MANCHESTER 1824
The University of Manchester

Unsupervised representation learning for spatial transcriptomics

A dissertation submitted to The University of Manchester for the degree of

Master of Science in Advanced Computer Science

in the Faculty of Science and Engineering

Year of submission

2025

Student ID

u04876ji

School of Engineering

Contents

Word count: 8449

Abstract

Single-cell RNA sequencing (scRNA-seq) and spatial transcriptomics (ST) provide powerful insights into cellular heterogeneity and tissue organization but demand robust computational frameworks. This thesis investigates Transformer-based foundation models for single-cell analysis, focusing on CellPLM. We revisit its theoretical underpinnings, including the cell language model, Flowformer attention, and Gaussian mixture priors, and extend its implementation to downstream tasks.

Through extensive experiments, we show that CellPLM produces biologically meaningful embeddings and surpasses PCA in clustering, even in zero-shot settings. Fine-tuning further enhances performance, with improved Macro F_1 scores for cell type annotation. A key contribution is the introduction of a supervised contrastive (SupConLoss) head, which achieves clustering accuracy comparable to cross-entropy loss while reducing training time by up to two orders of magnitude, enabling scalable analysis.

Despite these advances, ST clustering remains weaker, imputation correlations modest, and gene–gene interactions underexplored, highlighting the need for larger pretraining corpora, hybrid architectures, and improved fine-tuning strategies.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

i The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made *only* in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.library.manchester.ac.uk/about/regulations/`) and in The University's policy on Presentation of Theses.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisors, Prof. Hongpeng Zhou and Haiping Liu for their continuous support, invaluable guidance, and encouragement throughout my master's study. I am also grateful to my friends for their helpful discussions and assistance. Finally, I owe my deepest appreciation to my family for their unwavering love and support.

1 Introduction

1.1 Background and motivation

Single-cell RNA sequencing (scRNA-seq) technology has provided groundbreaking insights to the biological research community, offering a detailed view of cell types, cell states, and their dynamic changes during development, disease onset, and therapeutic response [1]. Since the emergence of this sequencing technology, vast amounts of data have been generated, driving strong interest in large-scale single-cell analytical methods.

Machine learning approaches have taken a leading role in analyzing scRNA-seq data. However, differences in model architectures, input data formats, and intended applications of early machine learning models have hindered effective knowledge transfer across tasks. To address this issue, the concept of a **foundation model** has emerged [2].

Foundation models are pretrained on large unlabeled corpora with unsupervised or self-supervised objectives, thereby learning universal representations that encode broad domain statistics. Consequently, many downstream tasks require only shallow task-specific heads and minimal additional training [3]. They also support few-shot fine-tuning and, in some cases, zero-shot inference [3]. Their transferability and modularity allow the same backbone to support diverse tasks through interchangeable heads [4]. Furthermore, scaling laws indicate predictable performance improvements as data and computational resources increase (up to a compute-optimal frontier) [5]. Exposure to diverse data distributions also enhances robustness against noise and distributional shifts (e.g., batch effects), though this benefit can vary depending on the task and dataset [5].

In parallel, a major turning point in the field of machine learning has been the introduction of the Transformer architecture. Originally developed for natural language processing (NLP) tasks such as large language models (e.g., ChatGPT-4.0) [6], Transformers have since been successfully applied in most foundation models across various domains [5]. Examples include the Vision Transformer (ViT) in computer vision [7], Whisper in speech processing [8], Informer for time-series forecasting [9], BioBERT for genomics [10], and AlphaFold for protein sequence modeling [11]—all of which demonstrate the broad adaptability of this architecture.

Building on these advances, a growing body of research has explored the development of Transformer-based foundation models specifically for scRNA-seq data analysis. These models conceptualize genes as words (tokens) and cells as sentences for Transformer training. For instance, as summarized in Table 1, **scBERT** [12] introduced a Transformer pretrained on large-scale scRNA-seq datasets, where continuous gene expression values are discretized into categorical tokens, thus

enabling application of the BERT architecture to single-cell data. **Geneformer** [13] extended this approach with a masking strategy designed to predict gene rankings within a cell. More recently, **scGPT** [14] adapted the GPT family of generative models to single-cell omics by segmenting gene expression values into discrete intervals and auto-regressively predicting subsequent expression patterns using attention mechanisms. These models learns contextualized gene embeddings and achieves strong predictive performance across biological tasks.

Despite these successes, applying Transformers to single-cell transcriptomics presents unique challenges, stemming from fundamental differences between single-cell data and natural language data. First, scRNA-seq data are inherently non-sequential. Rather than forming natural sequences, they are structured as a cell-by-gene count matrix that captures the abundance of individual genes within each cell [2]. Unlike token sequences in text, this representation complicates efforts to model gene–cell relationships with Transformers.

Second, relationships between cells are biologically more meaningful than relationships between tokens in a sentence. Cell–cell communication plays a critical role in shaping cellular states and developmental trajectories. Moreover, within tissues, many cells share common or related lineages, resulting in correlated gene expression profiles. These correlations provide valuable information for denoising and accurately identifying cell states [2].

Third, single-cell data are generally noisier and of lower quality compared to text corpora. For example, while the English dataset extracted from Common Crawl comprises roughly 32 billion high-quality sentences, the Human Cell Atlas—the largest collection of single-cell data—contains fewer than 50 million cells [2]. Furthermore, scRNA-seq data are often confounded by technical artifacts, dropout events, and pronounced batch effects across sequencing platforms and experiments [2].

To address these limitations, graph-based approaches have emerged as a promising direction. Several methods construct cell–cell graphs to improve representation learning in single-cell analysis. For example, **scGNN** [15] learns cell representations with a graph autoencoder built from scRNA-seq expression data, simultaneously performing dropout correction and clustering. **SpaGCN** [16] fuses spatial coordinates, expression profiles, and histological data in a graph and applies a Graph convolutional network(GCN) to detect spatial domains and spatially variable genes. **DeepST** [17] employs Graph neural network(GNN) autoencoder-based embeddings that respect spatial neighborhoods, thereby improving tissue domain segmentation and batch integration while coupling imputation with representation learning.

Nonetheless, while these GNN-based models provide important advances, generalizable graph foundation models remain difficult to establish. Two primary challenges persist. First, heterogene-

ity: datasets vary in node features, edge semantics, and graph-construction strategies, making it difficult for a single pretrained backbone to transfer across datasets [18]. Second, scalability: computational cost grows with graph density, and deep message passing tends to oversmooth features, hindering the capture of long-range dependencies [18].

These limitations highlight a central problem: how to design a foundation model for scRNA-seq that can capture cell–cell relationships, impose meaningful sequence structures, and overcome the constraints of data quality and quantity. Fortunately, recent advances in spatial transcriptomics (ST) technology provide a promising solution. ST captures spatial coordinates of cells along with their gene expression profiles in tissue slices. This spatial information not only offers a natural sequence of cells but also enables foundation models to learn biologically meaningful intercellular relationships.

Building on this insight, we introduce the single-**Cell P**re-trained **L**anguage **M**odel (CellPLM) [2], which explicitly incorporates spatial context into transcriptomic pretraining. First, it employs a cell language model to capture cell–cell relationships, initializing cell embeddings by aggregating gene embeddings to reflect the bag-of-words nature of gene expression features. Second, it incorporates ST data, which provide positional information that complements scRNA-seq data and enhances modeling of cell–cell interactions. Third, to alleviate the limitations of single-cell data quantity and quality, CellPLM introduces an inductive bias by applying a Gaussian mixture model as the prior distribution in the latent space, thereby generating smoother and more informative cell representations.

In addition, we introduce the latest ST foundation model, **SToFM** [19]. SToFM is a multi-scale foundation model designed to integrate information across three levels within a tissue slice: macro-scale tissue morphology, micro-scale local cellular environments, and gene-scale expression patterns. Each tissue slice is partitioned into subregions of varying sizes, and an SE(2) Transformer is employed to merge features across scales, producing high-quality cell embeddings. Powered by its multi-scale attention mechanism, SToFM achieves state-of-the-art performance in downstream tasks such as tissue domain segmentation and cell type annotation, and we compare its experimental results with our CellPLM-based results.

1.2 Project Achievements

CellPLM is the first single-cell foundation model to incorporate spatial context into transcriptomic pretraining by integrating gene expression embeddings with positional embeddings as Transformer inputs. To provide a deeper theoretical understanding of this framework, we revisit the principles

Table 1. Condensed summary of foundation models for scRNA-seq and spatial transcriptomics.

Model	Omic modalities	Pretraining scale	Key idea / Input embedding	Applications & Zero-shot
scVI (2018)[20]	scRNA-seq	$\sim 10^5$ cells	Variational autoencoder on raw counts (NB latent variables)	Batch correction, clustering, DE; limited imputation
scGNN (2021)	scRNA-seq	10k–30k cells	Graph neural network with iterative autoencoders; dropout modeling using LTMG	Gene imputation, clustering, trajectory inference; Alzheimer's disease analysis
SpaGCN (2021)	ST + histology	N/A	Graph CNN integrating expression + spatial coords + H&E	Spatial domain detection, SVGs; no zero-shot
DeepST (2022)	ST	N/A	Augmented gene expression + morphology + spatial coordinates; GNN-based autoencoders	Spatial domain detection, batch correction, cancer heterogeneity
scBERT (2022)	scRNA-seq	$\sim 10^6$ cells	Transformer encoder with gene ID + expression bins	Cell type annotation; limited zero-shot
Geneformer (2023)	scRNA-seq	30M+ cells	Rank-based gene ordering + MLM pretraining	Network biology, function prediction; clustering, GRN inference, perturbation
scGPT (2024)	scRNA-seq, CITE-seq, Perturb-seq	33M+ cells	GPT-style decoder with value binning (gene+expr.)	Annotation, perturbation, integration; clustering, simulation, imputation
CellPLM (2024)	**scRNA-seq, ST**	**11M+ cells**	**Cells as tokens; expression projection into PLM-style encoder**	**Annotation, imputation, perturbation; clustering, denoising**
SToFM (2025)	ST	88M cells	Multi-scale (gene/micro/-macro): domain-adapted cell encoder (Geneformer) + SE(2) Transformer; MCM & PDR self-supervision	Tissue region segmentation, cell type annotation, deconvolution, imputation; zero-shot clustering/visualization

of the Transformer architecture and derive the Gaussian mixture variational autoencoder from first principles, as detailed in Section 2.

While the original CellPLM paper demonstrated strong performance, its evaluation was limited to a small number of datasets and the released codebase provided only partial tutorials for downstream tasks. In this thesis, we address these limitations in two main ways.

First, we extend the empirical scope of evaluation by applying CellPLM to a wide range of scRNA-seq datasets, including breast cancer, colorectal cancer, lung, and liver, as well as ST datasets such as DLPFC Visium and MERFISH mouse brain2. Across these datasets, we demonstrate that CellPLM consistently outperforms PCA in clustering, achieves stronger Macro F_1 scores for minority cell populations in annotation, and exhibits context-dependent benefits in ST imputation.

Second, we extend the publicly available implementation to support fine-tuned cell embedding clustering and zero-shot cell type annotation, both of which were absent in the original repository. Building on this, we develop a novel fine-tuning pipeline for clustering that systematically compares self-supervised, supervised contrastive, and fully supervised learning. Notably, our introduction of a SupConLoss head yields clustering performance competitive with cross-entropy loss while reducing training time by up to two orders of magnitude, establishing a substantially more scalable and

practical alternative.

Summary of Contribution: Clustering Fine-tuning Pipeline

- **Codebase Extension.** The original CellPLM repository provided only limited tutorials (zero-shot clustering and fine-tuned cell type annotation). We extended the implementation to support:
  1. Fine-tuned cell embedding clustering
  2. Zero-shot inference for cell type annotation
- **Novel Clustering Pipeline.** We designed a new fine-tuning pipeline for clustering, systematically comparing three strategies:
  - *Self-supervised learning*:
    1. Decoder-based reconstruction loss — aligned with the original CellPLM pretraining objective, but yielded poor clustering quality (high reconstruction accuracy $\not\Rightarrow$ meaningful embeddings).
    2. KL-only latent loss — resulted in posterior collapse, with embeddings degenerating to the prior distribution.
  - *Supervised learning*: Standard cross-entropy loss achieved the strongest absolute clustering performance (e.g., ARI > 0.9 in Mouse Brain2) but required full label supervision and was computationally very expensive.
  - *Supervised contrastive learning (SupConLoss)*: Our introduced head achieved clustering performance competitive with cross-entropy loss while reducing training time by **30–300**$\times$, offering substantially greater scalability.
- **Key Finding.** SupConLoss provides a favorable trade-off between clustering performance and computational efficiency, making it a practical solution for large-scale or weakly labeled single-cell datasets.

Accordingly, our main contributions are summarized as follows:

1) We investigate the architectural and mathematical foundations of the CellPLM model.

