# Predicting protein levels from RNA expression data with spatial cellular context

Macrina Lobo mlobo6@gatech.edu
Mehak Bindra mbindra3@gatech.edu
Adrian Kalisz akalisz3@gatech.edu
Esther Shen eshen38@gatech.edu

## Abstract

*Human beings have almost identical DNA in every cell of their body. Despite this, cells form diverse tissues with different spatial structure and function. This is because of the complex and spatially varying expression pattern of genes in different cells of different types and the resulting variation in protein levels. Predicting how DNA, RNA, and protein measurements co-vary in single cells is essential for understanding this complex process. In this paper, we present an approach to predict protein levels from RNA expression levels of spatially profiled cells or spots (groups of cells) in tissue. Using a variational auto-encoder (VAE) framework, our encoder learns latent representations for the mean and variance of RNA and protein expression levels. Our decoder then estimates protein expression level from the RNA expression of the same spots. Unlike other approaches, we show that including the spatial coordinate locations as input to the encoder improves performance. Further, since neighboring cells in a tissue behave similarly, we introduce a "neighborhood loss" that ensures that spots in the same spatially localized neighborhood have similar latent means compared to spots that are far apart. We systematically evaluate the performance of our model on five human and four mouse tissues comprising RNA and protein expression levels along with spatial coordinates ranging from 900 to 2500 spatial spots per sample, with 189 proteins in mice and 273 proteins in humans. We benchmark our approach on a baseline model without spatial coordinates as well as against TotalVI, a state-of-the-art model for predicting protein levels from RNA expression designed in the absence of spatial profiling.*

## 1. Introduction

The human body consists of approximately 37 trillion cells, each exhibiting distinct behaviors and functions. A key challenge in biology is understanding how a single genome can give rise to such a diversity of cellular states. Typically, genetic information flows from DNA to RNA to proteins: DNA must be accessible for RNA transcription, and RNA is then translated into proteins. Regulatory feedback loops, such as proteins binding to DNA to promote or inhibit further RNA production, add complexity to this process. Predicting how DNA, RNA, and protein measurements co-vary in single cells is essential for understanding these intricate regulatory networks. Over the past decade, single-cell genomics has transformed our ability to explore these relationships by enabling the measurement of DNA, RNA, and proteins at the single-cell level. Additionally, advancements in experimental techniques have made it possible to measure multiple omic modalities simultaneously within the same cell. For example, CITE-Seq simultaneously profiles RNA and surface proteins.

In studies involving single-cell data when protein information is unavailable, RNA levels are often used as a suboptimal proxy. To address this limitation, we propose a method for predicting protein levels from RNA expression, which can then be applied in cases where protein data is not experimentally captured. Our approach is designed for the recent spatial CITE-seq assay [12], an extension of traditional CITE-seq that incorporates spatial context alongside RNA and surface protein measurements, providing more detailed insights into tissue organization and cell-cell interactions. The additional challenge in this dataset, is that each measured point comprises a "spot" which is a combination of an unknown number of cells (between 5-95) of the same or different type. Our input training data comprises a spot by gene matrix $X_{cite}$, a spot by protein matrix $Y_{cite}$ measured on the same set of spots, as well as a spot by x-y coordinate matrix $Z_{cite}$ capturing spatial location of the spot in the tissue. For testing, we infer $Y_{cite}$ from $X_{cite}$ and $Z_{cite}$.

## 2. Related Work

Data integration across different experiments, conditions (diseased vs healthy), and modality has been identified as an overarching challenge in single cell omics [8]. The Neurips 2021 challenge [10] made significant strides toward addressing this issue. One of the tasks in the challenge focused on predicting protein expression from RNA expression data obtained at single-cell resolution, using CITE-seq data col-

lected from multiple healthy donors. Several methods have been developed to address this including linear models [5], recurring neural networks (RNNs) [9], variational autoencoders (VAEs) [4, 17], adversarial networks [18], residual networks (ResNet) [20], diffusion [19], and transformer-based foundation models [11].

The top-performing method from NeurIPS 2021 [10] applied truncated SVD followed by kernel ridge regression (KRR) for multi-modal integration. Similarly, scLinear [5] utilizes multivariate regression for similar tasks. Multi-task sciPENN [9] tackles batch effects and missing data with feedforward networks, recurrent cells and censored loss for missing proteins. CrossmodalNet [18] employs Fader networks to disentangle time-specific embeddings from the shared embedding in temporal CITE-seq data using a Discriminator. scMMT [20] uses several ResNet blocks, an embedding block and separate celltype prediction and protein expression blocks with GradNorm gradient loss to balance gradients during backpropagation across the two tasks. scDM [19] uses a diffusion model pre-trained on text data by fine-tuning it for RNA-protein expression pair modeled as a text sample data pair. Gaussian noise is iteratively added to protein expression in forward diffusion. In reverse diffusion, protein expression is predicted from the learned noise with RNA expression serving as a conditional prior. The scTranslator [11] foundation model is pre-trained on paired bulk RNA-protein as well as paired single-cell RNA-protein datasets with optional fine-tuning. A customized transformer architecture comprising a reindexed gene positional encoding (GPE) to encode gene names, fast attention via positive orthogonal random features approach (FAVOR+) and a forward generation of decoder is used.

