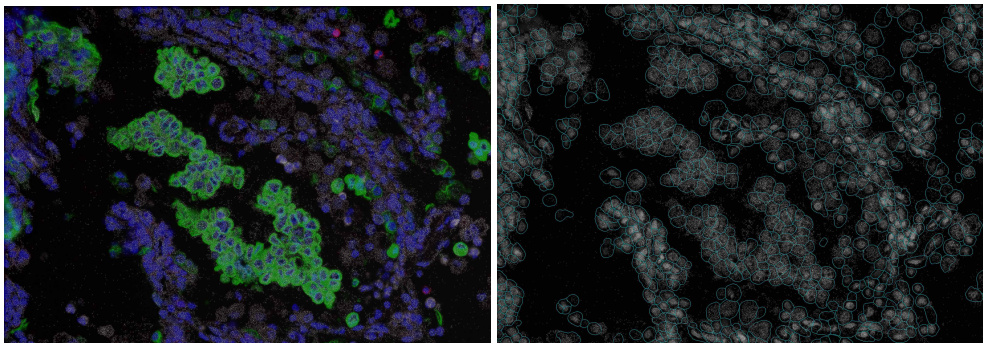
Appendix for CellPLM: Pre-training of Cell Language Model Beyond Single Cells

A SPATIALLY-RESOLVED TRANSCRIPTOMIC DATA

Recently, spatial transcriptomic technologies are developed to spatially resolve transcriptomics profiles Ståhl et al. (2016); Merritt et al. (2020). With spatial transcriptomics data, researchers can learn the spatial context of cells and cell clusters within a tissue Burgess (2019). The major technologies/platforms for spatial transcriptomics are Visium by 10x Ståhl et al. (2016), GeoMx Digital Spatial Profiler (DSP) Merritt et al. (2020) by NanoString and CosMx Spatial Molecular Imager (SMI) by NanoString, MERFISH, Vizgen, Resolve, Rebus, and molecular cartography. 10x Visium does not profile at single-cell resolution, and while GeoMx DSP is capable of single-cell resolution through user-drawn profiling regions, the scalability is limited. The most recent platform, CosMx Spatial Molecular Imager (SMI) He et al. (2022), can profile consistently at single-cell and even sub-cellular resolution. CosMx SMI follows much of the initial protocol as GeoMx DSP, with barcoding and ISH hybridization. However, the SMI instrument performs 16 cycles of automated cyclic readout, and in each cycle, the set of barcodes (readouts) are UV-cleaved and removed. These cycles of hybridization and imaging yield spatially resolved profiling of RNA and protein at single-cell ($\sim 10\mu m$) and subcellular ($\sim 1\mu m$) resolution. In this work, we use two published and one unpublished dataset produced by the CosMx platform. In order to obtain the cellular level gene expression, CellPose Stringer et al. (2021) software is applied to conduct cell segmentation.

To give a concrete example, we provide a sample field-of-view (FOV) in Fig. 4. Pre-selected types of RNA molecules are captured by the molecular imager, which are denoted as white dots in the figures. Colors in the first sub-figure indicate the protein molecules that are stained. These proteins contribute to the cell segmentation process, which results in the second sub-figure. The final output from the pipeline consists of the position of each cell and a cell-by-gene count matrix, which is produced by counting the number of RNA molecules within each cell. The difference between scRNA-seq and SRT data is further demonstrated in Fig. 5.



(a) Visualization of molecular image. (b) Visualization of cell segmentation.

Figure 4: (a) A sample image of protein and RNA molecules. (b) A sample image of segmented cells.

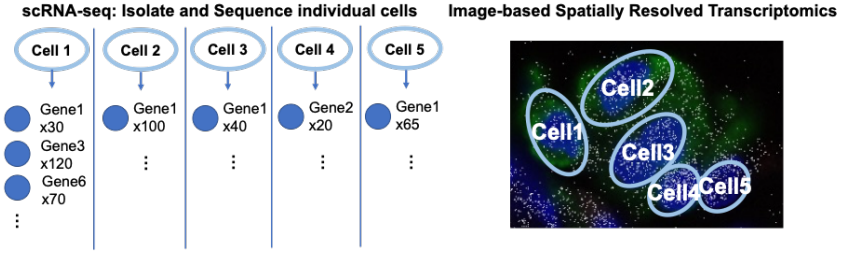


Figure 5: An illustration of the difference between scRNA-seq and SRT data.