2) We apply fine-tuning on various scRNA-seq and ST datasets for downstream tasks such as cell type classification, clustering, and spatial transcriptomics imputation, and compare the results against state-of-the-art benchmarks.

3) We develop a fine-tuning pipeline for cell embeddings, systematically comparing zero-shot infer-

ence, weakly supervised learning, and fully supervised learning strategies.

1.3 Report Structure

The remainder of this thesis is organized as follows:

Chapter 2 provides a detailed description of the proposed model architecture. **Chapter 3** presents the experiments conducted on various scRNA-seq and ST datasets, along with their results. **Chapter 4** concludes with a discussion of findings and potential directions for future research.

2 Methods

For this project, we adopt CellPLM's codebase as the baseline and utilize its pre-trained parameters as the backbone for fine-tuning. It is therefore essential to briefly review the model architecture and core principles of CellPLM. As illustrated in Fig. 3 [2], CellPLM consists of four main components: a **gene expression embedder, a transformer encoder, a latent space, and a decoder.** Notably, CellPLM employs Flowformer, a variant of the canonical Transformer designed to handle long sequences such as scRNA-seq data. In addition, its latent representation is modeled using a Gaussian Mixture Variational Autoencoder (GMVAE), which is more sophisticated than standard VAEs and thus requires closer examination. Finally, because pre-training for capturing cell-to-cell relationships and fine-tuning for downstream biological tasks follow substantially different frameworks, we describe them separately in the following sections.

2.1 Cell Language Model

Conventional single-cell Transformer models typically operate on the *cell-by-gene expression matrix*, where a cell is treated analogously to a sentence in natural language and each gene corresponds to a token [2]. In this setting, $X_{i,j}$ denotes the observed expression of gene j in cell i, with $i \in \{1, \ldots, N\}$ indexing cells and $j \in \{1, \ldots, K\}$ indexing genes. Let $O(i)$ be the set of observed genes in cell i and $U(i)$ the set of unobserved genes, which gives the conventional gene language model:

$$p\left(X_{i,j} \mid \{X_{i,o}\}_{o \in \mathcal{O}(i)}\right),\ j \in \mathcal{U}(i), \tag{1}$$

The original CellPLM framework [2] extends this idea by incorporating *cell-to-cell dependencies*, which are biologically crucial. Specifically, if M denotes the set of masked entries in X and M^c the unmasked entries, then the conditional distribution of a masked value $X_{i,j}$ depends not only on

genes from the same cell but also on other cells:

$$p\left(X_{i,j} \mid \{X_{u,v}\}_{(u,v)\in\mathcal{M}^c}\right),\ (i,j) \in \mathcal{M}, \tag{2}$$

This "cell language model" generalizes the conventional gene-centric view by capturing latent distributions over entire cells rather than individual genes.

2.2 Model Architecture

2.2.1 Gene expression Embedder

The first module in CellPLM is the *gene expression embedder*. Let h_j denote the embedding vector for gene j with dimensionality equal to the hidden size of the Transformer encoder layers. h_j is initialized randomly during pretraining and reused during downstream tasks. The i-th row of the embedding matrix E, E_i, is computed as:

$$E_i = \sum_j X_{i,j} \cdot h_j \tag{3}$$

2.2.2 Transformer Encoder

Transformer CellPLM adopts an encoder–decoder architecture, where the encoder is based on the standard Transformer attention mechanism. In a Transformer, the most important component is the *attention* mechanism. Attention operates across different tokens in the input sequence, assigning higher weights to tokens that are more relevant to the token currently being processed. Vaswani et al. [21] introduced the scaled dot-product method to compute attention. The *scaled dot-product attention* is defined as below and the structure is shown in Figure 1:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{4}$$

where Q, K, and V denote the query, key, and value matrices, respectively. The *query* represents the information being sought for the current token, the *key* represents the identifiers of each token, and the *value* represents the actual information associated with each token.

The similarity score is computed by taking the dot product of Q and K. To prevent the softmax function from producing extremely small gradients, the score is scaled by $\sqrt{d_k}$. The softmax then converts these similarity scores into probabilities that sum to 1. Finally, multiplying the probabilities with V yields the attention output as a weighted sum over the token representations.

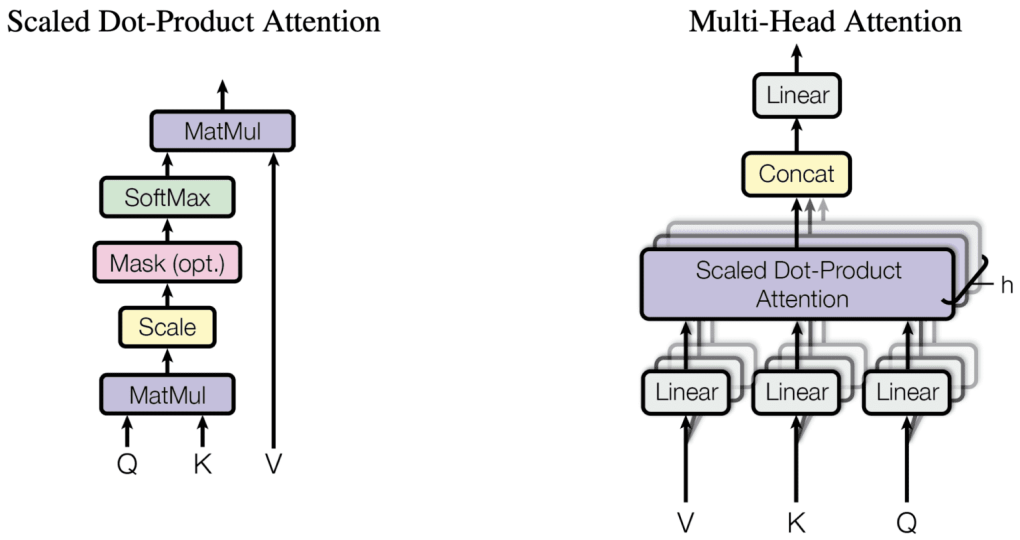

Fig. 1. (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel. [21]

However, using a single attention function may cause the model to focus only on a single type of relationship, potentially losing other important information. To mitigate this, the Transformer introduces *multi-head attention*. Specifically, h different sets of Q, K, and V transformations are created, and attention is computed in parallel for each set. Each head can learn to capture different types of relationships between tokens. The outputs from all heads are then concatenated and linearly transformed.

Positional Encoding For spatial transcriptomics (ST) datasets, the CellPLM embedder layer allows us to choose whether to apply positional encoding (PE) or not. When applied, PE incorporates two-dimensional spatial coordinates (x, y) using a sinusoidal encoding scheme.

Compared with the canonical Transformer PE defined in Eq. (5) and Eq. (6), the CellPLM PE is reformulated as shown in Eq. (7) and Eq. (8). The reason is that the canonical Transformer is designed for textual data, which only requires one-dimensional positional information, whereas ST data contains two-dimensional coordinates for each cell, thus necessitating a two-dimensional PE.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right), \tag{5}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \tag{6}$$

$$PE_{(x,y,2i)} = \sin\left(\frac{x}{10000^{4i/d}}\right), \quad PE_{(x,y,2i+1)} = \cos\left(\frac{x}{10000^{4i/d}}\right), \tag{7}$$

$$PE_{(x,y,2j+d/2)} = \sin\left(\frac{y}{10000^{4j/d}}\right), \quad PE_{(x,y,2j+1+d/2)} = \cos\left(\frac{y}{10000^{4j/d}}\right), \tag{8}$$

where d denotes the embedding dimension. The factor $4i/d$ (and $4j/d$ for y) ensures that the em-

bedding dimension is evenly divided between the x and y coordinates. In other words, half of the embedding dimensions are assigned to encode x and the other half to encode y, so that both spatial axes are represented with equal capacity. This two-dimensional encoding can either be summed with or concatenated to the expression embeddings.

Flowformer

What CellPLM actually employs for both pre-training and downstream tasks is Flowformer [22]. The softmax function used in the Transformer has a "winner-takes-all" nature, which induces competition among tokens. Specifically, if one similarity score (logit) is slightly larger, the exponential operation amplifies the difference, leading the token with the slightly higher score to receive a disproportionately large weight. Another reason for this competition lies in the normalization property of softmax, which enforces that the outputs sum to 1, as shown below:

$$\text{Softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{n} \exp(z_j)}.$$

Thus, when the attention score of one token increases, the weights assigned to the others must decrease, resembling a zero-sum game.

While this competitive mechanism allows important tokens to receive higher weights, the traditional Transformer computes similarity as $QK^\top$, comparing all token pairs, which results in quadratic complexity $O(n^2 d)$. Moreover, as the sequence length increases, the dependency between distant tokens tends to weaken. To reduce this complexity and make attention efficient even for long sequences, many approaches have been proposed. A representative example is linearization, which avoids computing softmax directly and instead decomposes similarity through kernel methods or matrix factorization, reducing the complexity to $O(n)$. However, this approach suffers from the problem of *trivial attention*, where irrelevant tokens still receive non-negligible attention weights. To address this, some studies introduced inductive biases, such as enforcing stronger connections between neighboring tokens, but these methods lack generality.

Flowformer provides a new perspective by interpreting attention as a flow network. The mechanism is illustrated in Fig.2. The sink node R corresponds to the result of canonical Transformer attention $\text{Attention}(Q, K, V)$, receiving incoming flows. The source node corresponds to V, which sends outgoing flows toward the sink R. Each edge is associated with a capacity, which represents the attention weight. By constraining the total incoming flow to the sink, competition naturally emerges among sources. Similarly, by constraining the total outgoing flow from each source, the model ensures that information is distributed across sinks. This principle is known as *flow conservation*. Formally, if f_{ij} denotes the flow from source i to sink j, then Flowformer enforces the following con-

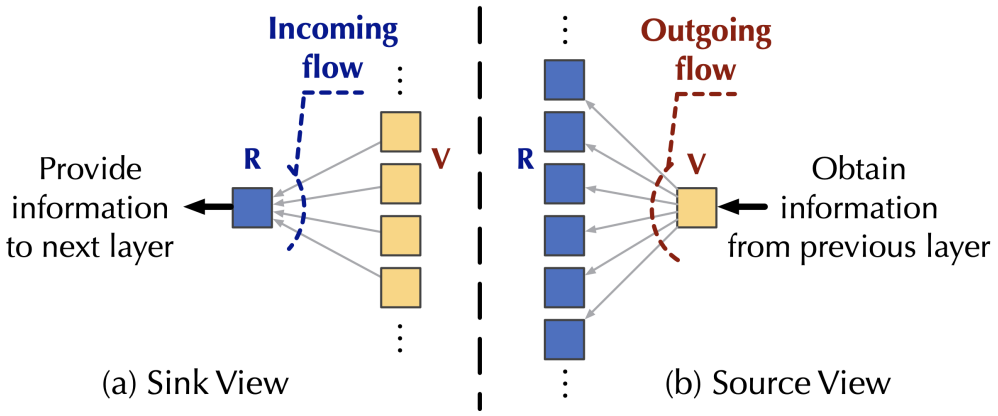

Fig. 2. The flow network for Attention.[22]

straints:

$$\sum_{i=1}^{n} f_{ij} = C_j, \quad \forall j \quad \text{(incoming flow conservation)}, \tag{9}$$

$$\sum_{j=1}^{m} f_{ij} = D_i, \quad \forall i \quad \text{(outgoing flow conservation)}, \tag{10}$$

where C_j is the fixed total incoming flow for sink j, and D_i is the fixed total outgoing flow for source i.

Through this flow-based interpretation, Flowformer achieves competition and allocation without explicitly relying on the softmax function, thereby avoiding trivial attention. Its computational complexity is $O(nd^2)$, which makes it particularly advantageous when the number of tokens n is much larger than the embedding dimension d, as is often the case in long sequences. Consequently, Flowformer is more suitable than the standard Transformer for learning from scRNA-seq and ST datasets, which typically contain sequences much longer than those in standard text corpora.

2.2.3 Gaussian Mixture Latent Space

While conventional VAEs have been widely applied in single-cell analysis to alleviate batch effects, they often fail to retain the structural information of heterogeneous cell groups. To better capture such diversity, CellPLM employs a Gaussian Mixture Variational Autoencoder (GMVAE).

Generative model. For each cell i ($i \in \{1, \ldots, N\}$),

$$p(y_i; \pi) = \text{Multinomial}(\pi), \tag{11}$$

$$p(z_i \mid y_i) = \prod_{l=1}^{L} \mathcal{N}\left(\mu_{y_i,l}, \mathsf{diag}(\sigma^2_{y_i,l})\right), \tag{12}$$

$$p_{\theta_{\mathsf{dec}}}(x_i \mid z_i) = \mathcal{N}\left(\mu_{z_i}, \sigma^2 I\right). \tag{13}$$

Here, y_i is a one-hot latent cluster indicator with prior π. The parameters $\mu_{y_i,l}$ and $\sigma^2_{y_i,l}$ define the l-th Gaussian component, while μ_{z_i} and $\sigma^2 I$ denote the decoder's output mean and variance. We assume σ^2 is constant and parameterize μ_{z_i} by $\mu_{z_i} = f_{\mathsf{dec}}(z_i; \theta_{\mathsf{dec}})$.