VAEs [3] have achieved state-of-the-art performance and are particularly well-suited for many single cell omics data integration and cell-type prediction tasks. One of the main advantages of VAEs is their ability to perform unsupervised learning, which is highly beneficial when dealing with high-dimensional, unlabeled omics data. Further, VAEs are capable of integrating data from multiple modalities into a shared latent space. For example, TotalVI [4], the multi-omic extension of scVI [13], models the joint distribution of RNA and protein expression. It incorporates several components, including one-hot encoded batch vectors, a shared cell embedding for both RNA and protein data, an RNA-specific size factor to adjust for sequencing depth, and a protein-specific prior to account for background noise in protein measurements. Similarly, Inclust+ [17] enhances VAE-based frameworks by adding additional embeddings to handle batch encoding and by using masking techniques to filter out missing or irrelevant values. Furthermore, VAEs excel at handling missing data, as they learn a probabilistic model that can infer missing values based on the overall structure of the data. This feature is particularly useful in multi-modal scenarios, where certain modalities—such as RNA or protein measurements—might have missing values for some cells.

Recently, the advent of spatial-citeseq [12], makes it necessary to possible to develop approaches for predicting protein levels from RNA expression for this type of data. Unlike traditional single cell resolution CITE-seq data, spatial-citeseq poses additional technical challenges. First, it captures groups of cells or spots instead of single cells. Second, the data captured is noisier and sparser since the sequencing is performed at lower depth. With the exception of the simplistic and limited scope linear regression based scLinear [5], current methods have not been developed to predict protein levels on this type of data. Other methods working with spatially sequenced data in different contexts incorporate spatial location information under the assumption that spatially proximal cells share similar properties [16]. Current approaches for protein level prediction from RNA expression data do not use spatial coordinates or explicitly model the fact that spatially proximal spots have similar RNA and protein levels.

Here, we address these two deficiencies with an approach specifically designed to predict protein levels from RNA expression on spatial citeseq data. Similar to existing VAE-based methods, like TotalVI, we integrate RNA and protein data into a shared latent space to handle missing data (in our case, all protein data is missing during inference). However, we focus on spot-based spatial omics data, using spatial coordinates in the encoder and introducing a loss term that promotes similar latent embeddings for spatially proximal spots compared to distant spots. With detailed experimentation, we show that using spatial coordinates and spatial neighborhoods improves predictions and exhibits state of the art performance.

## 3. Methodology and Approach

We design and implement a VAE model that predicts missing protein levels $Y_{cite}$(spot by protein matrix) from the noisy spot-based RNA levels $X_{cite}$(spot by gene matrix) derived from the spatial CITE-seq dataset and extend the model to incorporate the X and Y coordinate location matrix $Z_{cite}$. As discussed, we chose a VAE architecture since it has achieved state of the art performance on a variety of single-cell omics tasks such as data integration, batch correction, and cell-type prediction. After preprocessing the RNA and protein data in a manner similar to the best practices in the field [15], we developed a standard baseline VAE with a mean square error reconstruction loss and KL divergence loss. To additionally encode the spatial information, we experimented with several approaches:

- Directly concatenating spatial coordinates to the encoder input

- Encoding spatial locations with PCA on the random walks generated from the spatial nearest neighbor graph.
- Applying a weighted Gaussian kernel on the spatial k-nearest neighbors
- Adding a "neighborhood loss" that ensures that spots in the same spatially localized neighborhood have similar latent space means compared to spots that are far apart.

In this section we describe each of these steps in detail. After detailed experiments (as described subsequently), we chose an architecture as shown in Figure 1.

### 3.1. Data Processing

For each dataset, we normalize the spot by gene matrix by the total counts per cell to a target sum (10,000 in our case) to make expression levels comparable across cells, correcting for variations in sequencing depth. We apply a $log(1+x)$ transformation to the normalized counts to help in stabilizing variance. We perform a similar pre-processing step on the spot by protein matrix. Since we have over 20,000 genes in the spot by gene data, we identify the top $N$ genes that exhibit the most variance across the dataset. This feature selection step is critical for reducing dimensionality by excluding uninteresting genes and focusing analysis on genes that contribute most to biological variability. It has been previously shown that $N$ in the range of 1500-5000 achieves comparable results in single cell omics studies [15] so we chose $N = 4000$.

### 3.2. Base Model Architecture

The base Variational Autoencoder (VAE) model predicts missing protein levels from noisy RNA expression data, with the overall goal to learn a shared latent space that captures the underlying relationships between gene expression (RNA counts) and protein expression levels. Our model consists of a stochastic encoder that takes as input RNA and available protein data to learn a shared latent space from their profiles and a stochastic decoder that generates denoised RNA and protein vectors. The encoder consists of two fully connected layers with hidden dimensions of 1024 and 256, each followed by a non-linear activation function (ReLU) and LayerNorm for stability. It outputs the mean and log-variance of the latent Gaussian distribution. The reparameterization trick is employed to allow gradient-based optimization through the stochastic sampling process. Similar to the encoder, the decoder consists of two fully connected layers with hidden dimensions of 256 and 1024, and it reconstructs the input data from the latent variable. For regularization, we added a dropout layer after our two fully connected encoder layers and a dropout layer between each of our fully connected decoder layers. The loss function combines mean squared error (MSE) for reconstruction and KL divergence to regularize the latent space.

$$\mathcal{L}_{\text{reconstruction}} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{x}_i - x_i\|^2 + \|\hat{y}_i - y_i\|^2$$

$$\mathcal{L}_{D_{KL}} = \frac{1}{2}\sum_{i=1}^{d} \left(\sigma_i^2 + \mu_i^2 - \log \sigma_i^2 - 1\right)$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reconstruction}} + \lambda_{kl}\mathcal{L}_{D_{KL}}$$

where $x_i$ corresponds to the RNA expression values, and $y_i$ corresponds to the protein expression values. $\hat{x}_i$ and $\hat{y}_i$ are the reconstructed RNA and protein values predicted by the model. $\mu_i$ is the mean of the latent variable $z_i$ and $\sigma_i^2$ is the variance of the latent variable $z_i$. $\lambda_{kl}$ controls the trade-off between reconstruction accuracy and generalization. We choose $\lambda_{kl} = 0.0001$, giving a relatively mild penalty on the KL divergence.