B 2D SINUSOID POSITIONAL ENCODINGS

Since 2D sinusoidal PE achieves a competitive performance and has a lower complexity on SRT data Wen et al. (2023), in our transformer encoer, we generate a sinusoidal PE for cells in SRT data, formulated as:

$$\begin{aligned}
&\mathrm{PE}_{(x,y,2i)} = \sin\left(x/10000^{4i/d}\right), \mathrm{PE}_{(x,y,2i+1)} = \cos\left(x/10000^{4i/d}\right), \\
&\mathrm{PE}_{(x,y,2j+d/2)} = \sin\left(y/10000^{4j/d}\right), \mathrm{PE}_{(x,y,2j+1+d/2)} = \cos\left(y/10000^{4j/d}\right),
\end{aligned} \tag{8}$$

where d is the total dimension of positional encoding, $i, j \in [0, d/4)$ specify a specific feature dimension. Let $\tilde{\mathbf{C}} \in \mathcal{R}^{N\times 2}$ be a normalized coordinate matrix, where we normalize and truncate coordinates in $\mathbf{C}$ to integers ranging in $[0, 100)$. x, y then refer to the spatial coordinates from $\tilde{\mathbf{C}}$, e.g., $x = \tilde{\mathbf{C}}_{t,0}$ and $y = \tilde{\mathbf{C}}_{t,1}$ for cell t. In this way, we generate a PE matrix $\mathbf{P} \in \mathcal{R}^{N\times d}$ for every cell in SRT data, where $\mathbf{P}_i$ is the PE vector for cell i. Meanwhile, for scRNA-seq data, a randomly initialized d-dimensional vector p' is shared among all cells, which also results in a placeholder PE matrix $\mathbf{P}$.

C BROADER IMPACT

Our method lies in an emerging and important application area, single-cell analysis. Especially, we leverage a novel type of single-cell data, Spatially Resolved Transcriptomics (SRT). SRT is a rapidly developing technology that allows scientists to map the gene expression of individual cells in their tissue environment. It combines traditional imaging techniques with transcriptome analysis to provide a spatially resolved, high-resolution view of gene expression in complex tissues. Essentially, single-cell technologies and SRT allow researchers to see where specific genes are being expressed within a tissue sample, which can help them better understand cellular interactions and the function of specific genes in complex biological systems.

We evaluate our method on various downstream tasks and the empirical results demonstrate the practical value of our method. Specifically, scRNA-seq Denoising improves the data quality of scRNA-seq data, which often suffer from technical artifacts and dropout events Svensson et al. (2017); Qiu (2020), as well as significant batch effects between sequencing platforms and experiments Tran et al. (2020); Argelaguet et al. (2021). SRT imputation helps to obtain more precise cell state profiles for SRT data, while also resulting in more accurate integration and clustering between SRT data and scRNA-seq data. Perturbation prediction has great clinical value to aid in drug design and disease mechanism research.

While our work offers a significant contribution to the field of single-cell analysis, there are potential negative societal impacts that are important to consider: one of the primary potential negative societal impacts is privacy and data security. Single-cell analysis involves working with sensitive genetic information which, if mishandled, could lead to breaches in privacy and the misuse of personal data. Another potential negative impact is over-reliance on automated analysis. The complexity of single-cell data requires careful interpretation, and the risk of false-positive or false-negative results may be elevated due to computational errors or algorithmic biases. It is crucial to remember that these tools should serve as aids to human understanding and decision-making rather than replacements.

As single-cell technologies continue to evolve, it is critical that we continue to consider and address these broader societal impacts. Moving forward, it is crucial that our work is coupled with ongoing discussions on best practices in data management, privacy protection, and equitable access to technology. This includes strengthening collaborations with ethicists, policymakers, and regulatory bodies to navigate these complex issues.

D DENOISING VARIATIONAL LOWER BOUND FOR MASKED LANGUAGE MODELING

One of the highlights of *CellPLM* is the design of probabilistic latent space. Prior studies have employed variational autoencoders for single-cell analysis, which typically assumes an isotropic Gaussian distribution as the prior distribution of the latent space (Lopez et al., 2018; Xu et al., 2021).

While this approach can effectively remove batch effects, it may also result in a loss of information regarding the underlying biological structure of cell groups. To address this limitation, *CellPLM* incorporates the concept of Gaussian mixture variational encoder (Dilokthanakul et al., 2016; Yang et al., 2019; Xu et al., 2023), which utilizes a mixture of Gaussians to capture the information of distinct functional groups of cells. Formally, for $i \in \{1, \ldots, N\}$, the generative model of cell i can be formulated as:

$$\begin{aligned} p(\mathbf{y}_i; \boldsymbol{\pi}) &= \text{Multinomial}(\boldsymbol{\pi}), \\ p\left(\mathbf{z}_i \mid \mathbf{y}_i\right) &= \prod_{i=1}^{L} \mathcal{N}\left(\boldsymbol{\mu}_{y_{i,l}}, \operatorname{diag}\left(\boldsymbol{\sigma}_{y_{i,l}}^2\right)\right), \\ p_{\theta_{dec}}\left(\mathbf{x}_i \mid \mathbf{z}_i\right) &= \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{z}_i}, \sigma^2 \mathbf{I}\right), \end{aligned} \tag{9}$$

where $\mathbf{y}_i \in \mathcal{R}^L$ represents the one-hot latent cluster variable and $\boldsymbol{\pi}$ is its prior; $y_{i,l}$ denotes the l-th entry of $\mathbf{y}_i$; $\boldsymbol{\mu}_{y_l} \in \mathcal{R}^{d_z}$ and $\boldsymbol{\sigma}_{y_l}^2 \in \mathcal{R}^{d_z \times d_z}$ denote the mean and variance of the l-th Gaussian component, respectively; and $\boldsymbol{\mu}_{z_i} \in \mathcal{R}^k$ and $\sigma^2 \mathbf{I} \in \mathcal{R}^{k \times k}$ denote the posterior mean and variance of expression $\mathbf{x}_i$, respectively. In this work, we assume that σ^2 is a constant and the posterior mean is parameterized by $\boldsymbol{\mu}_{z_i} = f_{dec}(\mathbf{z}_i; \theta_{dec})$.