Inference model. The variational posteriors are

$$q_{\eta_\mu,\eta_\sigma}(z_i \mid x_i) = \mathcal{N}\left(\widehat{\mu}_i, \mathsf{diag}(\widehat{\sigma^2_i})\right), \tag{14}$$

$$q_{\eta_\pi}(y_i \mid z_i) = \mathsf{Multinomial}\left(\widehat{\pi}_i\right), \tag{15}$$

with estimates computed as

$$h_i = f_{\mathsf{enc}}(x_i; \eta_{\mathsf{enc}}), \tag{16}$$

$$\hat{\mu}_i = f_\mu(h_i; \eta_\mu), \tag{17}$$

$$\log(\hat{\sigma}^2_i) = f_\sigma(h_i; \eta_\sigma), \tag{18}$$

$$\hat{\pi}_i = f_\pi(z_i; \eta_\pi). \tag{19}$$

Here $f_{\mathsf{enc}}(\cdot; \eta_{\mathsf{enc}})$ is the Transformer encoder, and f_μ, f_σ, f_π are small neural networks.

Masking for CellPLM pre-training. In pre-training, CellPLM applies masked modeling to the input matrix $X \in \mathbb{R}^{N \times k}$. Let $\mathcal{M}$ be the set of masked entries and define an indicator matrix M by

$$M_{i,j} = \begin{cases} 1 & \text{if } (i,j) \notin \mathcal{M}, \\ 0 & \text{if } (i,j) \in \mathcal{M}. \end{cases}$$

The corrupted input is then $\tilde{X} = M \odot X$, where $\odot$ denotes element-wise multiplication.

The derivation of $\mathcal{L}_{\mathsf{CellLM}}$ proceeds as follows. Our modeling objective is to construct a model that can explain the observed data X, i.e., to estimate the distribution $p(X)$ that maximizes the likelihood of X. Taking the logarithm is useful because it converts products into sums, which makes optimization easier to handle.

However, $p(X)$ cannot be directly computed. Therefore, we introduce a latent variable z, assuming

that a conditional distribution exists, and use

$$p(X) = \int p(X, z)\, dz.$$

Furthermore, since CellPLM is based on a Gaussian mixture VAE, we additionally introduce a cluster latent variable y that selects among different Gaussian distributions. The final objective is:

$$\log p(X) = \log \int \sum_y p(X, z, y)\, dz$$

This integral is intractable due to the presence of multiple latent variables and the complexity of the integration. To resolve this, we use variational inference, introducing an approximate posterior $q(z, y \mid \tilde{X})$. Then:

$$\log p(X) = \log \int \sum_y \frac{q(z, y \mid \tilde{X})}{q(z, y \mid \tilde{X})} p(X, z, y)\, dz$$

By moving the logarithm inside the expectation, we obtain:

$$\log p(X) = \log \mathbb{E}_{q(z,y|\tilde{X})} \left[\frac{p(X, z, y)}{q(z, y \mid \tilde{X})} \right]$$

At this stage, we use the definition of KL divergence, $\mathrm{KL}(q \,\|\, p) = \mathbb{E}_q \Big[\log \frac{q}{p} \Big]$, which leads to:

$$\mathrm{KL}\Big(q(z, y \mid \tilde{X}) \,\|\, p(z, y \mid X) \Big) = \mathbb{E}_{q(z,y|\tilde{X})} \left[\log \frac{q(z, y \mid \tilde{X})}{p(z, y \mid X)} \right]$$

Applying $p(z, y \mid X) = \frac{p(X,z,y)}{p(X)}$, we get:

$$\mathrm{KL}\Big(q(z, y \mid \tilde{X}) \,\|\, p(z, y \mid X) \Big) = \mathbb{E}_{q(z,y|\tilde{X})} \left[\log \frac{q(z, y \mid \tilde{X})}{p(X, z, y)/p(X)} \right]$$

Expanding further:

$$\mathrm{KL}(q \,\|\, p) = \mathbb{E}_q \Big[\log q(z, y \mid \tilde{X}) - \log p(X, z, y) + \log p(X) \Big]$$

$$= -\mathbb{E}_q \left[\log \frac{p(X, z, y)}{q(z, y \mid \tilde{X})} \right] + \log p(X)$$

Thus:

$$\log p(X) = \underbrace{\mathbb{E}_{q(z,y|\tilde{X})} \left[\log \frac{p(X, z, y)}{q(z, y \mid \tilde{X})} \right]}_{\text{ELBO}} + \underbrace{\mathrm{KL}\Big(q(z, y \mid \tilde{X}) \,\|\, p(z, y \mid X) \Big)}_{\text{Cost}}$$

Since KL divergence is always non-negative:

$$\log p(X) \geq \underbrace{\mathbb{E}_{q(z,y|\tilde{X})}\left[\log \frac{p(X,z,y)}{q(z,y \mid \tilde{X})}\right]}_{\text{ELBO}}$$

Therefore, maximizing the ELBO becomes our optimization objective. From the generative model, we can factorize as:

$$p_\theta(X,z,y) = p_\theta(X \mid z)\, p_\theta(z \mid y)\, p_\theta(y).$$

From the inference model:

$$q_\eta(z,y \mid \tilde{X}) = q_{\eta_{\text{enc}}}(z \mid \tilde{X})\; q_{\eta_\pi}(y \mid z).$$

Moreover, since the masking process is stochastic, a single X can lead to multiple masked versions $\tilde{X}$. Therefore, the model must perform well under all possible maskings. This is achieved by averaging over $\tilde{X} \sim p(\tilde{X} \mid X)$, and the ELBO becomes:

$$\log p(X) \geq \mathbb{E}_{\tilde{X}\sim p(\tilde{X}|X)}\left[\mathbb{E}_{q(z,y|\tilde{X})}\left[\log \frac{p(X,z,y)}{q(z,y \mid \tilde{X})}\right]\right]$$

Expanding the above:

$$\mathcal{L}_{\text{CellPLM}} = \mathbb{E}_{q(z,y|\tilde{X})}\, \mathbb{E}_{\tilde{X}\sim p(\tilde{X}|X)}\Big[\log p(X,z,y) - \log q(z,y \mid \tilde{X})\Big]$$

$$= \mathbb{E}_{z\sim q_{\eta_{\text{enc}}}(z|\tilde{X}),\, y\sim q_{\eta_\pi}(y|z)}\Big[\underbrace{\log p_\theta(X \mid z)}_{\mathcal{L}_{\text{recon}}} + \underbrace{\log p_\theta(z \mid y) - \log q_{\eta_{\text{enc}}}(z \mid \tilde{X})}_{z\text{-norm}} + \underbrace{\log p_\theta(y) - \log q_{\eta_\pi}(y \mid z)}_{y\text{-norm}}\Big]$$

Each term can now be rewritten. For the reconstruction term, since $\log p_\theta(X \mid z)$ does not depend on y, the $q_{\eta_\pi}(y \mid z)$ term becomes 1 and disappears. Thus, the reconstruction term $\mathcal{L}_{\text{recon}}$:

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{q_{\eta_{\text{enc}}}(z|\tilde{X})}\, \mathbb{E}_{p(\tilde{X}|X)}\big[\log p_{\theta_{\text{dec}}}(X \mid z)\big]$$

For the z-norm term (Conditional prior term, $\mathcal{L}_{\text{cond}}$ [2]):

$$\mathcal{L}_{\text{cond}} = \mathbb{E}_{z\sim q_{\eta_{\text{enc}}}(z|\tilde{X}),\, y\sim q_{\eta_\pi}(y|z)}\Big[\log p_\theta(z \mid y) - \log q_{\eta_{\text{enc}}}(z \mid \tilde{X})\Big]$$

Since

$$\int q_{\eta_{\text{enc}}}(z \mid \tilde{X})\Big[\log p_\theta(z \mid y) - \log q_{\eta_{\text{enc}}}(z \mid \tilde{X})\Big]\, dz = -\text{KL}\Big(q_{\eta_{\text{enc}}}(z \mid \tilde{X}) \,\|\, p_\theta(z \mid y)\Big),$$

we obtain:

$$\mathcal{L}_{\text{cond}} = -\mathbb{E}_{q_{\eta_\pi}(y|z)}\Big[\text{KL}\Big(q_{\eta_{\text{enc}}}(z \mid \tilde{X}) \,\|\, p(z \mid y)\Big)\Big]$$

For the y-norm term (Y prior term, $\mathcal{L}_{\mathcal{Y}}$ [2]):

$$\mathcal{L}_{\mathcal{Y}} = \mathbb{E}_{z\sim q_{\eta_{\text{enc}}}(z|\tilde{X}),\, y\sim q_{\eta_\pi}(y|z)}\Big[\log p_\theta(y) - \log q_{\eta_\pi}(y \mid z)\Big]$$

$$\mathbb{E}_{q_{\eta_\pi}(y|z)}\big[\log p_\theta(y) - \log q_{\eta_\pi}(y \mid z)\big] = -\mathrm{KL}(q_{\eta_\pi}(y \mid z) \,\|\, p_\theta(y))$$

$$\mathcal{L}_{\mathcal{Y}} = -\mathbb{E}_{q_{\eta_{\text{enc}}}(Z|\tilde{X})}\Big[\mathrm{KL}(q_{\eta_\pi}(Y \mid Z) \,\|\, p(Y))\Big]$$

Finally, note that $\mathcal{L}_{\text{cond}}$ and $\mathcal{L}_{\mathcal{Y}}$ are both computed from $q_{\eta_{\text{enc}}}(Z \mid \tilde{X})$ given a fixed $\tilde{X}$, so they do not require averaging over masks. The final CellPLM loss is therefore:

$$\mathcal{L}_{\text{CellLM}} = \mathbb{E}_{q_\eta(z,y|\tilde{X})}\, \mathbb{E}_{p(\tilde{X}|X)}\left[\ln \frac{p_\theta(X, Z, Y)}{q_\eta(Z, Y \mid \tilde{X})}\right] \tag{20}$$

$$= \underbrace{\mathbb{E}_{q_{\eta_{\text{enc}}}(Z|\tilde{X})}\, \mathbb{E}_{p(\tilde{X}|X)}\big[\log p_{\theta_{\text{dec}}}(X \mid Z)\big]}_{\mathcal{L}_{\text{recon}}} - \underbrace{\mathbb{E}_{q_{\eta_\pi}(Y|Z)}\Big[\mathrm{KL}\Big(q_{\eta_{\text{enc}}}(Z \mid \tilde{X}) \,\|\, p(Z \mid Y)\Big)\Big]}_{\mathcal{L}_{\text{cond}}}$$

$$- \underbrace{\mathbb{E}_{q_{\eta_{\text{enc}}}(Z|\tilde{X})}\Big[\mathrm{KL}(q_{\eta_\pi}(Y \mid Z) \,\|\, p(Y))\Big]}_{\mathcal{L}_Y}$$

Together, these three terms ensure that the latent space preserves biological structure while supporting denoising of masked expressions.