Traditional VAEs use cross-entropy instead of MSE for the reconstruction loss. This assumes a Bernoulli likelihood on input data. However, here we normalize and log transform the input with the result that it has a Gaussian distribution and is real valued. Therefore, the reconstruction term in the Evidence Lower Bound (ELBO) reduces to an MSE term which is why we chose it here. Since cosine distances have been widely used in single-cell omics to measure distance between two cells [6], we also experimented with cosine loss $1 - cosine(\hat{x}_i - x_i)$ (results not shown) but achieved poor performance so we reverted to our original choice of MSE reconstruction loss.

During training, both RNA and protein data are provided as input. During testing, protein data is not provided; predictions are based solely on RNA input.

### 3.3. Explicit Spatial Information in Input

To explicitly incorporate spatial information into the model input, we explore two different approaches:

- In this approach, spatial coordinates are directly added as two additional nodes to the encoder input. This allows the model to consider the spatial location of each cell alongside RNA and protein data when learning the latent space.
- In this approach, a K-nearest neighbors ($k = 15$) graph is first constructed from the spatial coordinates, A random walk with restart is then performed on from each node of this graph to generate a vector representation for each cell reflecting its spatial location. Principal Component Analysis (PCA) is applied to reduce the dimensionality of these vectors, and the resulting lower-

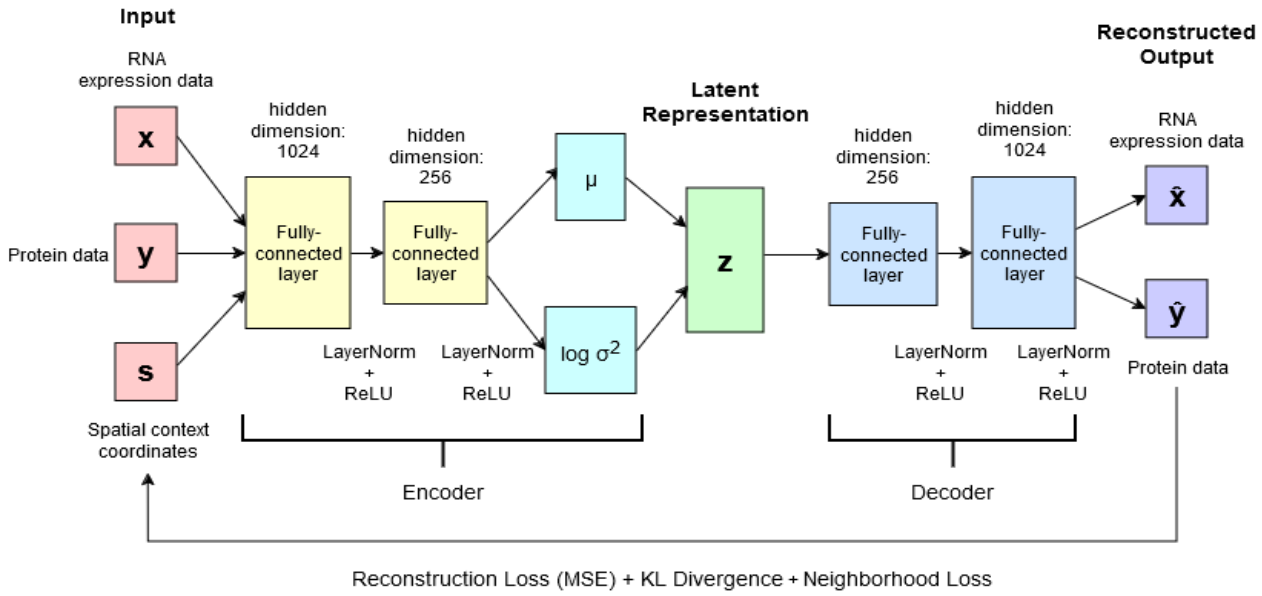


Figure 1. Scheme of network architecture

dimensional coordinates are added as input to the encoder. This method allows for a more nuanced representation of spatial structure in the model's latent space.

### 3.4. Weighted Gaussian Kernel

We implement a weighted Gaussian kernel method on spatial k-nearest neighbors. The weight of each neighboring spot is calculated as follows:

$$\text{weight} = \exp\left(-0.5 \cdot \left(\frac{\text{distance}}{\text{bandwidth}}\right)^2\right)$$

A smaller bandwidth results in more localized influences, while a larger bandwidth allows for a broader range of neighboring spots to significantly influence the weighted average. For each spot, the $k = 15$ nearest neighbors are identified, and the Gaussian kernel is applied to compute and normalize their weights. Finally, the weighted average of the RNA expression values from these neighbors is calculated, providing a spatially influenced estimate for each spot's expression profile.