To estimate the posterior of $\mathbf{z}_i$ and $\mathbf{y}_i$, we parameterize the inference process with neural networks. Specifically, we assume that the cluster variables $\mathbf{y}$ are independent of the expression $\mathbf{x}_i$ condition on latent variables $\mathbf{z}_i$. The inference model can be formulated as:

$$\begin{aligned} q_{\eta_\mu, \eta_\sigma}(\mathbf{z}_i \mid \mathbf{x}_i) &= \mathcal{N}\left(\hat{\boldsymbol{\mu}}_i, \operatorname{diag}\left(\hat{\boldsymbol{\sigma}}_i^2\right)\right), \\ q_{\eta_\pi}(\mathbf{y}_i \mid \mathbf{z}_i) &= \text{Multinomial}(\hat{\boldsymbol{\pi}}_i), \end{aligned} \tag{10}$$

where the estimations are given by

$$\begin{aligned} \mathbf{h}_i &= f_{enc}(\mathbf{x}_i; \eta_{enc}), \\ \hat{\boldsymbol{\mu}}_i &= f_\mu\left(\mathbf{h}_i; \eta_\mu\right), \\ \log\left(\hat{\boldsymbol{\sigma}}_i^2\right) &= f_\sigma\left(\mathbf{h}_i; \eta_\sigma\right), \\ \hat{\boldsymbol{\pi}}_i &= f_\pi\left(\mathbf{z}_i; \eta_\pi\right). \end{aligned} \tag{11}$$

Here $f_{enc}(\cdot; \eta_{enc})$ represents the transformer encoder, $f_\mu(\cdot; \eta_\mu)$, $f_\sigma(\cdot; \eta_\sigma)$ and $f_\pi(\cdot; \eta_\pi)$ are neural networks. A log-evidence lower bound (ELBO) can be derived from this generative model for the optimization purpose (Dilokthanakul et al., 2016). However, as mentioned in Section 3.1, our pre-training framework incorporates a cell language model, where parts of the input gene expression matrix $\mathbf{X}$ are masked. This will result in a modified objective. To formalize the problem, recall that previously we defined the masked set as $\mathcal{M}$. On top of that, we denote $\mathbf{M} \in \mathcal{R}^{N \times k}$ as a mask indicator matrix such that

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

Let $\tilde{\mathbf{X}} \in \mathcal{R}^{N \times k}$ be the masked gene expression matrix given by the element-wise multiplication $\tilde{\mathbf{X}} = \mathbf{M} \odot \mathbf{X}$. The objective of cell language model with Gaussian mixture prior, i.e., a denoising variational lower bound (Im Im et al., 2017), can be formulated as:

$$\begin{aligned} \mathcal{L}_{\text{CellLM}} =& \mathbb{E}_{q(\mathbf{Z}, \mathbf{Y} \mid \tilde{\mathbf{X}})} \mathbb{E}_{p(\tilde{\mathbf{X}} \mid \mathbf{X})} \left[\ln \frac{p_\theta(\mathbf{X}, \mathbf{Z}, \mathbf{Y})}{q_\eta(\mathbf{Z}, \mathbf{Y} \mid \tilde{\mathbf{X}})} \right] \\ =& \underbrace{\mathbb{E}_{q_{\eta_{enc}}(\mathbf{Z} \mid \tilde{\mathbf{X}})} \mathbb{E}_{p(\tilde{\mathbf{X}} \mid \mathbf{X})} \left[\log p_{\theta_{dec}}(\mathbf{X} \mid \mathbf{Z}) \right]}_{\mathcal{L}_{\text{recon}}} - \underbrace{\mathbb{E}_{q_{\eta_\pi}(\mathbf{Y} \mid \mathbf{Z})} \left[\mathrm{KL}\left(q_{\eta_{enc}}(\mathbf{Z} \mid \tilde{\mathbf{X}}) \| p(\mathbf{Z} \mid \mathbf{Y}) \right) \right]}_{\mathcal{L}_{\text{cond}}} \\ & - \underbrace{\mathbb{E}_{q_{\eta_{enc}}(\mathbf{Z} \mid \tilde{\mathbf{X}})} \left[\mathrm{KL}\left(q_{\eta_\pi}(\mathbf{Y} \mid \mathbf{Z}) \| p(\mathbf{Y}) \right) \right]}_{\mathcal{L}_{\text{Y}}}. \end{aligned} \tag{12}$$

E Pre-training Settings

E.1 Hyperparameter Settings

We pre-trained *CellPLM* model with the hyperparameters specified in Table 5.

	CellPLM
encoder hidden dim	1024
encoder layers	4
latent dimension	512
decoder hidden dim	1024
decoder layers	2
model dropout	0.2
cell mask rate	0.75
gene mask rate	0.25
learning rate	2e-4
weight decay	1e-8
num of cluster (for GMM)	16
total parameter	82,402,543

Table 5: Hyperparameters for pretraining *CellPLM* model.