2.2.4 Batch-Aware Decoder

To account for technical variation across batches, a batch-specific embedding vector is retrieved from a learnable table and added to the latent representation. This adjusted input, defined as $h^{(0)} = z + b$ is then passed through multiple feed-forward layers to produce the reconstructed gene expression profile. In this way, batch effects are absorbed at the decoder stage, enabling the latent space to remain biologically meaningful and batch-invariant. Formally, the decoder can be written as:

$$\mathbf{h}^{(l)} = \mathrm{FFLayer}^{(l)}\left(\mathbf{h}^{(l-1)}\right)$$

2.3 Model Pre-training & Fine-tuning

2.3.1 Pre-training

The pre-training objective Eq(20) is defined to model the cell language by predicting masked gene expressions. Unlike text masking, where entire tokens may be masked, CellPLM masks only sub-

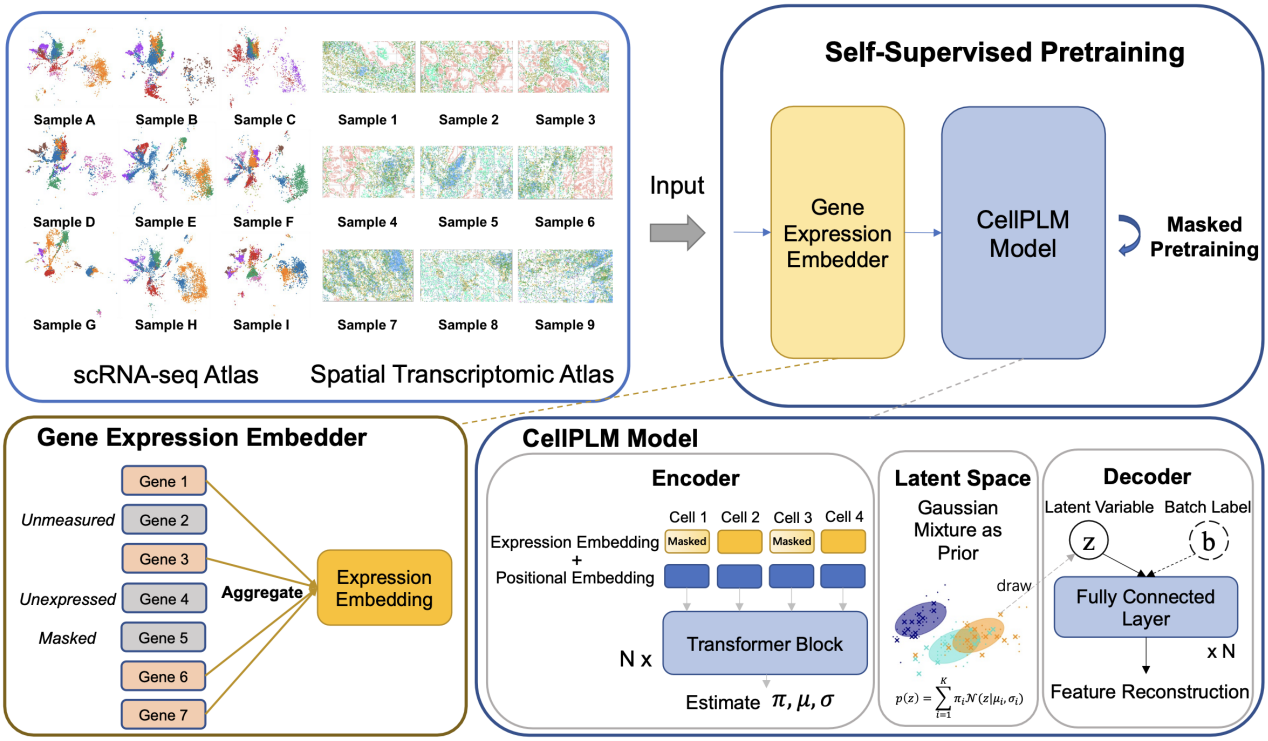

Fig. 3. An illustration of CellPLM pretraining framework[2]

sets of genes within a cell to avoid unrealistic imputation. This design choice is important because the number of measurable genes differs substantially between scRNA-seq and ST data. If an entire cell were masked, the model would be forced to infer a full gene expression profile, which is an ill-posed problem.

The objective is optimized using the reparameterization trick, Monte Carlo sampling, and independence assumptions, while replacing the reconstruction term with a mean squared error (MSE) loss. Using the reparameterization trick:

$$z = \mu + \sigma \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{21}$$

Monte Carlo sampling:

$$\mathbb{E}_{q_\phi(z|\tilde{x})}\left[\log p_\Theta(\tilde{x} \mid z)\right] \approx \frac{1}{S}\sum_{s=1}^{S} \log p_\Theta(\tilde{x} \mid z_s)$$

The pre-training objective for masked gene prediction is formulated as:

$$\mathcal{L}_{\text{MSE}} = \left\| \mathbf{M} \odot \left(\mathbf{H}^{(L)} - (1 - \mathbf{M}) \odot \mathbf{X}\right) \right\|_F^2, \quad \mathcal{L}_{\text{train}} = \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{cond}} + \mathcal{L}_Y \tag{22}$$

where **H** indicates the decoder output.

2.3.2 Fine-tuning

Fig. 4 demonstrates the framework of each downstream task. Our baseline zero-shot inference pipeline for cell clustering comprises an embedder, encoder, variational autoencoder (VAE), and an embedder head that generates latent representations. The architecture for cell type annotation includes an embedder, encoder, autoencoder, and an annotation head. For spatial transcriptomics imputation, an embedder, encoder, VAE, and a Negative Binomial MLP decoder are used.

Cell Embedding. Since no established cell embedding clustering pipeline exists for fine-tuning, we explored multiple architectural choices and evaluated three different loss functions. First, we experimented with a **decoder-based reconstruction loss** using the latent layer of the VAE, motivated by its alignment with self-supervised learning paradigms and its presence in the original CellPLM. However, this approach performed poorly in clustering metrics such as ARI and NMI, as high reconstruction accuracy did not necessarily correlate with high-quality clustering.

Second, we investigated a **KL-only latent loss** without a decoder, under the assumption that our primary goal was to obtain embeddings that capture cell-cell relationships. However, this led to posterior collapse, with embeddings collapsing toward the prior distribution and losing informative variance necessary for meaningful representation.

Finally, we successfully adopted the **supervised contrastive loss (SupConLoss)**, incorporating an additional MLP head trained with a contrastive objective to enhance representation quality. SupConLoss extends traditional contrastive learning to the supervised setting by leveraging label information [23]: it encourages embeddings of samples from the same class to be pulled closer together while pushing apart those from different classes. Unlike triplet or N-pair losses, SupConLoss enables each anchor to consider multiple positive pairs, thus promoting richer intra-class alignment and inter-class separability.

Given a normalized feature vector z_i and a temperature scaling parameter τ, we compute pairwise cosine similarity and apply a log-softmax over all samples excluding the anchor itself. The SupConLoss [23] is defined as:

$$\mathcal{L}_{supCon} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(z_i\cdot z_p/\tau)}{\sum_{a\in A(i)}\exp(z_i\cdot z_a/\tau)}$$

where $P(i)$ denotes the set of all positive indices (i.e., same-class samples) for anchor i, and $A(i)$ represents the set of all other samples in the batch excluding i. This formulation explicitly structures the representation space using label supervision, enabling the model to learn highly discriminative

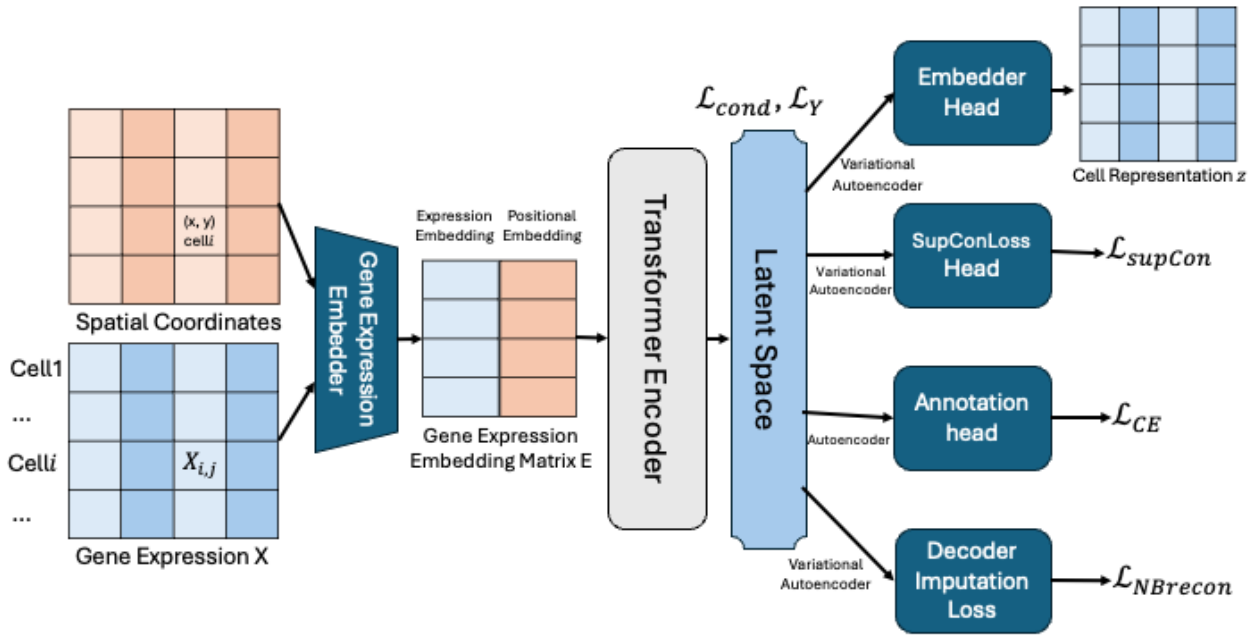

Fig. 4. An illustration of CellPLM downstream task framework

and generalizable embeddings.

Cell Type Annotation. A cross-entropy loss in Annotation head is applied between the predicted logits and true cell-type labels, using the top 3,000 highly variable genes as input features. The loss function is given by:

$$\mathcal{L}_{\text{CE}} = -\sum_{c} y_c \log \hat{y}_c$$

Spatial Transcriptomic Imputation. In this task, a subset of cells is masked, and their gene expression is imputed using reference scRNA-seq data. The Negative Binomial distribution is chosen because it better captures the overdispersed and sparse count nature of gene expression data.

The Negative Binomial distribution is formulated as:

$$P(X = x) = \frac{\Gamma(x+\theta)}{x!\,\Gamma(\theta)} \left(\frac{\theta}{\theta+\mu}\right)^{\theta} \left(\frac{\mu}{\theta+\mu}\right)^{x},$$

and the corresponding loss is:

$$\mathcal{L}_{\text{NB}} = -\log \text{NB}(X; \mu, \theta), \tag{23}$$

where μ is the predicted mean expression and θ is the dispersion parameter.

3 Experiments

3.1 Experiment Settings

Hardware Environment. All experiments were conducted on the EURO HPC supercomputing infrastructure, equipped with an NVIDIA A100-SXM GPU with 64 GB of VRAM (Driver Version: 535.54.03, CUDA Version: 12.2). The GPU was configured in persistence mode and operated at a temperature of approximately 43°C in idle state. The maximum rated power consumption of the GPU is 458 W. No other processes were utilizing the GPU during the experiments.

Datasets. The datasets used in the experiments were selected such that none, except Lung Cancer, had been included in the pre-training of CellPLM. The complete list of pre-training datasets is provided in Table 2. The Lung Cancer dataset was included in the experiments to evaluate the performance of CellPLM on a dataset seen during pre-training.

As summarized in Table 3, the benchmark datasets include multiple scRNA-seq datasets (breast cancer, colorectal cancer, lung cancer, Liver cancer, Lung, and Aorta) as well as spatial transcriptomics datasets (Dorsolateral prefrontal cortex(DLPFC) Visium and MERFISH Mouse brain2). Note that DLPFC Visium and Mouse brain2 datasets refer to concatenated collections of 12 samples and 5 samples, respectively. The liver cancer and lung cancer datasets were preprocessed to retain only genes included in the pre-trained CellPLM gene list.

Source	Datasets
HTCA	HTAN_HTAPP, HTAN_Stanford, HTAN_Vanderbilt, HTAN_BU, cxg_PBMCs, EGAS00001004571_PBMCs, eQTLAutoimmune, covid19autoimmunityPBMCs, VanDerWijst-Human-10x5pv1, cxg_Airways, COMBAT2022, TabulaSapiens, PAN.A01.v01.raw_count.20210429.PFI.embedding
HCA	GTEx_8_tissues_snRNAseq_atlas_071421.public_obs
GEO	GSE139324, GSE136246, GSE179994, GSE131907, GSE171145, GSE139555, GSE156728_CD4, GSE148071, PMID_34663877, Qian_et_al_2020_LC, GSE176021, GSE156728_CD8
Other Atlas (deduplicated)	MalteEtAl_LungAtlas, TICAtlas

Table 2. List of datasets and data sources used for pretraining CellPLM [2].

Data Preprocessing. Names in `.obs` and `.var` were made unique, and the columns corresponding to batch information and cell type labels were standardized. If the gene expression matrix was not already in sparse format, it was converted to compressed sparse row (`csr`) format. The `common_preprocess` and `transcriptomics_dataset` functions built into CellPLM were used to convert AnnData objects into PyTorch tensors, while filtering out genes not present in the pretrained gene list. Additionally,

Table 3. Benchmark datasets used for evaluation.

Dataset	Type	Number of Cells	Number of Genes	Source
Breast Cancer	scRNA-seq	10,639	18,064	[24]
Colorectal Cancer	scRNA-seq	24,349	21,383	GSE132465 (Subset) [25]
Lung	scRNA-seq	32,472	15,148	[26]
Liver Cancer	scRNA-seq	56,721	407	GSE151530 [27]
Lung Cancer	scRNA-seq	208,506	423	GSE131907 [28]
Aorta	scRNA-seq	48,082	12,382	GSE155468 [29]
Frontal Cortex	scRNA-seq	10,319	34,305	GSE97930 [30]
Mouse Brain	scRNA-seq	14,437	23,284	GSE87544 [31]
DLPFC	ST	47,681	33,538	Visium [32]
Mouse Brain2	ST	28,317	155	MERFISH [33]

when the `.var` index was in gene symbol format rather than ENSEMBL ID, it was converted accordingly.

