rohit Gupte, Roberto Bonasio, Junhyong Kim, John Murray, and Arjun Raj. Rare cell detection by single-cell rna sequencing as guided by single-molecule rna fish. *Cell systems*, 6(2):171–179, 2018.

[17] 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.

[18] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, 42(6):927–935, 2024.

[19] 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.

[20] 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*, 21:1–32, 2020.

[21] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[24] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[26] Harm-Jan Westra and Lude Franke. From genome to function by studying eqtls. *Biochimica et Biophysica Acta (BBA)-molecular basis of Disease*, 1842(10):1896–1902, 2014.

[27] Cédric Arisdakessian, Olivier Poirion, Breck Yunits, Xun Zhu, and Lana X Garmire. Deepimpute: an accurate, fast, and scalable deep neural network method to impute single-cell rna-seq data. *Genome biology*, 20:1–14, 2019.

[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] Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360, 2016.

[30] Sonya A MacParland, Jeff C Liu, Xue-Zhong Ma, Brendan T Innes, Agata M Bartczak, Blair K Gage, Justin Manuel, Nicholas Khuu, Juan Echeverri, Ivan Linares, et al. Single cell rna sequencing of human liver reveals distinct intrahepatic macrophage populations. *Nature communications*, 9(1):4383, 2018.

[31] Nadim Aizarani, Antonio Saviano, null Sagar, Laurent Mailly, Sarah Durand, Josip S Herman, Patrick Pessaux, Thomas F Baumert, and Dominic Grün. A human liver cell atlas reveals heterogeneity and epithelial progenitors. *Nature*, 572(7768):199–204, 2019.

[32] 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.

[33] Malte D Luecken, Daniel Bernard Burkhardt, Robrecht Cannoodt, Christopher Lance, Aditi Agrawal, Hananeh Aliee, Ann T Chen, Louise Deconinck, Angela M Detweiler, Alejandro A Granados, et al. A sandbox for prediction and integration of dna, rna, and proteins in single cells. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (Round 2)*, 2021.

[34] 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.

[35] Alexandra-Chloé Villani, Rahul Satija, Gary Reynolds, Siranush Sarkizova, Karthik Shekhar, James Fletcher, Morgane Griesbeck, Andrew Butler, Shiwei Zheng, Suzan Lazo, et al. Single-cell rna-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science*, 356(6335):eaah4573, 2017.

[36] Xiaojie Qiu, Yan Zhang, Jorge D Martin-Rufino, Chen Weng, Shayan Hosseinzadeh, Dian Yang, Angela N Pogson, Marco Y Hein, Kyung Hoi Joseph Min, Li Wang, et al. Mapping transcriptomic vector fields of single cells. *Cell*, 185(4):690–711, 2022.

[37] Jan Krumsiek, Carsten Marr, Timm Schroeder, and Fabian J Theis. Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. *PloS one*, 6(8):e22649, 2011.

[38] T Yvanka de Soysa, Sanjeev S Ranade, Satoshi Okawa, Srikanth Ravichandran, Yu Huang, Hazel T Salunga, Amelia Schricker, Antonio Del Sol, Casey A Gifford, and Deepak Srivastava. Single-cell analysis of cardiogenesis reveals basis for organ-level developmental defects. *Nature*, 572(7767):120–124, 2019.

[39] Gioele La Manno, Kimberly Siletti, Alessandro Furlan, Daniel Gyllborg, Elin Vinsland, Alejandro Mossi Albiach, Christoffer Mattsson Langseth, Irina Khven, Alex R Lederer, Lisa M Dratva, et al. Molecular architecture of the developing mouse brain. *Nature*, 596(7870):92–96, 2021.

[40] Junqiang Wang, Ruoshi Yuan, Xiaomei Zhu, and Ping Ao. Adaptive landscape shaped by core endogenous network coordinates complex early progenitor fate commitments in embryonic pancreas. *Scientific reports*, 10(1):1112, 2020.

[41] Sophia K Longo, Margaret G Guo, Andrew L Ji, and Paul A Khavari. Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. *Nature Reviews Genetics*, 22(10):627–644, 2021.

[42] Jun Ding, Nadav Sharon, and Ziv Bar-Joseph. Temporal modelling using single-cell transcriptomics. *Nature Reviews Genetics*, 23(6):355–368, 2022.

[43] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences*, 9(5):909, 2019.

[44] Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon. A survey on universal adversarial attack. *arXiv preprint arXiv:2103.01498*, 2021.

[45] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760*, 2020.

[46] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions of the paper, introducing TABULA, a federated foundation model for single-cell data that incorporates tabular modeling and achieves strong performance across benchmark tasks.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are acknowledged in the Appendix I, Limitations in TABULA.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not contain formal theoretical results or proofs. It is an empirical paper focused on model development and biological evaluation.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Detailed descriptions of data, model architecture, and training setup are provided in Sections 3 and 4, with additional implementation details and evaluation protocols in Appendix A, C, and E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper will provide open-source code along with an Appendix. More data and implementation details can be found in Appendix A, C, and E.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: They are provided in Appendix A, C, and E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Statistical significance is reported in Figure 4b with Wilcoxon P-values < 0.05 marked by * for gene imputation. This demonstrates proper statistical testing for core performance claims.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C includes details on computer resources for TABULA pretraining.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are addressed throughout the introduction and discussion. TABULA reduces privacy risks in genomics, enabling collaborative model training across institutions without data sharing. Separate section talking about broader impact of TABULA in Appendix L.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Although the model is not released yet, the design of TABULA includes safeguards such as federated learning and local embedders to mitigate misuse and protect sensitive genomic data.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external datasets and models (e.g., scGPT, CELLxGENE) are properly credited in the references and their terms of use respected. The paper relies on public datasets available under research licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new foundation model TABULA for single-cell has been trained and released along with its public training dataset after our in house processing.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The study does not involve human subject research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM just used for writing grammar checking.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.