Source	Datasets
HTCA	HTAN_HTAPP, HTAN_Stanford, HTAN_Vanderbilt, HTAN_BU
HCA	cxg_PBMCs, EGAS00001004571_PBMCs, eQTLAutoimmune, covid19autoimmunityPBMCs, VanDerWijst-Human-10x5pv1, cxg_Airways, COMBAT2022, TabulaSapiens, PAN.A01.v01.raw_count.20210429.PFI.embedding, 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 6: List of dataset and data sources. External links will be included in our github repo.

E.2 Datasets for Pre-training

The dataset for pre-training contains 11.4 million cells from scRNA-seq and SRT data. scRNA-seq data consist of 4.7 million cells from human tumor cell atlas (HTCA, `https://humantumoratlas.org/`), 1.4 million cells from human cell atlas (HCA, `https://www.humancellatlas.org/`), and 2.6 million cells from Gene Expression Omnibus (GEO, `https://www.ncbi.nlm.nih.gov/geo/`). All of them are public available data, elucidated in table 6. A more detailed list and external links will be disclosed in our GitHub repository. Note that although our *CellPLM* is capable to handle various input feature sets, when we concatenated these scRNA-seq datasets, we used inner join by default of Anndata package. As a result, all scRNA-seq datasets only contain a $13,500$ common gene set. We will address this issue and increase the size of the gene set in future versions of *CellPLM*.

The SRT datasets we used are publicly available on Nanostring official website: `https://nanostring.com/products/cosmx-spatial-molecular-imager/nsclc-ffpe-dataset/`, where 2.7 million cells and $1,000$ genes are measured. Both scRNA-seq and SRT data are preprocessed with library size normalization and log1p transformation, following the convention in Stuart et al. (2019),

F Additional Experimental Details

In this section, we provide more experimental details about fine-tuning, baselines, and evaluation metrics under each downstream task.

F.1 scRNA-seq Denoising

Downstream Task Datasets. In scRNA-seq denoising task, we evaluate *CellPLM* on two datasets, i.e., PBMC 5K and Jurkat from 10x Genomics lin (a). It is worth noting that during the prepossessing stage, we performed sub-setting on both datasets to ensure that all the genes were included in the gene set of pre-training data. Additionally, any genes with zero counts were removed from the analysis. We list the statistics of them in Table 7.

Table 7: scRNA-seq denoising datasets

	5K PBMC	Jurkat
Number of genes	33,538	32,738
Number of cells	5,247	3,258
Num genes picked	7,197	7,618

Evaluation Metrics. Following the setting of scGNN Wang et al. (2021), scGNN2.0 Gu et al. (2022) and DeepImpute Arisdakessian et al. (2019), we performed synthetic dropout simulation with missing at random (MAR) setting. While scGNN only considered a simple scenario, i.e., randomly flipped 10% of the non-zero entries to zeros, DeepImpute applied cell-wise mask with masking probability given by a multinomial distribution. Specifically, we adapted the setting from DeepImpute with exponential kernel. For cell i that contains at least 5 expressed genes, the probability that one non-zero count $x_{i,j}$ is masked during the training process is given by $\mathrm{Exp}(0, 20)$:

$$p_{i,j} = \frac{1}{20} e^{-\frac{x}{20}},$$
$$q_{i,j} = \frac{p_{i,j}}{\sum_{j=0}^{J_i} p_{i,j}},$$

where J_i is the number of non-zero counts within cell i. We masked 10% of the non-zero counts according to $\{q_{i,j}\}_{j=0}^{J_i}$ and evaluate model performance on the masked entries. We calculate the root mean squared error (RMSE) and mean absolute error (MAE) between the predicted values and ground truth.

Baselines (1) DeepImpute Arisdakessian et al. (2019) employed a strategy of dividing genes into subsets and constructing deep neural networks to impute scRNA-seq data. We implemented DeepImpute with default settings in DANCE Ding et al. (2022) package. (2) scGNN2.0 Gu et al. (2022) incorporated a feature autoencoder, a cluster autoencoder and a graph attention autoencoder for simultaneous imputation and clustering. scGNN2.0 is implemented by DANCE package with default settings. (3) GraphSCI Rao et al. (2021) combined autoencoders with graph convolution networks among a gene-gene similarity graph. We accommodated the implementation of GraphSCI in DANCE package. (4) SAVER Huang et al. (2018) leveraged Poisson LASSO regression to model the scRNA-seq counts with Poisson–gamma mixture. We utilized R package SAVER to illustrate the performance of it. (5) DCA Eraslan et al. (2019) introduced an autoencoder framework based on zero inflated negative binomial (ZINB) distribution. We applied DCA to aforementioned datasets with its Python pacakge. (6) MAGIC Van Dijk et al. (2018) utilized Markov affinity to capture gene-gene relationship and impute missing gene expression. We adapted its Python package to access the performance of it. (7) scImpute Li & Li (2018) developed a Gamma and Gaussian mixture model to identify dropout values. We revealed the performance of scImpute with its R pacakge.