Baseline Selection and Downstream Task Setup. For clustering, CellPLM embeddings were compared against PCA embeddings computed with the Scanpy library, as well as results reported in the DeepST paper. For cell type annotation, results were compared with those reported in the SToFM paper. Unfortunately, benchmark scores for ST imputation on DLPFC and Mouse Brain2 are not available, as these datasets were released only recently. All downstream tasks were tuned via hyperparameter optimization.

3.2 Evaluation Metrics

3.2.1 Cell Embedding clustering

Clustering evaluation metrics included the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). The ARI is based on the Rand Index (RI), which measures the proportion of agreement (both same-cluster and different-cluster assignments) between the predicted clustering and the ground truth. ARI corrects RI by adjusting for chance grouping, and ranges from -1 to 1:

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]} \tag{24}$$

Here, RI is the Rand Index, $\mathbb{E}[\mathrm{RI}]$ is its expected value under random clustering, and $\max(\mathrm{RI})$ is the maximum possible value. A value of 0 indicates random clustering, while higher values indicate stronger agreement between predicted and true clusters.

Normalized Mutual Information (NMI) quantifies the information shared between predicted labels U

and ground-truth labels V. It is defined as:

$$\text{NMI}(U, V) = \frac{2 \cdot I(U; V)}{H(U) + H(V)}, \tag{25}$$

where $I(U; V)$ denotes the mutual information between the predicted clustering U and the ground-truth clustering V, and $H(\cdot)$ is the Shannon entropy. NMI ranges from 0 (independent assignments) to 1 (perfect alignment). Both ARI and NMI were computed using `adjusted_rand_score` and `normalized_mutua` from the `sklearn` library. We selected these two metrics because they are widely used in the machine learning community to quantitatively evaluate clustering performance against ground-truth labels.

3.2.2 Cell type annotation

The goal of cell type annotation is to classify cells in the query dataset according to the labels of a reference dataset. During training, true labels are provided and the model parameters are optimized in a supervised manner. In classification tasks, we denote:

- TP: True Positives
- FP: False Positives
- TN: True Negatives
- FN: False Negatives

The evaluation metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{26}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{27}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{28}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{29}$$

Accuracy measures the proportion of correctly predicted samples and is the most intuitive metric. However, when the dataset is imbalanced, high accuracy may not be meaningful, as it can be dominated by the majority class. Precision measures the fraction of correctly predicted positive samples among all predicted positives, making it important in applications where false positives are costly

(e.g., spam detection). Recall measures the fraction of correctly predicted positives among all actual positives, which is critical when false negatives should be minimized (e.g., cancer diagnosis). The F1-score is the harmonic mean of Precision and Recall, balancing the trade-off between the two.

Micro vs. Macro F1 There are two common variants of F1-score.

Micro F1: For Micro F1, TP, FP, and FN are aggregated across all classes before computing the score:

$$\text{Micro F1} = \frac{2 \cdot \sum \text{TP}}{2 \cdot \sum \text{TP} + \sum \text{FP} + \sum \text{FN}}. \tag{30}$$

Because it reflects the average performance over all samples, majority classes have a greater influence on Micro F1. In highly imbalanced datasets, Micro F1 tends to be dominated by the performance on large classes.

Macro F1: For Macro F1, the F1-score is calculated for each class individually and then averaged:

$$\text{Macro F1} = \frac{1}{K} \sum_{k=1}^{K} \text{F1}_k, \tag{31}$$

where K is the number of classes. Each class contributes equally, regardless of its frequency. Therefore, Macro F1 is more sensitive to poor performance on minority classes.

In this study, minority classes play a crucial role in cell type annotation (e.g., rare cell populations). Hence, we adopt Macro F1 as the primary evaluation metric, while also reporting Accuracy for completeness and interpretability. These metrics were computed using `multiclass_f1_score`, `multiclass_accurac` `multiclass_precision`, and `multiclass_recall` from the `torchmetrics.functional.classification` library.

3.2.3 Spatial Transcriptomics Imputation

Spatial transcriptomics (ST) datasets shows high resolution to represent the information of spatial coordinates of cells. This brought a huge change in single-cell analysis but in order to preserve location data, ST data generally contain far fewer genes than scRNA-seq datasets, making it necessary to predict (impute) missing genes using reference scRNA-seq data. The evaluation metrics are as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{32}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{33}$$

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{34}$$

$$\text{PCC} = \frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_i (y_i - \bar{y})^2}\sqrt{\sum_i (\hat{y}_i - \bar{\hat{y}})^2}} \tag{35}$$

$$\text{Cosine}(y, \hat{y}) = \frac{y \cdot \hat{y}}{\|y\|\|\hat{y}\|} \tag{36}$$

Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) directly measure the numerical distance between predicted and ground truth values. Among them, MSE is more sensitive to large errors because of squaring, while MAE and RMSE are easier to interpret. Since spatial imputation focuses on predicting masked gene expression values, these metrics are useful for quantifying absolute accuracy.

Pearson's Correlation Coefficient (PCC) evaluates the degree of linear correlation between predicted and actual expression profiles. A high PCC indicates that the predicted expression pattern closely follows the true one, which is biologically meaningful when spatial trends matter more than exact values. Cosine similarity focuses on the angle between vectors, thus capturing similarity in overall expression patterns regardless of magnitude. This metric complements PCC by emphasizing directionality of expression changes. Taken together, these five metrics provide a balanced evaluation: **absolute accuracy (MSE/MAE/RMSE) and pattern similarity (PCC/Cosine).**

During evaluation, 100 target genes were masked in the query ST dataset. Target genes were selected by stratifying all genes into four bins by sparsity and sampling 25 genes from each [2]. Only genes included in the pretrained gene embedding list (13,500 genes) were eligible for imputation.

3.3 Cell Embedding (PCA – Zero-shot – Fine-tuning)

Zero-shot cell embedding clustering in CellPLM was performed by generating latent embeddings from the pretrained parameters and applying the Leiden algorithm in Scanpy (`sc.tl.leiden`). The Leiden algorithm is a graph-based clustering method, in contrast to K-means which relies solely on Euclidean distances. Since single-cell RNA-seq data are high-dimensional and characterized by

Table 4. Clustering performance of cell embeddings across datasets. "*" denotes Spatial Transcriptomics (ST) data. Bold figures are the best scores.

	DLPFC*		Mouse Brain*	
	ARI	NMI	ARI	NMI
PCA	0.09	0.18	0.26	0.61
CellPLM(zero-shot)	0.20	0.28	0.46	0.54
CellPLM(fine-tuned SuperConLoss)	0.51 ± 0.02	0.66 ± 0.03	0.70 ± 0.04	0.71 ± 0.02
CellPLM(fine-tuned CE)	$\mathbf{0.60 \pm 0.01}$	$\mathbf{0.74 \pm 0.00}$	$\mathbf{0.96 \pm 0.00}$	$\mathbf{0.94 \pm 0.00}$
	Breast		**Aorta**	
	ARI	NMI	ARI	NMI
PCA	0.22	0.59	0.24	0.54
CellPLM(zero-shot)	0.59	0.71	0.87	0.84
CellPLM(fine-tuned SuperConLoss)	$\mathbf{0.94 \pm 0.03}$	0.88 ± 0.03	$\mathbf{0.96 \pm 0.01}$	0.92 ± 0.01
CellPLM(fine-tuned CE)	0.80 ± 0.15	$\mathbf{0.90 \pm 0.03}$	0.96 ± 0.03	$\mathbf{0.95 \pm 0.03}$

Table 5. Clustering training time of cell embeddings across datasets on an NVIDIA A100-SXM 64GB GPU. "*" denotes Spatial Transcriptomics (ST) data.

	DLPFC*	Mouse Brain*	Breast	Aorta
CellPLM(fine-tuned SuperConLoss)	**05m 51s**	**59s**	**01m 23s**	**20m 11s**
CellPLM(fine-tuned CE)	9h 43m 26s	4h 41m 21s	2h 02m 32s	9h 34m 02s

nonlinear relationships, graph-based approaches leveraging KNN graphs and modularity optimization are more appropriate than simple distance-based clustering methods. Due to project time constraints, we conducted three seeded runs for each fine-tuning configuration and report the mean and standard deviation, whereas for both PCA and CellPLM zero-shot inference, we performed only a single experiment per dataset.

As shown in Table 4, CellPLM consistently outperformed PCA across all datasets in zero-shot clustering tasks, indicating that the pretrained representations effectively capture both gene–cell and cell–cell relationships. In scRNA-seq datasets, zero-shot ARI and NMI scores consistently exceeded 0.5, and in the Aorta dataset, CellPLM achieved ARI/NMI scores above 0.8 without any task-specific finetuning.

For ST datasets, although CellPLM still surpassed PCA embeddings, the absolute clustering performance was comparatively lower. This discrepancy is likely attributable to the smaller scale of the ST pretraining corpus (2.7 million cells) relative to the scRNA-seq corpus (8.7 million cells), along with the inherent sparsity and reduced gene coverage in ST data. For example, the MERFISH mouse brain dataset contains only 155 genes, compared to 23,284 genes in its scRNA-seq counterpart (GSE87544 mouse brain). Furthermore, ST data are collected at the spot level, where signals from multiple cells are aggregated, potentially distorting gene expression profiles and introducing measurement noise from imaging-based technologies.

As described in Section 2, we implemented a fine-tuning pipeline using Supervised Contrastive Loss (SupConLoss) and compared its performance with a standard Cross Entropy Loss (CELoss) approach for cell type annotation. Since both methods produce latent representations, we used the resulting embeddings for downstream clustering evaluation. The Fig. 5 and Fig. 6 present clustering results under different strategies, demonstrating that fine-tuning leads to higher ARI and NMI scores as well as more distinct cluster boundaries. Notably, SupConLoss results on Breast and Aorta scRNA-seq datasets are competitive with those of CELoss.

Although CELoss fine-tuning resulted in higher absolute performance (e.g.,Mouse Brain2 ARI/NMI ≈ 0.95) compared to SupConLoss (≈ 0.70), it required significantly more training time, approximately 280 minutes versus 59 seconds. As a result, as shown in Table 5, CellPLM fine-tuned with SupConLoss demonstrates substantially improved efficiency, achieving approximately 30 times to 300 times faster training compared to CellPLM fine-tuned with CELoss, depending on the dataset. This discrepancy can be attributed to several factors.

First, CELoss training involves computing classification metrics such as accuracy, F1 score, precision, and recall at every batch. These metrics are aggregated at the epoch level and directly influence the model optimization, introducing additional computational overhead. Second, the loss landscape of Cross Entropy is inherently geared toward precise classification. The optimization must continuously align model outputs with one-hot label distributions, often requiring more sensitive and fine-grained gradient updates. As a result, the training process is not only longer, but also more computationally intensive. Finally, CELoss requires full label supervision, which restricts its scalability in partially labeled or large-scale datasets, and increases the amount of training data actively used in each step.

In contrast, SupConLoss does not rely on batch-wise evaluation metrics or full label coverage. Since ARI and NMI require complete predictions, they cannot be computed during training iterations. The absence of such runtime evaluations, combined with a representation-centric optimization goal, makes SupConLoss computationally lightweight.

Taken together, these results suggest that SupConLoss offers a favorable balance between efficiency and clustering quality. In particular, for scRNA-seq datasets where partial labels, limited annotations, and large-scale data are common, SupConLoss represents a more practical and scalable alternative to CELoss while still delivering competitive performance.

DLPFC: Individual Sample Experiments.