### 3.5. Neighborhood Loss

We incorporate a neighborhood loss to minimize the distance between the latent distributions of neighboring cells and maximizing it for distant cells, based on the premise that cells within a spatial neighborhood exhibit similar omic profiles, while cells from different neighborhoods have distinct profiles. The distances are computed using the Euclidean norm between the latent points, and the loss is adjusted by a weight factor $\lambda_{nl}$.

$$\mathcal{L}_{\text{neigh}} = \sum_i \left( \sum_{j \in \mathcal{N}_{\text{close}}(i)} \|\mu_i - \mu_j\|_2 - \sum_{k \in \mathcal{N}_{\text{far}}(i)} \|\mu_i - \mu_k\|_2 \right)$$

where $\mu_i$ is the mean of the latent distribution for spot $i$, $\mathcal{N}_{\text{close}}(i)$ is the set of indices of the closest neighbors to the $i$-th cell and $\mathcal{N}_{\text{far}}(i)$ is the set of indices of the furthest neighbors to the $i$-th cell. The complete loss function is then

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reconstruction}} + \lambda_{kl}\mathcal{L}_{D_{KL}} + \lambda_{nl}\mathcal{L}_{\text{neigh}}$$

## 4. Data

We utilized spatial CITE-Seq datasets from Liu et al. (2023) [12], which are publicly available through the Gene Expression Omnibus (GEO) repository (accession number GSE213264). The datasets were released on January 4, 2023, and include samples from both Homo sapiens (humans) and Mus musculus (mice). These datasets were generated using the Illumina HiSeq 4000 platform for high-throughput sequencing. In total, 18 samples are included, providing both RNA and protein data from various tissues, such as the mouse spleen, colon, intestine, kidney, and human tonsil, spleen, thymus, skin, and glioblastoma multiforme (GBM). The datasets capture whole transcriptomes and profile 189 proteins in mice and 273 proteins in humans, spanning between 900 and 2,500 spatial spots per sample. For our pre-training experiment [Table 8], we utilize the Neurips 2021 dataset. Kaggle dataset [1] comprising single-cell multiomics data from Hematopoietic stem cells and

progenitor cells (6 cell types) collected from four healthy human donors.

## 5. Experiments and Results

### 5.1. Evaluation Metrics

We use three metrics to evaluate our model: RMSE [4], Pearson correlation [14], and SSIM [7].
Root mean sqaure error (RMSE) measures the average prediction error, with lower values indicating better accuracy.

$$RMSE = \sqrt{\frac{1}{spots}\sum_{i=1}^{spots}\frac{1}{proteins}\sum_{j=1}^{proteins}(y_{cite,ij} - y_{pred,ij})^2}$$

Pearson Correlation Coefficient (PCC) assesses the linear relationship between predicted and true values, with values closer to 1 showing stronger correlation.

$$r_{cite_i,pred_i} = \frac{\sum_{j=1}^{prot}(y_{cite_i,j} - \bar{y}_{cite_i})(y_{pred_i,j} - \bar{y}_{pred_i})}{\sqrt{\sum_{j=1}^{prot}(y_{cite_i,j} - \bar{y}_{cite_i})^2}\sqrt{\sum_{j=1}^{prot}(y_{pred_i,j} - \bar{y}_{pred_i})^2}}$$

$$PCC = \frac{1}{spots}\sum_{i=1}^{spots} r_{cite_i,pred_i}$$

Structural Similarity Index (SSIM) evaluates structural similarity, with higher values reflecting closer structural alignment between predicted and true data.

$$SSIM(y_{cite}, y_{pred}) = \frac{(2\mu_{y_{cite}}\mu_{y_{pred}} + C_1)(2\sigma_{y_{cite}y_{pred}} + C_2)}{(\mu_{y_{cite}}^2 + \mu_{y_{pred}}^2 + C_1)(\sigma_{y_{cite}}^2 + \sigma_{y_{pred}}^2 + C_2)}$$

where $C_1 = (k_1 L)^2$, $C_2 = (k_2 L)^2$, $k_1 = 0.01$, $k_2 = 0.03$, $L = max(y_{pred}) - min(y_{pred})$

### 5.2. Experimental Framework

For all the results in this section, each dataset was split into $80\%$ training and $20\%$ testing. The model is configured with a latent dimension of 32, and trained using the Adam optimizer with a learning rate of $0.001$ for $100$ epochs. The training process uses a batch size of $64$, unless otherwise stated.

### 5.3. Baseline

As a baseline, we apply the base model to the spot-level spatial CITE-seq data without incorporating spatial information.

| Dataset | Training | | | Testing | | |
|---|---|---|---|---|---|---|
| | **R** | **P** | **S** | **R** | **P** | **S** |
| humanGBM | 0.91 | 0.90 | 0.76 | 0.93 | 0.89 | 0.76 |
| humanskin | 1.29 | 0.72 | 0.57 | 1.35 | 0.71 | 0.53 |
| humanspleen | 1.29 | 0.81 | 0.58 | 1.34 | 0.81 | 0.55 |
| humanthymus | 0.65 | 0.91 | 0.86 | 0.65 | 0.91 | 0.86 |
| humantonsil | 0.64 | 0.91 | 0.85 | 0.65 | 0.91 | 0.85 |
| mousecolon | 1.84 | 0.77 | 0.45 | 1.90 | 0.76 | 0.42 |
| mouseintestine | 1.40 | 0.79 | 0.61 | 1.62 | 0.76 | 0.48 |
| mousekidney | 1.02 | 0.85 | 0.73 | 1.07 | 0.85 | 0.74 |
| mousespleen | 1.23 | 0.79 | 0.64 | 1.32 | 0.77 | 0.60 |
| **average** | **1.14** | **0.83** | **0.67** | **1.19** | **0.82** | **0.65** |

Table 1. Baseline training and test RMSE (R), Average Pearson correlation (P), SSIM (S) (up to 2 decimal places) on various spatial datasets

Our training and testing errors are compatible indicating that our base model does not overfit. Since our PCC and SSIM are not 1.0, we know that we can still improve our model further.