Fine-tuning. Since denoising task requires model to recover the gene expression matrix, we can directly get the zero shot performance of *CellPLM* by specifying the gene set of target dataset. Additionally, we fine-tuned *CellPLM* by replacing the pre-trained decoder with a MLP head and initializing encoder with pre-trained weights. Additionally, for methods require model selection on validation set, we performed another 10% simulation dropout and treat masked entries as validation set. The fine-tuned *CellPLM* was trained on MSE reconstruction loss, while the best model was selected by evaluating MSE on validation set.

F.2 Spatial Tanscriptomic Imputation

Downstream Task Datasets. To evaluate spatial tanscriptomic imputation models at single-cell resolution, we collected two samples from MERSCOPE FFPE Human Immuno-oncology Data lin

(b). Specifically, we chose "Lung cancer 2" and "Liver cancer 2" as our samples, and subsequently referred to them as "Lung2" and "Liver2" respectively. The Lung2 and Liver2 datasets were subsetted to align with the gene set of the pre-training data. Additionally, we removed the fields of view (FOVs) that contained fewer than 100 cells and retained only the first 100 FOVs from both datasets. Note that all baselines require reference scRNA-seq datasets to impute the unseen genes of SRT data, we collected GSE131907 Kim et al. (2020) and GSE151530 Ma et al. (2021) for lung cancer and liver cancer, respectively. The statistics of all datasets are illustrated in Table 8.

Table 8: Spatial tanscriptomic imputation datasets.

	Lung2	Liver2	GSE131907	GSE151530
Number of genes	500	500	29,634	18,667
Number of cells	836,739	598,141	208,506	56,721
Num genes picked	462	446	All	ALL
Num cells picked	40,114	20,629	All	All

Evaluation Metrics. Following the evaluation pipeline proposed by Avşar et al. Avşar & Pir (2023), we selected target genes of SRT data with stratified sampling according to gene sparsity. Specifically, we grouped genes into four categories: low sparse, moderate sparse, high sparse, and very-high sparse. Empirically, the boundaries were defined as $[x < 75, 75 \leq x < 90, 90 \leq x < 95, 95 \leq x]$ to approximate the Gaussian mean and standard deviation slices. Subsequently, we randomly selected 25 genes from each sparsity group and remove them from training data. After training the models, we calculate the evaluation metrics on the target genes. Namely, we compute the root mean squared error (RMSE), Pearson's correlation coefficient (PCC) and cosine similarity (Cosine) between the ground truth values and the corresponding imputed values in a gene-wise approach.

Baselines. (1) SpaGE Abdelaal et al. (2020) relied on domain adaptation to map scRNA-seq data onto SRT data and utilized a k-nearest-neighbor (k-NN) graph to predict unseen genes. We implemented SpaGE with default settings on both datasets. (2) stPlus Shengquan et al. (2021) developed an autoencoder framework for learning cell embeddings and imputing SRT genes using a weighted k-NN approach. The performance of stPlus is accessed by its Python package. (3) gimVI Lopez et al. (2019) introduced a variational autoencoder based model with protocol-specific treatments on scRNA-seq data and SRT data. We applied the scvi-tools lin (c) Python package with default settings to evaluate the performance of gimVI. (4) Tangram Biancalani et al. (2021) utilized a deep learning approach to learn the spatial alignment of scRNA-seq data based on a reference SRT dataset with consistent spatial maps. We evaluated Tangram with its Python package.

Fine-tuning. Similar to scRNA-seq denoising, the spatial tanscriptomic imputation task requires the ouput of the model to be the gene expression. Thus, we directly fine-tune *CellPLM* on the pre-trained weights while specifying the input genes and target genes. The last two batches were hold out for validation.

Visualization of attention. One essential multi-cell task is cell-cell communication (CCC) inference, where CCC mainly represents biochemical signaling through ligand-receptor binding across cells (Cang et al., 2023). Our *CellPLM* applies self-attention mechanism on cell level, from which we can study the interaction strength given by cell attention matrix. As a preliminary study, we extract the attention matrix between cells from a random chosen field of view (FOV) in Cosmx Liver dataset. The attention matrix is treated as CCC scores, and we visualize the results following the stream plot setting in Cang et al. (2023). As shown in the Figure 6 in our supplementary PDF, there are some strong trends on the left side and right side of the FOV, suggesting further exploration of specific signaling pathways for the included cells. This case study showcase the potential of our *CellPLM* model in cell-cell communication research. We hope our model can facilitate more insightful biological research in the future.