We further conducted clustering on each individual DLPFC Visium sample using cell-level embeddings. Although DeepST operates at the spot level for spatial domain recognition, both methods re-

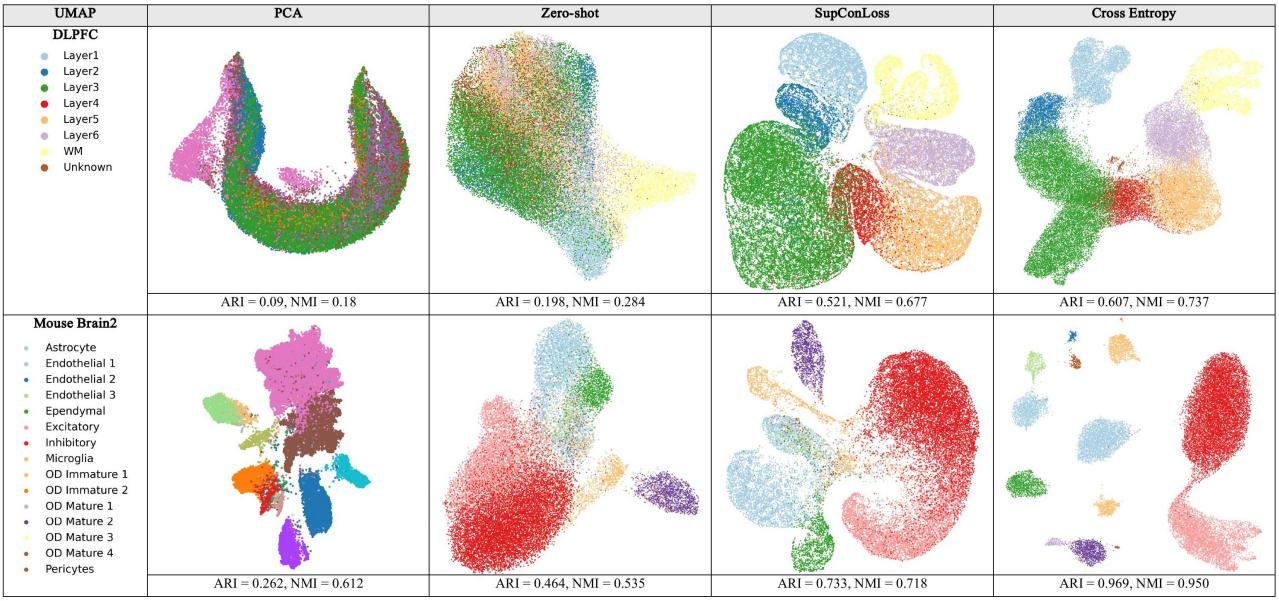

Fig. 5. UMAP visualization of cell representations on the DLPFC and Mouse Brain2 dataset.

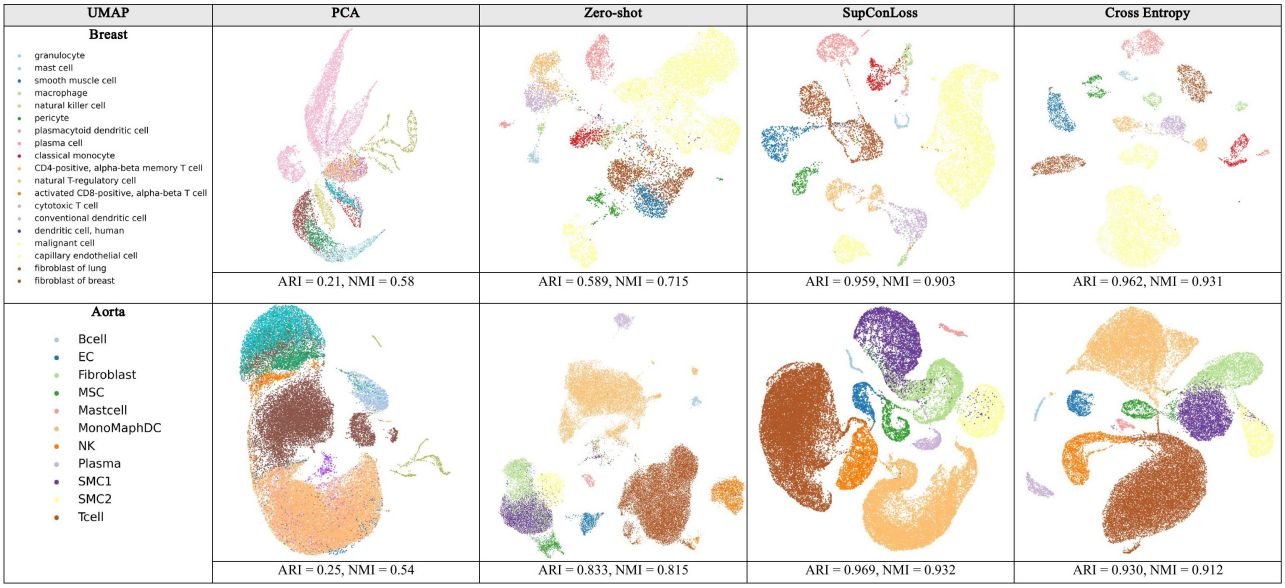

Fig. 6. UMAP visualization of cell representations on the Breast and Aorta dataset.

port ARI scores for DLPFC layer clustering, allowing for a direct comparison.

It is important to note, however, that DeepST is a self-supervised method, whereas CellPLM fine-tuned with SupConLoss is weakly supervised, and CellPLM fine-tuned with CELoss is fully supervised. Therefore, the ARI comparison should not be interpreted strictly as an indication of superiority, but rather as an illustration of the differences among unsupervised, weakly supervised, and fully supervised learning paradigms.

Before discussing the results in detail, we emphasize that three seeded runs were conducted for SupConLoss fine-tuning, and the mean and standard deviation are reported. In contrast, for PCA, CellPLM zero-shot inference, and some of the CELoss experiments, we performed only a single run

per dataset.

Overall, as shown in Table 6, the fully supervised variant of CellPLM consistently achieved superior performance compared to the other methods across most samples. Notably, for slices Sample 3 and Sample 9, DeepST outperformed CellPLM fine-tuned with CELoss, highlighting dataset-specific variability. Nevertheless, Fig. 7 corroborates that supervised training with comprehensive label information provides the most powerful embeddings, underscoring the advantage of fully supervised learning when annotations are available.

Interestingly, the aggregated DLPFC dataset exhibited relatively small standard deviations across runs (e.g., SupConLoss: 0.02, CELoss: 0.03), whereas most individual samples showed substantially larger variances. For instance, in Sample 4, the standard deviation reached 0.675 for SupConLoss and 0.206 for CELoss. This discrepancy can be attributed to the limited number of cells in individual samples. As shown in Table 3, Sample 4 contains only 3,460 cells, which is less than one-tenth of the 47,681 cells in the complete DLPFC dataset. These findings suggest that deep learning models yield more stable and consistent performance when trained on large-scale data, while small datasets are more susceptible to sampling noise and variability.

Additionally, we observed that for a specific sample (e.g., Sample 5), supervised contrastive learning (0.229) performed worse than the zero-shot baseline (0.245). This discrepancy may stem from various factors such as data imbalance, label noise, or ambiguous cluster boundaries. However, the exact cause has not been clearly identified in this study.

As a reference, we also experiment for Mouse brain2 sample1 (Fig. 8), it demonstrates better performance for every different strategies than DLPFC even they are both ST dataset. The discrepancy between Visium and MERFISH performance arises from fundamental differences in resolution and profiling. Visium provides whole-transcriptome measurements at spot-level resolution, where multiple cells contribute to each signal, resulting in increased noise and blurred boundaries. Conversely, MERFISH targets predefined marker genes at single-cell resolution, producing cleaner and more distinct expression profiles that align closely with anatomical structures, thereby enhancing clustering accuracy and ARI.

3.4 Cell Type Annotation (Zero-shot – Fine-tuning)

DLPFC Layer Segmentation Note that for the DLPFC dataset, we performed layer segmentation. Technically, the principle is identical to cell type annotation: the ground truth label is not the *cell type* but the *layer*. Therefore, we include layer segmentation in the cell type annotation section.

Table 6. Clustering performance on individual DLPFC samples. "*" indicates results reported in [17].

	Sample1 ARI	**Sample2** ARI	**Sample3** ARI	**Sample4** ARI
PCA	0.155	0.09	0.09	0.18
CellPLM(zero-shot)	0.276	0.295	0.305	0.278
DeepST*	0.514	0.622	**0.765**	0.479
CellPLM(SuperConLoss)	0.306 ± 0.100	0.600 ± 0.013	0.608 ± 0.010	0.588 ± 0.675
CellPLM(fine-tuned CE)	$\mathbf{0.621 \pm 0.08}$	$\mathbf{0.691 \pm 0.011}$	0.609 ± 0.062	$\mathbf{0.713 \pm 0.206}$

	Sample5 ARI	**Sample6** ARI	**Sample7** ARI	**Sample8** ARI
PCA	0.20	0.16	0.18	0.14
CellPLM(zero-shot)	0.245	0.345	0.291	0.298
DeepST*	N/A	0.525	0.522	0.514
CellPLM(SuperConLoss)	0.229 ± 0.005	0.402 ± 0.1234	0.408 ± 0.735	0.387 ± 0.109
CellPLM(fine-tuned CE)	**0.578**	**0.857**	**0.547**	**0.520**

	Sample9 ARI	**Sample10** ARI	**Sample11** ARI	**Sample12** ARI
PCA	0.16	0.16	0.05	0.06
CellPLM(zero-shot)	0.403	0.235	0.235	0.317
DeepST*	**0.551**	0.574	0.514	0.524
CellPLM(SuperConLoss)	0.417 ± 0.092	0.458 ± 0.106	0.491 ± 0.095	0.519 ± 0.31
CellPLM(fine-tuned CE)	0.483	**0.783**	**0.774**	**0.537**

CellPLM	
encoder hidden dim	1024
encoder layers	4
latent dimension	512
decoder hidden dim	1024
model dropout	0.2
cell mask rate	0.75
gene mask rate	0.25
learning rate	5e-3
weight decay	1e-7
drop node rate	0.3
latent model	**autoencoder**
head type	annotation
highly variable genes (HVG)	3000
Positional encoding (PE)	**0**

Table 7. Hyperparameters for cell type annotation with the *CellPLM* model.

Table 7 summarizes the hyperparameters and model configurations used for the cell type annotation task. We tested several assumptions, and their validity was examined through the ablation study (Table 8):

1) **Increasing the number of highly variable genes (HVGs) would improve performance.** *Contrary to our expectation, performance did not change significantly when using 4,000 HVGs instead*

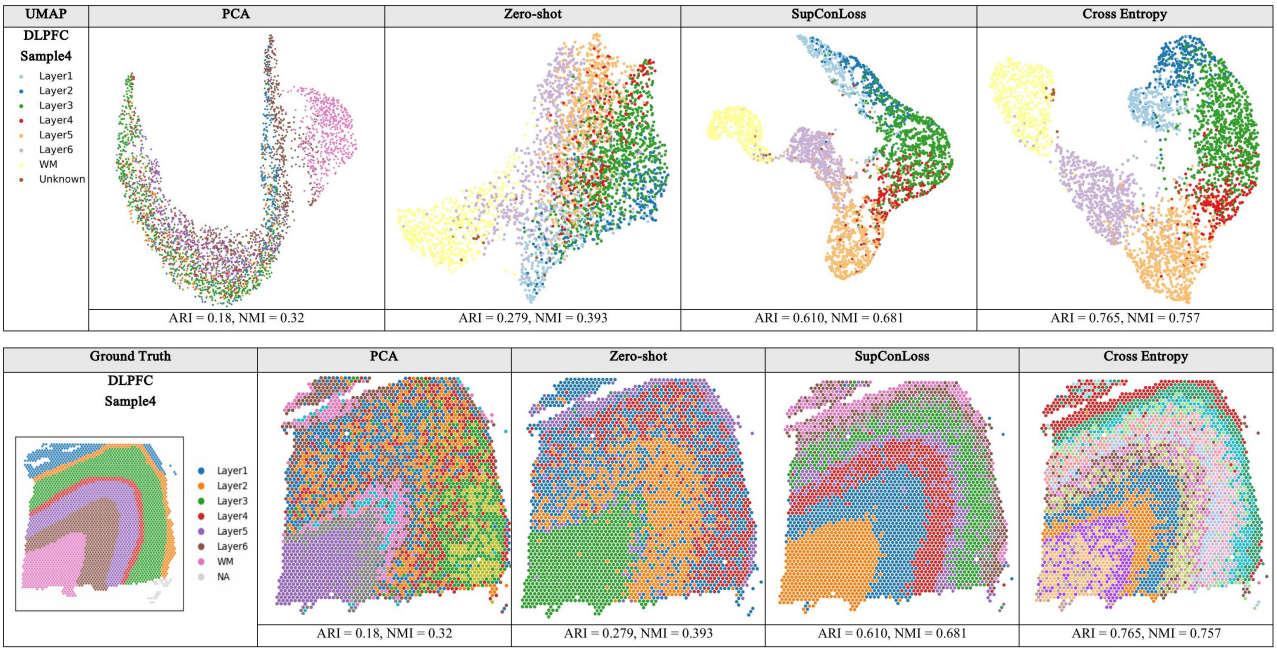

Fig. 7. UMAP Visualization and spatial layer segmentation of DLPFC Sample 4.