### 5.4. Explicit Spatial Information in Input

As discussed earlier, spatial location of a spot adds additional information since spatially proximal spots are likely to have similar RNA and protein expression. By adding spatial coordinates directly as two additional nodes to the encoder input, there is an improvement in average SSIM, which increases from 0.65 to 0.74, average PCC which increases from 0.82 to 0.86 and a reduction in RMSE from 1.19 to 1.08. [Table 2].

| Dataset | RMSE | Pearson correlation | SSIM |
|---|---|---|---|
| humanGBM | 0.91 | 0.90 | 0.78 |
| humanskin | 1.32 | 0.74 | 0.54 |
| humanspleen | 1.04 | 0.82 | 0.68 |
| humanthymus | 0.66 | 0.91 | 0.86 |
| humantonsil | 0.66 | 0.90 | 0.84 |
| mousecolon | 1.22 | 0.81 | 0.65 |
| mouseintestine | 1.54 | 0.76 | 0.51 |
| mousekidney | 1.01 | 0.85 | 0.74 |
| mousespleen | 1.30 | 0.78 | 0.60 |
| **average** | **1.08** | **0.86** | **0.74** |

Table 2. RMSE, Average Pearson correlation, SSIM (up to 2 decimal places) with Spatial Coordinates Directly Added to the Encoder Input

### 5.5. Spatial neighborhoods with Random Walks and PCA

We believe that in addition to spatial coordinates, spatial neighborhoods can also be leveraged. Next, we con-

struct a K-nearest neighbors graph ($k = 15$) from the spatial coordinates, followed by a random walk with restart to generate cell-specific vector representations reflecting spatial locations. These vectors are then reduced using PCA and incorporated into the encoder input. For random walk with restart we use $\alpha = 0.15$ as the probability of restart and iterate for 100 steps. We show results for when number of PCA components=2 [Table 3] but get similar results for when number of PCA components=16(not shown).

| Dataset | RMSE | Pearson correlation | SSIM |
|---|---|---|---|
| humanGBM | 0.91 | 0.90 | 0.78 |
| humanskin | 1.29 | 0.74 | 0.56 |
| humanspleen | 1.03 | 0.82 | 0.69 |
| humanthymus | 0.65 | 0.91 | 0.86 |
| humantonsil | 0.67 | 0.90 | 0.84 |
| mousecolon | 1.32 | 0.80 | 0.63 |
| mouseintestine | 1.48 | 0.77 | 0.54 |
| mousekidney | 1.01 | 0.85 | 0.74 |
| mousespleen | 1.31 | 0.77 | 0.60 |
| **average** | **1.08** | **0.84** | **0.74** |

Table 3. RMSE, Average Pearson correlation, SSIM (up to 2 decimal places) for the Random Walk with Restart and PCA-based Spatial Encoding

We achieve similar results to when spatial coordinates are directly added to the encoder input, with a similarly higher SSIM compared to baseline. In the next subsection, we explore yet another method of encoding spatial neighborhoods.

### 5.6. Weighted Gaussian Kernel

We apply a weighted Gaussian kernel method to spatial k-nearest neighbors. A bandwidth of 0.05 is selected to concentrate the spatial influence on nearby spots, while minimizing the impact of distant spots and reducing the potential for noise.

The Weighted Gaussian Kernel, while useful in certain settings, does not show significant improvements in this case over the baseline. In fact, it has a slightly higher RMSE than the baseline [Table 4]. This may be because tissue regions with high spatial heterogeneity may be reducing the effectiveness of the spatial kernel. Moreover, the choice of the bandwidth may not be optimal for these particular datasets, leading to over-smoothing. Too much influence from distant, dissimilar cells could be distorting the spatial relationships and potentially leading to dilution of spatially relevant patterns.

| Dataset | RMSE | Pearson correlation | SSIM |
|---|---|---|---|
| humanGBM | 0.94 | 0.90 | 0.76 |
| humanskin | 1.37 | 0.70 | 0.52 |
| humanspleen | 1.43 | 0.79 | 0.51 |
| humanthymus | 0.63 | 0.91 | 0.87 |
| humantonsil | 0.65 | 0.90 | 0.84 |
| mousecolon | 1.87 | 0.76 | 0.43 |
| mouseintestine | 1.58 | 0.76 | 0.50 |
| mousekidney | 1.00 | 0.85 | 0.74 |
| mousespleen | 1.31 | 0.77 | 0.60 |
| **average** | **1.20** | **0.84** | **0.64** |

Table 4. RMSE, Average Pearson correlation, SSIM (up to 2 decimal places) for Weighted Gausian Kernel

### 5.7. Neighborhood Loss

Instead of explicitly modeling neighborhoods, we decided to maximize similarity within a neighborhood compared to outside of the neighborhood. This should overcome the excessive smoothening issue from the previous subsection. We apply the loss function

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reconstruction}} + \lambda_{kl}\mathcal{L}_{D_{KL}} + \lambda_{nl}\mathcal{L}_{\text{neigh}}$$

with $\lambda_{nl} = 1$.