F.3 Perturbation Prediction

The perturb-seq technology has been established to examine the gene expression response at the single-cell level when subjected to pooled perturbations (Dixit et al., 2016). By comparing the gene expression before and after perturbation, downstream analysis of differential expression (DE) enables the identification of genes that play a crucial role in disease progression. To assess the potential

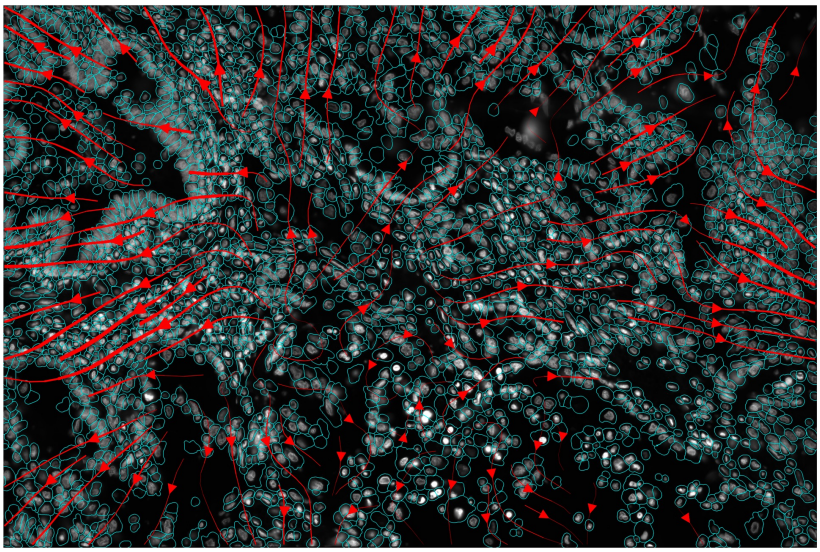


Figure 6: Visualization of attention matrix demonstrate cell-cell communication.

benefits of *CellPLM* in gene-level tasks, we conduct experiments to predict the expression value of genes after perturbation. Following the setting of GEARS (Roohani et al., 2022), we partition the perturbations into training, validation, and test sets, ensuring that none of the test perturbations are encountered during the optimization process.

Two perturbation datasets are employed for evaluation: (1) the Adamson Perturb-Seq dataset (Adamson et al., 2016), consisting of 87 one-gene perturbations; and (2) the Norman Perturb-Seq dataset (Norman et al., 2019), containing 131 two-gene perturbations and 105 one-gene perturbations. To evaluate the performance of perturbation prediction, we employ Root Mean Square Error (RMSE) to measure the degree of similarity between the non-zero ground-truth expression values and corresponding predicted gene expressions. In addition, following previous settings in GEARS (Roohani et al., 2022), we also present the RMSE calculated on the top 20 deferentially-expressed genes.

We compare the performance between *CellPLM* and two baselines, i.e., a recent preprint GEARS method (Roohani et al., 2022), and scGen (Lotfollahi et al., 2019). The results in Figure 7 imply that *CellPLM* achieves the lowest RMSE values across all settings.

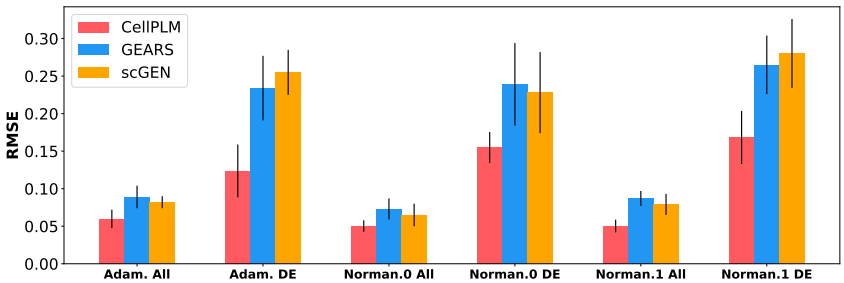


Figure 7: (*Task 3*) The RMSE performance (↓) on Adamson Perturb-Seq and the Norman Perturb-Seq datasets. The Norman Perturb-seq dataset consists of two settings: one-gene perturbations and two-gene perturbations, denoted as Norm.0 and Norm.1, respectively.

Downstream Task Datasets. We included the Adamson Perturb-Seq dataset Adamson et al. (2016) for one-gene perturbations and the Norman Perturb-Seq dataset Norman et al. (2019) for two-gene perturbations. We followed the preprocess pipeline of GEARS Roohani et al. (2022) and both datasets were then gene-wise subsetted to fit in the gene set of pre-training data. The statistics are summaried in Table 9.

Table 9: Perturbation prediction datasets.

	Adamson	Norman
Number of genes	5,060	5,045
Number of cells	68,603	91,205
Num genes picked	3,246	2,353
Num one-gene pert.	87	105
Num two-gene pert.	–	131

Evaluation Metrics. Following the setting of GEARS Roohani et al. (2022), we applied data split such that the testing perturbation are unseen during the training process. Specifically, For Adamson dataset, we randomly hold out 25% of the perturbations for testing and 10% of the perturbations within the training set for validation. For Norman dataset, two settings for two-gene perturbations are implemented for evalutation purpose: $1/2$ unseen and $2/2$ unseen. We excluded all two-gene combinations in which at least one of the individual genes involved in the combination belonged to the unseen set. Finally, we evaluate the performance by calculating the root mean squared error (RMSE) between the predictions and the true values within the testing set.

Baselines. (1) GEARS Roohani et al. (2022) utilized gene co-expression knowledge graph and Gene Ontology-derived knowledge graph to model the influence of perturbations. We followed the recommended parameter settings within its Python package to access the performance. (2) scGen Lotfollahi et al. (2019) built a conditional variational autoencoders and incoporated vector arithmetics to model phenomena response. We implemented scGen with its Python package on both datasets.