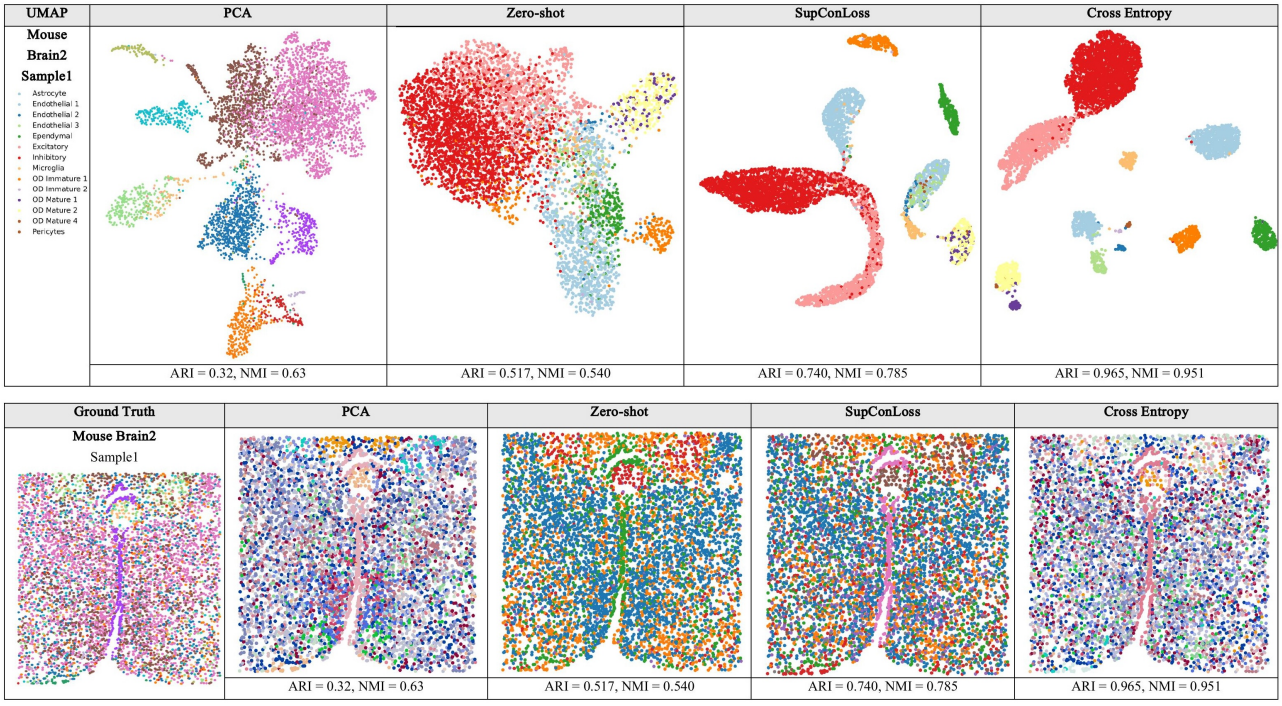

Fig. 8. UMAP Visualization and spatial cell clustering of Mouse Brain2 sample1.

DLPFC	Accuracy	F1
Base Result	**0.6963**	**0.6812**
hvg:4000	0.6846	0.6892
w PE	0.6717	0.6768
w/o Batch	0.6223	0.6257
w PE, latent mod: VAE	0.6834	0.6850
w PE, latent mod: GMVAE	0.6703	0.6797

Table 8. Ablation study results. VAE indicates Variational Autoencoder, GMVAE denotes Gaussian Mixture Variational Autoencoder.

of 3,000. This suggests that including less informative genes can introduce noise, which negatively impacts representation quality.

2) **Incorporating spatial coordinates via positional encoding (PE) would enhance classification.** *This assumption was not supported. Incorporating positional encoding degraded performance, likely raw spatial coordinates in Visium often reflect technical artifacts (spot density, slicing differences) rather than biologically meaningful structures, so they can act as noise in cell type classification.*

3) **A more complex latent space model (VAE or GMVAE) would yield better results than a simple autoencoder.** *Again, this assumption was invalid. The autoencoder achieved the best performance, possibly because cell type annotation requires sharp decision boundaries rather than generative flexibility, so stochastic latent variables (VAE, GMVAE) added unnecessary noise.*

4) **Batch information is not critical for classification.** *This assumption was disproven. Removing batch metadata significantly reduced performance, indicating that batch labels not only encode technical effects but also preserve meaningful sample heterogeneity. Removing batch information collapses such variation and harms classification accuracy.*

DLPFC: Individual Sample Experiments

Model	DLPFC1		DLPFC2		DLPFC3		DLPFC4		Average	
	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1
scGPT*	0.6943	0.6599	0.6565	0.5908	0.6919	0.6151	0.6696	0.6055	0.6781	0.6178
Geneformer*	0.6404	0.5770	0.6286	0.5464	0.6373	0.5677	0.6157	0.5513	0.6305	0.5606
Nicheformer*	0.5712	0.5319	0.5942	0.5300	0.5588	0.5192	0.5648	0.5141	0.5723	0.5238
SToFM*	0.7082	0.6755	**0.6974**	0.6453	**0.7157**	**0.6659**	**0.6856**	**0.6274**	**0.7014**	0.6535
CellPLM(Zero-shot)	0.1472	0.0567	0.1653	0.0595	0.1354	0.0476	0.1394	0.0538	0.1468	0.0544
CellPLM(Fine-tuned)	**0.7110**	**0.7155**	0.6812	**0.6864**	0.6308	0.6319	0.6197	0.6201	0.6607	**0.6635**

Table 9. Performance of different models on the DLPFC layer segmentation task. Note that we conducted three seeded runs CellPLM fine-tuning and report the mean. "*" indicates results directly taken from [19].

As shown in Table 9, CellPLM performed worse than SToFM, especially in terms of Accuracy, although the gap in F1 was less pronounced or even favorable in some cases. The notable point is the substantially lower zero-shot performance compared to fine-tuned performance. The average accuracy (0.1468) is only marginally higher than the probability of randomly assigning a cell to one of the eight layers (0.125, i.e., 1/8). This finding is consistent with recent work by Kedzierska et al.[34], which demonstrated that the zero-shot inference performance of scRNA-seq foundation models often falls short of traditional, lighter machine learning approaches. These results highlight, in line with their conclusions, that supervised learning with labeled data remains essential for robust cell type annotation, as current foundation models cannot yet capture the full diversity of cell types across tissues and species. Nevertheless, the development of a universal foundation model for this

Table 10. Results of cell type annotation on various datasets. "*" indicates Spatial Transcriptomics (ST) data.

Models	DLPFC*		Mouse Brain*		Liver Cancer		Lung Cancer	
	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1
CellPLM(fine-tuned)	0.6963	0.6812	0.7960	0.8113	0.9060	0.8947	0.9551	0.9468
Models	Breast		Colorectal		Lung		Aorta	
	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1
CellPLM(fine-tuned)	0.8234	0.8092	1	1	0.9551	0.9468	0.9502	0.9543

Table 11. The result of Spatial transcriptomics imputation on DLPFC and Mouse Brain2. Note that we conducted three seeded runs CellPLM fine-tuning and report the mean. Bold figures are the best scores.

	DLPFC				
	MSE	RMSE	MAE	Corr	Cosine
CellPLM(zero-shot)	0.0886	0.3278	0.1783	0.1967	0.3743
CellPLM(fine-tuned w scRNA)	0.0858	0.3257	0.1917	0.2023	0.3804
CellPLM(fine-tuned w/o scRNA)	**0.0756**	**0.3079**	**0.1467**	**0.2105**	**0.3855**
	Mouse Brain2				
	MSE	RMSE	MAE	Corr	Cosine
CellPLM(zero-shot)	0.5360	0.6682	0.4580	0.1306	0.4307
CellPLM(fine-tuned w scRNA)	**0.4347**	**0.5881**	**0.3723**	**0.1638**	**0.4410**
CellPLM(fine-tuned w/o scRNA)	0.6298	0.6958	0.4726	0.0924	0.4025

task remains a promising avenue for future work.

Cell type annotation - Various datasets

Fine-tuning results (Table 10) followed the same trend as in the cell embedding task: scRNA-seq datasets consistently yielded higher performance than ST datasets. For Lung Cancer data, which were included in CellPLM pretraining, we observed near-perfect Accuracy (0.95), as expected. Importantly, datasets not included in pretraining—such as Aorta, Lung, and Colorectal cancer—also achieved high accuracy and F1-scores above 0.95, demonstrating the strong generalization ability of CellPLM.

3.5 Spatial Transcriptomics Imputation (Zero-shot – Fine-tuning)

Fine-tuning was performed using two approaches. First, ST was used as the query and matched scRNA-seq data as the reference; the last batch was used for validation. Second, the model was trained without scRNA-seq reference data.

As shown in Table 11, the effect of incorporating reference scRNA-seq data varied across datasets. For the DLPFC dataset, which already contains a relatively rich set of genes (33,538), the inclusion of reference data resulted in little to no improvement. In contrast, for the MERFISH Mouse

Brain2 dataset, which contains only 155 genes, leveraging reference scRNA-seq (23,284 genes in GSE87544) led to substantial improvements in error-based metrics (MSE, RMSE, MAE).

Overall, these findings suggest that the utility of reference scRNA-seq data depends strongly on dataset characteristics, particularly the number of measured genes and the degree of sparsity. While CellPLM can produce reasonable imputations in a zero-shot setting, fine-tuning with appropriate reference data becomes particularly beneficial when the ST dataset is extremely sparse. Nevertheless, all fine-tuned metrics differed by less than 0.1 compared to zero-shot inference. Moreover, PCC and cosine similarity values for CellPLM remained below 0.5 across datasets. This indicates that although the model captures certain cell–gene patterns, task-specific imputation models may still be necessary to fully recover fine-grained spatial expression trends.

4 Conclusions and future work

4.1 Conclusion

This thesis explored the design, implementation, and evaluation of Transformer-based foundation models for single-cell transcriptomics, with a particular focus on CellPLM. By revisiting the theoretical underpinnings of the model—its formulation of a cell language model, adoption of Flowformer for efficient attention, and integration of Gaussian mixture priors in the latent space—we provided a rigorous foundation for understanding its architecture.

Through extensive experiments, we demonstrated that CellPLM produces biologically meaningful embeddings and consistently surpasses classical baselines such as PCA in clustering tasks, even in zero-shot inference. Fine-tuning further enhanced performance, as reflected in improved Macro F_1 scores for cell type annotation. Our analysis of ST imputation revealed that the effectiveness of incorporating reference scRNA-seq data is highly dependent on dataset characteristics: while beneficial in sparse datasets such as MERFISH mouse brain2, it provides limited advantage for richer datasets like DLPFC Visium.

Importantly, we identified a trade-off between fine-tuning strategies. Cross-entropy loss achieved the strongest absolute performance but incurred significantly higher computational cost and required full supervision, whereas supervised contrastive learning achieved competitive clustering quality with substantially greater efficiency, making it more scalable for weakly supervised or semi-supervised contexts. These findings highlight that both methodological and practical considerations must guide the deployment of foundation models in single-cell biology.

Nevertheless, challenges remain. Overall clustering performance on ST datasets was weaker than on scRNA-seq datasets, likely due to the limited scale of pretraining and the inherent sparsity of ST data. Moreover, despite improved performance with fine-tuning, correlation-based metrics in ST imputation (PCC, Cosine) remained below 0.5, suggesting that foundation models still struggle to fully reconstruct spatial gene–gene patterns.

CellPLM, as a cell language model rather than a gene language model, primarily focuses on modeling cell–cell interactions and relationships while paying less direct attention to individual gene–gene interactions. This design limits the model's ability to explicitly capture regulatory relationships or co-expression patterns at the gene level. Encouragingly, recent ST foundation models such as SToFM [19] attempt to account for micro-scale gene expression patterns and demonstrate strong performance. Building on this, investigating how the absence of explicit gene-level relational learning affects downstream performance, and whether integrating modules dedicated to gene–gene interaction modeling could complement a cell-centric framework, remains an important direction for future research.

In addition, the experiments in this thesis were restricted to a fixed pretrained gene list (13,500 genes). When new query datasets include genes outside this list, performance may degrade, limiting applicability for biologists studying specific genes of interest. Extending pretraining to cover a broader gene space would help mitigate this limitation.

Another limitation concerns positional encoding. This thesis employed Sinusoidal Positional Encoding, but recent studies have proposed alternatives such as Learnable Positional Encoding [35], Relative Positional Encoding [36], and Rotary Positional Encoding [37]. How the choice of positional encoding impacts representation quality and downstream performance remains an open research question.

Due to project time constraints, we also did not explore additional tasks supported by the CellPLM framework, such as scRNA-seq denoising or perturbation prediction. Including these tasks would provide a more complete assessment of the model's versatility. Moreover, when comparing foundation models such as CellPLM and SToFM, differences in pretraining datasets complicate fair evaluation. Our comparison therefore relied on reported scores from prior work rather than fully aligned pretraining conditions, which future studies should address. As suggested in prior research, developing a standardized benchmark corpus for evaluation will be essential, similar to practices in other domains.