| Dataset | RMSE | Pearson correlation | SSIM |
|---|---|---|---|
| humanGBM | 0.90 | 0.90 | 0.79 |
| humanskin | 1.27 | 0.75 | 0.56 |
| humanspleen | 1.02 | 0.82 | 0.70 |
| humanthymus | 0.69 | 0.91 | 0.87 |
| humantonsil | 0.65 | 0.91 | 0.85 |
| mousecolon | 1.12 | 0.82 | 0.70 |
| mouseintestine | 1.24 | 0.80 | 0.62 |
| mousekidney | 1.00 | 0.85 | 0.75 |
| mousespleen | 1.30 | 0.78 | 0.61 |
| **average** | **1.09** | **0.87** | **0.77** |

Table 5. RMSE, Average Pearson correlation, SSIM (up to 2 decimal places) for Neighborhood Loss

We achieve the best SSIM and Pearson's correlation with this method and nearly the best RMSE [Table 5]. Thus, we conclude that for spatial cite-seq data encoding the spatial coordinates, maximizing within neighborhood similarity and minimizing similarity across neighborhoods leads to significant improvements in prediction of protein levels from RNA expression data.

### 5.8. Poisson Negative Log Likelihood Loss

In this trial, we utilized the same model, but with the loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reconstruction}} + \lambda_{kl}\mathcal{L}_{D_{KL}} + \lambda_{pl}\mathcal{L}_{\text{poissonNLL}}$$

| Dataset | RMSE | Pearson correlation | SSIM |
|---|---|---|---|
| humanGBM | 0.90 | 0.90 | 0.79 |
| humanskin | 1.27 | 0.75 | 0.58 |
| humanspleen | 1.02 | 0.83 | 0.71 |
| humanthymus | 0.62 | 0.91 | 0.87 |
| humantonsil | 0.65 | 0.91 | 0.85 |
| mousecolon | 1.10 | 0.82 | 0.72 |
| mouseintestine | 1.17 | 0.81 | 0.68 |
| mousekidney | 1.00 | 0.85 | 0.75 |
| mousespleen | 1.29 | 0.78 | 0.62 |
| **average** | **1.00** | **0.84** | **0.73** |

Table 6. RMSE, Average Pearson correlation, SSIM (up to 2 decimal places) for Poisson Negative Log Likelihood

While this approach did not achieve either the best Pearson correlation nor the best SSIM it managed to score the best on the RMSE [Table 6].

### 5.9. Performance of state-of-the-art TotalVI [4] on noisy spatial data

TotalVI is not designed for spatial cite-seq datasets but rather predicts protein levels from RNA expression in non-spatial citeseq data. As we have discussed, this is an easier problem. Here, we wanted to see if the same model can perform well on the more noisy spatial dataset. We apply the off-the-shelf version of TotalVI to the spot-level spatial CITE-seq data (without incorporating spatial information) for benchmarking with our approach.

| Dataset | Pearson correlation | SSIM |
|---|---|---|
| humanGBM | 0.94 | 0.77 |
| humanskin | 0.87 | 0.83 |
| humanspleen | 0.90 | 0.62 |
| humanthymus | 0.95 | 0.75 |
| humantonsil | 0.94 | 0.80 |
| mousecolon | 0.89 | 0.72 |
| mouseintestine | 0.91 | 0.56 |
| mousekidney | 0.95 | 0.64 |
| mousespleen | 0.93 | 0.77 |
| **average** | **0.92** | **0.71** |

Table 7. Average Pearson correlation, SSIM (upto 2 decimal places) with TotalVI on various spatial test datasets

Spatial transcriptomics datasets are often noisy due to various factors, including biological variability, technical noise, and spatial heterogeneity. Despite the higher noise of spatial transcriptomic data, the vanilla TotalVI model originally developed for non spatial citeseq data showed good performance on several spot-based spatial transcriptomic datasets [Table 2]. The model outperforms our best model [Table 5] on Pearson correlation (0.92 versus 0.87) but is inferior to our best model on the SSIM metric (0.71 versus 0.77). A key difference with TotalVI is that instead of normalizing and log transforming the data prior to training and then using MSE reconstruction loss, they use the library sizes of the input data as prior input weights which could be the reason for the improved performance.

### 5.10. Pre-training on non-spatial CITE-Seq and Fine-tuning on spatial CITE-Seq

Since TotalVI performed well on spatial cite-seq data, we wanted to see if its performance could be further improved by pre-training on a less noisy non-spatial citeseq dataset. To do this, we used the non-spatial CITE-seq data(Neurips 2021 dataset), followed by fine-tuning on a training subset of spatial CITE-Seq dataset (without actually incorporating any spatial information). We test on the remaining spatial CITE-Seq data. The goal is to evaluate whether pre-training on the large, dense and less noisy non-spatial data dataset followed by fine-tuning on the sparse and noisy spatial data improves performance compared to vanilla TotalVI and if this is something we should be doing with our model as well. We encode the spatial dataset as a separate batch by adding a one-hot input node to the existing set of batch nodes (which currently encodes donor information) in the encoder. This allows the model to learn distinct weights for the spatial data by incorporating this newly added node into the encoding process. All the other encoder weights are fixed.