Fine-tuning. For one perturbation, we set the input of perturbed genes to be -100 to mimic the gene perturbation action. During the fine-tuning process, we substituted the original batch-aware decoder with a simplified MLP decoder. Additionally, we initialized the remaining components of *CellPLM* with pre-trained weights. The final model was chosen to be the best-performed model on the validation set.

G Cell Type Annotation

Cell type annotation is a crucial step in single-cell analysis as it enables the identification and characterization of distinct cell populations within a tissue or organism. This information is crucial for understanding the functional diversity, developmental trajectories, and disease relevance of different cell types, providing insights into biological processes and facilitating targeted therapeutic approaches.

Downstream Task Datasets. We assess the performance of *CellPLM* on the task of cell type annotation on hPancreas (Chen et al., 2023) and Multiple Sclerosis (MS) (Schirmer et al., 2019), which are suggested by Cui et al. (2023). The hPancreas dataset contains five scRNA-seq datasets of human pancreas cells, divided into reference and query sets with annotations, including 13 cell types and 11 cell types, respectively. The Multiple Sclerosis dataset (M.S.), sourced from EMBL-EBI, includes 9 healthy control and 12 M.S. samples. 3,000 highly variable genes were retained.

Evaluation Metrics. We evaluate cell type annotation performance based on two standard classification metrics, Macro Precision and Macro F1 score.

Baselines. To benchmark the performance of *CellPLM*, we compare it with both pre-trained models including scGPT Cui et al. (2023), scBERT Yang et al. (2022), as well as non-pre-trained SOTA models including ACTINN Ma & Pellegrini (2020), CellTypist Domínguez Conde et al. (2022), SingleCellNet Tan & Cahan (2019), and TOSICA Chen et al. (2023). For baseline methods, we adhere to their provided guidelines and utilize the default parameter setting. The performance metrics reported for scBERT, TOSICA and scGPT in this task are directly obtained from scGPT papers.

Fine-tuning. For *CellPLM* model, we attach a feed forward layer to the pre-trained encoder and latent space and tune the downstream model on the downstream dataset with a standard cross entropy loss.

H Additional Visualization

H.1 Comparison between *CellPLM* and scVI

As a supplement to the zero-shot clustering experiments in Section 4.1, we add an additional comparison with scVI (Lopez et al., 2018) on the same dataset. As shown in Figure 8, *CellPLM* successfully outperforms scVI without any training or fine-tuning, while the latter was trained on this specific dataset.

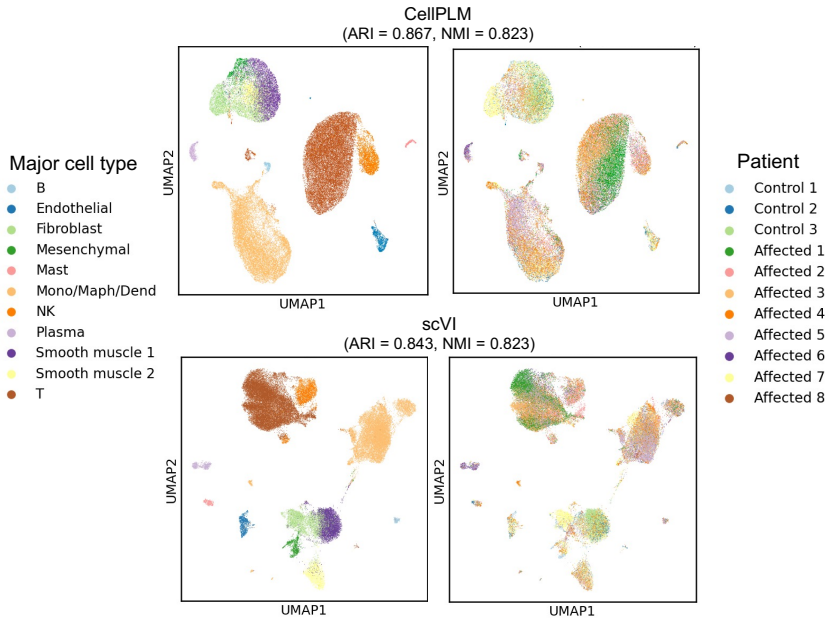


Figure 8: Visualization and comparison between *CellPLM* (zero-shot) and scVI on the clustering task.

H.2 Visualization of Gene Embeddings

In order to examine whether gene interactions can be encoded in *CellPLM*, we present a visualization of pre-trained gene embeddings from the gene expression embedder in Figure 9. From the visualization, the gene embeddings maintain some latent structures. To further verify the effectiveness of the latent structure, we highlight a specific family of genes, HLA genes. There are multiple classes of genes in HLA gene family (Cruz-Tapias & Anaya, 2021). For example, HLA class I genes (e.g., HLA-A, -B, and -C) present endogenous peptides to responding CD8+ T Cells while the class II (e.g., HLA-DR, -DP, and –DQ) process exogenous peptides for presentation to CD4+ helper T Cells. From the UMAP visualization, HLA gene embedding clusters perfectly match the functionality and characteristics of those genes.