Furthermore, although promising self-supervised methods for clustering such as constrained pairwise clustering [38] have been proposed, we were unable to incorporate them in this work. How-

ever, as demonstrated here, semi-supervised learning with SupConLoss shows strong efficiency and competitive performance. Therefore, exploring the integration of state-of-the-art clustering methods into fine-tuning pipelines would be a fruitful avenue for future research.

Recent findings [1], [34] suggest that although single-cell foundation models can generalize across datasets, simpler models such as logistic regression or task-specific architectures may still outperform them in certain tasks, particularly in the zero-shot setting. Thus, the development of a universal foundation model for single-cell sequencing, one that achieves both broad transferability and biologically meaningful representations, remains an open challenge.

In conclusion, this thesis contributes both theoretical and practical insights into Transformer-based foundation models for single-cell data. By extending CellPLM and systematically benchmarking its performance, we demonstrated its strengths in generalization and efficiency, while also delineating its current limitations. Future work should focus on enlarging and diversifying pretraining corpora, developing hybrid architectures that better capture both cell–cell and gene–gene interactions, and designing more effective fine-tuning strategies to close the gap with specialized task-specific models. Such advances will be essential to realize the full potential of foundation models in single-cell and spatial omics research.

References

[1] A. Szałata *et al.*, "Transformers in single-cell omics: A review and new perspectives," *Nature Methods*, vol. 21, no. 8, pp. 1430–1443, Aug. 2024, Epub 2024 Aug 9. DOI: 10.1038/s41592-024-02353-z (cited on pp. 8, 40).

[2] H. Wen *et al.*, "Cellplm: Pre-training of cell language model beyond single cells," *bioRxiv*, 2023. DOI: 10.1101/2023.10.03.560734. eprint: https://www.biorxiv.org/content/early/2023/10/05/2023.10.03.560734.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2023/10/05/2023.10.03.560734 (cited on pp. 8–10, 13, 20–22, 25, 29, 45).

[3] Q. Li *et al.*, "Progress and opportunities of foundation models in bioinformatics," *Briefings in Bioinformatics*, vol. 25, no. 6, bbae548, Oct. 2024, ISSN: 1477-4054. DOI: 10.1093/bib/bbae548. eprint: https://academic.oup.com/bib/article-pdf/25/6/bbae548/60105244/bbae548.pdf. [Online]. Available: https://doi.org/10.1093/bib/bbae548 (cited on p. 8).

[4] C. V. Theodoris *et al.*, "Transfer learning enables predictions in network biology," *Nature*, vol. 618, no. 7965, pp. 616–624, Jun. 2023. DOI: 10.1038/s41586-023-06139-9 (cited on p. 8).

[5] R. Bommasani *et al.*, *On the opportunities and risks of foundation models*, 2022. arXiv: 2108.07258 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2108.07258 (cited on p. 8).

[6] OpenAI, “Gpt-4 technical report,” *CoRR*, vol. abs/2303.08774, 2023. DOI: 10.48550/ARXIV.2303.08774. eprint: arXiv:2303.08774. [Online]. Available: https://arxiv.org/abs/2303.08774 (cited on p. 8).

[7] A. Dosovitskiy *et al.*, “An image is worth 16×16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations (ICLR)*, arXiv preprint arXiv:2010.11929, 2021 (cited on p. 8).

[8] A. Radford *et al.*, “Robust speech recognition via large-scale weak supervision,” 2022, OpenAI Whisper paper. [Online]. Available: https://cdn.openai.com/papers/whisper.pdf (cited on p. 8).

[9] H. Zhou *et al.*, “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, arXiv preprint arXiv:2012.07436, vol. 35, 2021, pp. 11 106–11 115. DOI: 10.1609/aaai.v35i12.17325 (cited on p. 8).

[10] J. Lee *et al.*, “Biobert: A pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020, Preprint arXiv:1901.08746 (cited on p. 8).

[11] J. Jumper *et al.*, “Highly accurate protein structure prediction with alphafold,” *Nature*, vol. 596, no. 7873, pp. 583–589, 2021. DOI: 10.1038/s41586-021-03819-2 (cited on p. 8).

[12] F. Yang *et al.*, “Scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data,” *Nature Machine Intelligence*, vol. 4, no. 10, pp. 852–866, 2022. DOI: 10.1038/s42256-022-00534-z. [Online]. Available: https://www.nature.com/articles/s42256-022-00534-z (cited on p. 8).

[13] C. V. Theodoris, J. Xiao, *et al.*, “Transfer learning enables predictions in network biology,” *Nature*, vol. 618, no. 7966, pp. 616–624, 2023. DOI: 10.1038/s41586-023-06139-9 (cited on p. 9).

[14] H. Cui *et al.*, “Scgpt: Toward building a foundation model for single-cell multi-omics using generative ai,” *Nature Methods*, vol. 21, no. 8, pp. 1470–1480, 2024. DOI: 10.1038/s41592-024-02201-0 (cited on p. 9).

[15] H. Gu *et al.*, “Scgnn 2.0: A graph neural network tool for imputation and clustering of single-cell rna-seq data,” *Bioinformatics*, vol. 38, no. 23, pp. 5322–5325, Oct. 2022, ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btac684. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/23/5322/47466040/btac684.pdf. [Online]. Available: https://doi.org/10.1093/bioinformatics/btac684 (cited on p. 9).

[16] J. Hu *et al.*, “Spagcn: Integrating gene expression, spatial location and histology to identify spatial domains of tissues at single-cell resolution,” *Nature Methods*, vol. 18, pp. 1342–1351, 2021. DOI: 10.1038/s41592-021-01255-8 (cited on p. 9).

[17] C. Xu *et al.*, “Deepst: Identifying spatial domains in spatial transcriptomics by deep learning,” *Nucleic Acids Research*, vol. 50, no. 22, e131–e131, Oct. 2022. DOI: 10.1093/nar/gkac901. eprint: https://academic.oup.com/nar/article-pdf/50/22/e131/48488890/gkac901.pdf. [Online]. Available: https://doi.org/10.1093/nar/gkac901 (cited on pp. 9, 34).

[18] Z. Wang *et al.*, *Graph foundation models: A comprehensive survey*, 2025. arXiv: 2505.15116 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2505.15116 (cited on p. 10).

[19] S. Zhao *et al.*, *Stofm: A multi-scale foundation model for spatial transcriptomics*, 2025. arXiv: 2507.11588 [q-bio.GN]. [Online]. Available: https://arxiv.org/abs/2507.11588 (cited on pp. 10, 36, 39).

[20] R. Lopez *et al.*, “Deep generative modeling for single-cell transcriptomics,” *Nature Methods*, vol. 15, pp. 1053–1058, 2018. DOI: 10.1038/s41592-018-0229-2. [Online]. Available: https://www.nature.com/articles/s41592-018-0229-2 (cited on p. 11).

[21] A. Vaswani *et al.*, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762 (cited on pp. 14, 15).

[22] H. Wu *et al.*, “Flowformer: Linearizing transformers with conservation flows,” in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri *et al.*, Eds., ser. Proceedings of Machine Learning Research, vol. 162, PMLR, 17–23 Jul 2022, pp. 24 226–24 242. [Online]. Available: https://proceedings.mlr.press/v162/wu22m.html (cited on pp. 16, 17).

[23] P. Khosla *et al.*, *Supervised contrastive learning*, 2021. arXiv: 2004.11362 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2004.11362 (cited on p. 23).

[24] Y. Dong *et al.*, “Transcriptome analysis of archived tumor tissues by visium, geomx dsp, and chromium methods reveals inter- and intra-patient heterogeneity,” *bioRxiv*, 2024. DOI: 10.1101/2024.11.01.621259. eprint: https://www.biorxiv.org/content/early/2024/11/03/2024.11.01.621259.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2024/11/03/2024.11.01.621259 (cited on p. 26).

[25] H.-O. Lee *et al.*, “Lineage-dependent gene expression programs influence the immune landscape of colorectal cancer,” *Nature Genetics*, vol. 52, no. 6, pp. 594–603, 2020. DOI: 10.1038/s41588-020-0636-z (cited on p. 26).

[26] P. A. Reyfman *et al.*, “Single-cell transcriptomic analysis of human lung provides insights into the pathobiology of pulmonary fibrosis,” *American Journal of Respiratory and Critical Care Medicine*, vol. 199, no. 12, pp. 1517–1536, 2019, PMID: 30554520. DOI: 10.1164/rccm.201712-2410OC. eprint: https://doi.org/10.1164/rccm.201712-2410OC. [Online]. Available: https://doi.org/10.1164/rccm.201712-2410OC (cited on p. 26).

[27] L. Ma *et al.*, “Single-cell atlas of tumor clonal evolution in liver cancer,” *bioRxiv*, 2020. DOI: 10.1101/2020.08.18.254748. eprint: https://www.biorxiv.org/content/early/2020/08/19/2020.08.18.254748.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2020/08/19/2020.08.18.254748 (cited on p. 26).

[28] N. Kim *et al.*, “Single-cell rna sequencing demonstrates the molecular and cellular reprogramming of metastatic lung adenocarcinoma,” *Nature Communications*, vol. 11, no. 1, Article 2285, 2020. DOI: 10.1038/s41467-020-16164-1 (cited on p. 26).

[29] Y. Li *et al.*, “Single-cell transcriptome analysis reveals dynamic cell populations and differential gene expression patterns in control and aneurysmal human aortic tissue,” *Circulation*, vol. 142, no. 14, pp. 1374–1388, 2020. DOI: 10.1161/CIRCULATIONAHA.120.046528. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCULATIONAHA.120.046528. [Online]. Available: https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.120.046528 (cited on p. 26).

[30] T. M. Consortium, “Single-cell transcriptomics of 20 mouse organs creates a tabula muris,” *Nature*, vol. 562, no. 7727, pp. 367–372, 2018. DOI: 10.1038/s41586-018-0590-4 (cited on p. 26).

[31] R. Chen *et al.*, “Single-cell rna-seq reveals hypothalamic cell diversity,” *Cell Reports*, vol. 18, no. 13, pp. 3227–3241, 2017, ISSN: 2211-1247. DOI: https://doi.org/10.1016/j.celrep.2017.03.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211124717303212 (cited on p. 26).

[32] K. R. Maynard *et al.*, “Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex,” *Nature Neuroscience*, vol. 24, no. 3, pp. 425–436, 2021. DOI: 10.1038/s41593-020-00787-0 (cited on p. 26).

[33] J. R. Moffitt *et al.*, “Molecular, spatial, and functional single-cell profiling of the hypothalamic preoptic region,” *Science*, vol. 362, no. 6416, eaau5324, 2018. DOI: 10.1126/science.aau5324. eprint: https://www.science.org/doi/pdf/10.1126/science.aau5324. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aau5324 (cited on p. 26).

[34] K. Z. Kedzierska *et al.*, “Assessing the limits of zero-shot foundation models in single-cell biology,” *bioRxiv*, 2023. DOI: 10.1101/2023.10.16.561085. eprint: https://www.biorxiv.org/content/early/2023/10/17/2023.10.16.561085.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2023/10/17/2023.10.16.561085 (cited on pp. 36, 40).

[35] J. Devlin *et al.*, “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)*, Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://arxiv.org/abs/1810.04805 (cited on p. 39).

[36] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Association for Computational Linguistics, Jun. 2018, pp. 464–468 (cited on p. 39).

[37] J. Su *et al.*, “Roformer: Enhanced transformer with rotary position embedding,” *CoRR*, vol. abs/2104.0986
2021. [Online]. Available: https://arxiv.org/abs/2104.09864 (cited on p. 39).

[38] H. Liu *et al.*, “Biobatchnet: A dual-encoder framework for robust batch effect correction in imaging mass cytometry,” *bioRxiv*, 2025. DOI: 10.1101/2025.03.15.643447. eprint: https://www.biorxiv.org/content/early/2025/03/17/2025.03.15.643447.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2025/03/17/2025.03.15.643447 (cited on p. 39).

Appendices

A Use of Generative AI

We used ChatGPT to assist in creating code for supervised contrastive learning and to obtain advice on a fine-tuning pipeline for cell clustering. To ensure correctness, we compared the generated code with the original paper and debugged it to confirm that the process aligned with our assumptions. The draft of this report was written by ourselves, but we used ChatGPT to improve clarity and academic tone. All key concepts and references were verified through independent study.

B Statement on Use of CellPLM Equations

In this thesis, several mathematical formulations and derivations are presented in detail. For baseline description, we intentionally adopted the equations and notations of the CellPLM model [Wen et al., 2023][2] without modification. The rationale behind this choice is that the exact reproduction of CellPLM's mathematical form is essential to ensure a precise and faithful understanding of the baseline framework, which serves as the foundation of our research. Therefore, although some equations appear verbatim identical to those in the original CellPLM paper, this does not constitute plagiarism but rather reflects the academic necessity of preserving accuracy in baseline reproduction. All such instances are clearly cited to the original source.