| Dataset | Pearson correlation | SSIM |
|---|---|---|
| humanGBM | 0.70 | 0.18 |
| humanskin | 0.74 | 0.43 |
| humanspleen | 0.82 | 0.36 |
| humanthymus | 0.78 | 0.14 |
| humantonsil | 0.68 | 0.09 |
| mousecolon | 0.68 | 0.44 |
| mouseintestine | 0.45 | 0.16 |
| mousekidney | 0.80 | 0.41 |
| mousespleen | 0.62 | 0.29 |
| **average** | **0.70** | **0.27** |

Table 8. Average Pearson correlation, SSIM (upto 2 decimal places) evaluated on various spatial test datasets with TotalVI pre-trained on non-spatial Citeseq and finetuned on spatial citeseq data

Simply pre-training plus fine-tuning resulted in much worse performance [Table 8] compared to vanilla TotalVI [Table 7]. This is likely because the non-spatial citeseq data is from a different tissue (Blood) compared to the spatial cite-seq data and thus has a very different distribution. Non-spatial citeseq data is currently not available for similar tissues so we did not pursue a pre-training / fine-tuning strategy in the context of our method.

### 5.11. Incorporating spatial information in the TotalVI model

We also tried to incorporate spatial information with a Gaussian kernel in a manner similar to that applied for our model. Similar to the approach in our method, we choose a bandwidth of $0.05$ to ensure that the spatial influence is focused on nearby spots, without introducing too much noise from distant spots. For each spot, we identify $k = 15$ nearest neighbors and apply the Gaussian kernel to compute and normalize their weights. Finally, we compute the weighted average of the RNA expression values from the neighbors.

| Dataset | Pearson correlation | SSIM |
|---|---|---|
| humanGBM | 0.92 | 0.52 |
| humanskin | 0.87 | 0.35 |
| humanspleen | 0.94 | 0.42 |
| humanthymus | 0.85 | 0.08 |
| humantonsil | 0.91 | 0.33 |
| mousecolon | 0.88 | 0.14 |
| mouseintestine | 0.87 | 0.15 |
| mousekidney | 0.94 | 0.39 |
| mousespleen | 0.90 | 0.23 |
| **average** | **0.89** | **0.29** |

Table 9. Average Pearson correlation, SSIM (upto 2 decimal places) on various spatial test datasets with our Gaussian kernel implementation to encode spatial coordinates

Similar to what we observed for our method, a Gaussian kernel does not improve the performance of TotalVI.

### 5.12. Result Summary

Here is a summary of the key experiments and results.

| Method | Ref | Average PCC | Average SSIM |
|---|---|---|---|
| Baseline VAE | Table 1 | 0.82 | 0.65 |
| Improved VAE | Table 5 | 0.87 | 0.77 |
| TotalVI | Table 7 | 0.92 | 0.71 |

Table 10. Average across all datasets for Pearson correlation, SSIM (upto 2 decimal places) for the key results

In all experiments on our method as well as with our benchmarking experiments, we observed consistent performance between the training and test sets, suggesting that the model generalizes well and is not overfitting. We evaluated our VAE model performance with various hyperparameters, including the learning rate, number of training epochs, the latent dimensionality, the hidden layer dimensions, the KL and NL regularization terms ($\lambda_{kl}, \lambda_{nl}$), value of $k$ for weighted Gaussian and neighborhood loss, the number of genes with top variance to use, the number of iterations for the random walk and the restart probability ($\alpha$). We experimented with different values for each of these hyperparameters, but the model showed strong robustness across a wide range of settings. For each experiment, we observed a general reduction and plateauing of the loss function during training confirming convergence to a minima.

Our VAE model was implemented from scratch in PyTorch and run using a GPU either on Google Colab (Adrian/Esther) or on our lab server (Macrina/Mehak). Comparisons with TotalVI [4] were done using the layers and modules from the scvi-tools package [2] with custom code for the modifications described.

## 6. Conclusion

In this paper, we present an approach to infer protein levels from RNA expression data. Unlike other tools, our method is designed for the noisier, sparser spatial-citeseq experimental assay which captures RNA expression, protein levels, and spatial location for each groups of cells or "spot". We make use of a VAE architecture which has previously achieved state of the art performance on a variety of omics tasks such as data integration, batch correction, and cell type prediction. In addition to RNA and protein, we experiment with several methods to incorporate spatial coordinate information. After pre-processing, our input data is a Gaussian distribution so our baseline VAE model uses MSE and KL divergence loss terms. We extend our model by concatenating our encoder input with nodes for spatial coordinates and demonstrate superior performance. We further improve performance by defining a new neighborhood loss term which ensures that the latent means of spatially proximal spots are closer than that of spatially distant spots. Finally, we compare the performance of our model to a state of the art method TotalVI which does not utilize spatial information. Results show comparable performance of our method with TotalVI which we attribute to the fact that TotalVI learns normalization parameters by modeling them as priors instead of applying standard pre-processing like we do.

We believe our model has great potential. For example, the neighborhood loss can be further improved by capturing the variance of the distribution along with the mean. Further, a more detailed exploration of the input distribution might enable us to learn preprocessing parameters like TotalVI which could further improve performance.