I Ablation Study

To further verify the contribution of each component in *CellPLM* model, we add three new ablation studies on two representative tasks to examine the effectiveness of proposed latent distribution and transformer encoder, presented in Table 10. In each setting, we change one component in the model architecture and go through the whole pre-train and fine-tune pipeline to get the downstream performance. Specifically,

1. First, when we replace the proposed mixture of gaussian prior distribution with a gaussian prior distribution (noted as “w/o Mixture of Gaussian", commonly used in previous methods like scVI), the performance significantly drops on all datasets, indicating that an unsuitable prior distribution can greatly hurt the performance. A regular Gaussian distribution cannot accommodate the highly heterogeneous data present in the pre-train dataset, which were collected from different people, organs, and sequencing platforms.

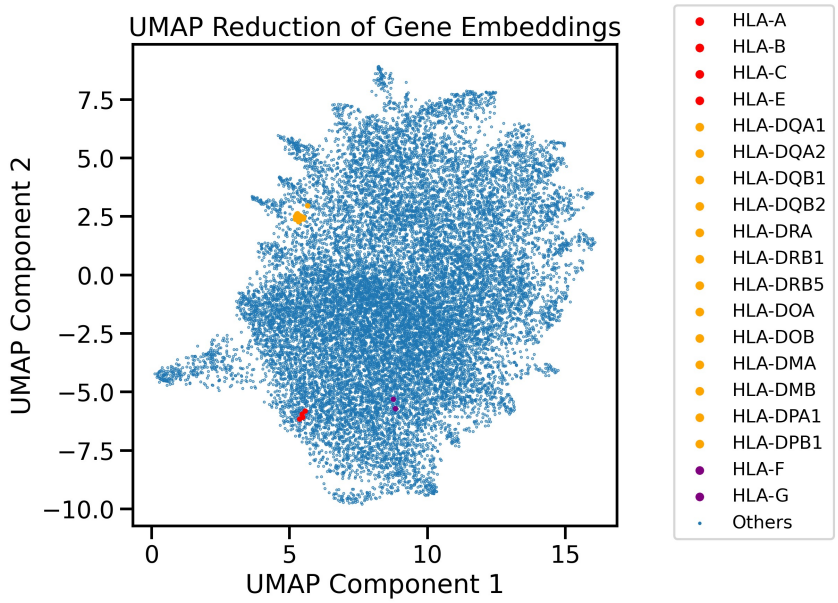


Figure 9: Visualization of gene embeddings in the pre-trained *CellPLM* demonstrate that *CellPLM* successfully captures gene interactions in the initial gene embeddings. For example, HLA Class I genes and HLA Class II perfectly form two clusters in the gene embedding space.

2. Second, we removed the latent distribution in its entirety, noted as “w/o latent distribution", i.e., we converted from a VAE-like probabilistic generative model to a deterministic masked auto-encoder. The performance consistently falls between the original one and the first ablation. On one hand, this supports our motivation of using probabilistic models with Gaussian mixture prior distribution. The latent distribution helps model the uncertainty of the data and address the high noise inherent in transcriptomic data, which results in a robust cell representation. On the other hand, the selection of prior distribution is very important because an improper prior (e.g., regular Gaussian) can be even worse than no latent distribution.
3. Lastly, we replace the transformer encoder with an MLP encoder (noted as “w/o transformer"), keeping the same number of layers and hidden dimension (the total parameters reduce from 85M to 50M). The performance significantly drops on spatial imputation task, while the gap is relatively small on cell-type classification task. This aligns with our intuition, as spatial transcriptomic data provide spatial location information, enabling the model to better identify and utilize the relationships between cells. In contrast, the cell type annotation dataset does not provide spatial location information, which makes the benefits gained from the transformer encoder more limited.

Overall, through a series of ablation studies, we have verified that our *CellPLM* model can capture the relationships between cells via the transformer encoder and enhance the performance of downstream tasks, generating more robust and useful cell representations through appropriate prior distributions.

	Cell-type Classification			
	MS		hPancreas	
	f1	precision	f1	precision
CellPLM	0.766 ± 0.007	0.803 ± 0.008	0.749 ± 0.010	0.753 ± 0.010
w/o Mixture of Gaussian	0.737 ± 0.042	0.766 ± 0.069	0.711 ± 0.025	0.701 ± 0.025
w/o Latent Distribution	0.750 ± 0.024	0.809 ± 0.032	0.733 ± 0.034	0.731 ± 0.033
w/o Transformer Encoder	0.750 ± 0.050	0.794 ± 0.074	0.751 ± 0.010	0.750 ± 0.012
	Spatial Imputation			
	Lung		Liver	
	corr	cosine	corr	cosine
CellPLM	0.318 ± 0.015	0.481 ± 0.011	0.328 ± 0.011	0.481 ± 0.010
w/o Mixture of Gaussian	0.258 ± 0.011	0.449 ± 0.005	0.232 ± 0.013	0.433 ± 0.008
w/o Latent Distribution	0.262 ± 0.011	0.449 ± 0.008	0.246 ± 0.017	0.428 ± 0.012
w/o Transformer Encoder	0.244 ± 0.016	0.443 ± 0.008	0.250 ± 0.032	0.440 ± 0.021

Table 10: Ablation studies on latent distribution and transformer encoder.