## 7. Team Contributions

## References

[1] Andrew Benz Peter Holderrieth Jonathan Bloom Christopher Lance Ashley Chow Ryan Holbrook Daniel Burkhardt, Malte Luecken. Open problems - multimodal single-cell integration, 2022. 4

| Name | Contributions Summary |
|---|---|
| Macrina Lobo | project idea, datasets, architecture, loss function variations, totalVI experiments, report |
| Mehak Bindra | architecture, loss function design, VAE implementation, loss function variations, totalVI experiments, hyperparameter tuning, report |
| Adrian Kalisz | exploration of TotalVI, loss function variations, report |
| Esther Shen | hyperparameter tuning, architecture diagram, exploring sciPENN [9] model, report |


[2] Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, et al. A python library for probabilistic analysis of single-cell omics data. *Nature biotechnology*, 40(2):163–166, 2022. 8

[3] Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, Yining Liu, Jules Samaran, Gabriel Misrachi, Achille Nazaret, Oscar Clivio, Chenling Xu, Tal Ashuach, Mariano Gabitto, Mohammad Lotfollahi, Valentine Svensson, Eduardo da Veiga Beltrame, Vitalii Kleshchevnikov, Carlos Talavera-López, Lior Pachter, Fabian J. Theis, Aaron Streets, Michael I. Jordan, Jeffrey Regier, and Nir Yosef. A python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology*, Feb 2022. 2

[4] Adam Gayoso, Zoë Steier, Romain Lopez, Jeffrey Regier, Kristopher L Nazor, Aaron Streets, and Nir Yosef. Joint probabilistic modeling of single-cell multi-omic data with totalvi. *Nature methods*, 18(3):272–282, 2021. 2, 5, 7, 8

[5] Daniel Hanhart, Federico Gossi, Maria Anna Rapsomaniki, Marianna Kruithof-de Julio, and Panagiotis Chouvardas. Sclinear predicts protein abundance at single-cell resolution. *Communications biology*, 7(1):267, 2024. 2

[6] Yuge Ji, Tessa Green, Stefan Peidli, Mojtaba Bahrami, Meiqi Liu, Luke Zappia, Karin Hrovatin, Chris Sander, and Fabian Theis. Optimal distance metrics for single-cell rna-seq populations. *bioRxiv*, pages 2023–12, 2023. 3

[7] Nelson Johansen, Hongru Hu, and Gerald Quon. Projecting rna measurements onto single cell atlases to extract cell type-specific expression profiles using scprojection. *Nature Communications*, 14(1):5192, 2023. 5

[8] David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J McCarthy, Stephanie C Hicks, Mark D Robinson, Catalina A Vallejos, Kieran R Campbell, Niko Beerenwinkel, Ahmed Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21:1–35, 2020. 1

[9] Justin Lakkis, Amelia Schroeder, Kenong Su, Michelle YY Lee, Alexander C Bashore, Muredach P Reilly, and Mingyao Li. A multi-use deep learning method for cite-seq and single-cell rna-seq data integration with cell surface protein prediction and imputation. *Nature machine intelligence*, 4(11):940–952, 2022. 2, 9

[10] Christopher Lance, Malte D Luecken, Daniel B Burkhardt, Robrecht Cannoodt, Pia Rautenstrauch, Anna Laddach, Aidyn Ubingazhibov, Zhi-Jie Cao, Kaiwen Deng, Sumeer Khan, et al. Multimodal single cell data integration challenge: results and lessons learned. *BioRxiv*, pages 2022–04, 2022. 1, 2

[11] Linjing Liu, Wei Li, Ka-Chun Wong, Fan Yang, and Jianhua Yao. A pre-trained large generative model for translating single-cell transcriptome to proteome. *bioRxiv*, pages 2023–07, 2023. 2

[12] Yang Liu, Marcello DiStasio, Graham Su, Hiromitsu Asashima, Archibald Enninful, Xiaoyu Qin, Yanxiang Deng, Jungmin Nam, Fu Gao, Pino Bordignon, et al. High-plex protein and whole transcriptome co-mapping at cellular resolution with spatial cite-seq. *Nature Biotechnology*, 41(10):1405–1409, 2023. 1, 2, 4

[13] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018. 2

[14] Malte D Luecken, Daniel Bernard Burkhardt, Robrecht Cannoodt, Christopher Lance, Aditi Agrawal, Hananeh Aliee, Ann T Chen, Louise Deconinck, Angela M Detweiler, Alejandro A Granados, et al. A sandbox for prediction and integration of dna, rna, and proteins in single cells. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (Round 2)*, 2021. 5

[15] Malte D Luecken and Fabian J Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular systems biology*, 15(6):e8746, 2019. 2, 3

[16] Ying Ma and Xiang Zhou. Spatially informed cell-type deconvolution for spatial transcriptomics. *Nature biotechnology*, 40(9):1349–1359, 2022. 2

[17] Lifei Wang, Rui Nie, Xuexia Miao, Yankai Cai, Anqi Wang, Hanwen Zhang, Jiang Zhang, and Jun Cai. Inclust+: the deep generative framework with mask modules for multimodal data integration, imputation, and cross-modal generation. *BMC bioinformatics*, 25(1):41, 2024. 2

[18] Yongjian Yang, Yu-Te Lin, Guanxun Li, Yan Zhong, Qian Xu, and James J Cai. Interpretable modeling of time-resolved single-cell gene–protein expression with cross-modalnet. *Briefings in Bioinformatics*, 24(6):bbad342, 2023. 2

[19] Hanlei Yu, Yuanjie Zheng, and Xinbo Yang. scdm: A deep generative method for cell surface protein prediction with diffusion model. *Journal of Molecular Biology*, 436(12):168610, 2024. 2

[20] Songqi Zhou, Yang Li, Wenyuan Wu, and Li Li. scmmt: a multi-use deep learning approach for cell annotation, protein

prediction and embedding in single-cell rna-seq data. *Briefings in Bioinformatics*, 25(2):bbad523, 2024. 2