STProtein: predicting spatial protein expression from multi-omics data

Anonymous submission

Abstract

The integration of spatial multi-omics data from single tissues is crucial for advancing biological research. However, a significant data imbalance impedes progress: while spatial transcriptomics data is relatively abundant, spatial proteomics data remains scarce due to technical limitations and high costs. To overcome this challenge we propose STProtein, a novel framework leveraging graph neural networks with multi-task learning strategy. STProtein is designed to accurately predict unknown spatial protein expression using more accessible spatial multi-omics data, such as spatial transcriptomics. We believe that STProtein can effectively addresses the scarcity of spatial proteomics, accelerating the integration of spatial multi-omics and potentially catalyzing transformative breakthroughs in life sciences. This tool enables scientists to accelerate discovery by identifying complex and previously hidden spatial patterns of proteins within tissues, uncovering novel relationships between different marker genes, and exploring the biological "Dark Matter".

Introduction

Recently, spatial transcriptomics has been evolved into spatial multi-omics, enabling the visualization and analysis of multiple omics within a single tissue sample. This significant progress can be mainly attributed to the advancement of spatial multi-omics techniques, including SPOTS (Ben-Chetrit et al. 2023), STARmap PLUS (Zeng et al. 2023), 10x Genomics Xenium s(Janesick et al. 2023), Stero-CITE-seq (Liu et al. 2023), and Stereo-CITE-seq (Liao et al. 2023). These techniques enable the acquisition of multiple complementary perspectives of individual cells with spatial information, offering significant potential for revealing insights into cellular and previously undiscovered tissue properties. In this field, developing an AI-powered toolkit, serving as a precise "telescope", can assist scientists in exploring the research horizons and uncovering scientific discoveries by deciphering the majority of the unannotated data, also called the "Dark Matter" (Han et al. 2024).

In order to fully use spatial multi-omics data for a thorough understanding of the tissue, it is vital to integrate multi-omics data to perform analysis. However, the main challenge is the scarcity of multi-omics data that has greatly hindered comprehensive analysis. Furthermore, the cost of spatial proteomics sequencing is typically between $3,000 and $7,000 more expensive than that of spatial transcriptomics sequencing (Ben-Chetrit et al. 2023) in SPOTS (Ben-Chetrit et al. 2023) sequencing platform. This has given rise to a phenomenon in which the pace of spatial transcriptomics data generation has exceeded that of spatial proteomics, resulting in an imbalance in the volume of data between the two modalities. This imbalance poses a significant barrier to widespread adoption and constraints the advancement of studies in spatial multi-omics.

In this paper, we propose STProtein, a framework for predicting spatial protein expression from spatial multi-omics data. STProtein leverages graph neural networks (GNNs) to effectively model complex spatial relationships, integrating RNA and protein expression with cellular interactions within tissue structures. In addition to its predictive capabilities, STProtein serves as a powerful computational toolkit that can accelerate scientific discovery by enabling the identification of complex, hidden spatial protein patterns within tissues, uncovering previously undetectable relationships between marker genes, and facilitating the exploration of "Dark Matter" within the biological world.

Related Work

Multi-omics Prediction

Multi-omics prediction aims to infer unmeasured molecular features—such as protein expression or chromatin accessibility—from available omics data, typically from transcriptomic data. Existing methods for multi-omics prediction can be divided into three main categories: 1): matrix factorization methods; 2): probabilistic methods and 3): deep learning methods. Matrix factorization methods, like non-negative matrix factorization, break down multi-omics datasets into shared latent factors to create unified representations (Abe and Shimamura 2023). Probabilistic methods, often based on Bayesian statistics theory, use conditional probability distributions to predict one omics modality from another, typically requiring prior biological knowledge (Argelaguet et al. 2020). Compared with two traditional methods mentioned above, deep learning methods have become increasingly popular due to their generality and strong predictive performance across diverse datasets. For example, totalVI (Gayoso et al. 2021), a variational autoencoder-based algorithm, can jointly model single-cell RNA and pro-

tein data. By mapping both modalities into a shared low-dimensional latent space, enabling cross-modality prediction. In addition, scArches (Lotfollahi et al. 2022) uses transfer learning strategy by fine-tuning pre-trained models to adapt to new datasets. For protein expression prediction, scArches can apply a model trained on large-scale multi single-cell omics dataset to make protein expression prediction on a new dataset. Although predicting certain omic expression from multi-omics, these deep learning methods ignore the influence of spatial location on omics features.

Computational Methods for Spatial Resolved Omics

Recently, spatial resolved omics technologies, particularly spatial transcriptomics, have been introduced to combine spatial information with transcriptomic data. This advancement has led to the development of several computational methods for analyzing these complex datasets. For instance, Seurat (Satija et al. 2015) is a widely used tool that facilitates data preprocessing, including normalization and dimensionality reduction using Principal Component Analysis (PCA) (Abdi and Williams 2010) and Uniform Manifold Approximation and Projection (UMAP) (McInnes, Healy, and Melville 2018). Seurat also employs clustering algorithms to identify distinct cell populations based on gene expression patterns. In addition, STAGATE (Dong and Zhang 2022), recognized as one of the top advancements in bioinformatics in China in 2022, uses a graph attention network to capture the local structure and spatial dependencies in transcriptomic context. By focusing on the interrelationships between spatially located cells, STAGATE can enhance the understanding of cellular behavior within their microenvironments. The most recent work, SpatialGlue (Long et al. 2024), an integration algorithm, constructs a K-nearest neighbor (KNN) graph (Kang 2021) to model semantic relationships among different cell types. It integrates spatial information with multi-modal omics through attention weights, offering a comprehensive representation for spatila multi-omics. This integration allows for a more nuanced interpretation of how spatial context influences gene expression and cellular interactions. Despite the progress made by existing methods, there is still a notable gap when it comes to prediction algorithms that can use transcriptomic data to predict proteomic data. Our work aims to address this gap by introducing a new algorithm designed to predict spatial protein expression from transcriptomic profiles, thus improving predictive capabilities in spatial omics. Additionally, spatial transcriptomic data is abundant, while spatial proteomic data is limited and costly to sequence compared to transcriptomic data. This creates a data imbalance issue. Our prediction algorithm can generate new proteomics data based on known transcriptomics data, helping to mitigate this problem.

Method

The design of STProtein framework is inspired by the algorithm BulkTrajBlend in Omicverse (Zeng et al. 2024), which can deconvolve single-cell data from bulk RNA-seq and employ a GNN-based algorithm to identify contiguous cell populations within the generated single-cell dataset. STProtein framework can be divided into three main parts: 1): Construction of feature graph; 2): Graph attention autoencoder block; and 3): Upstream and downstream tasks. The training process of STProtein, including the first two parts, is shown in Figure 1. The workflow of the upstream and downstream tasks within STProtein is shown in Figure 2. In this study, X and X' represent the RNA expression and its reconstruction, respectively, while Z denotes the embedding.

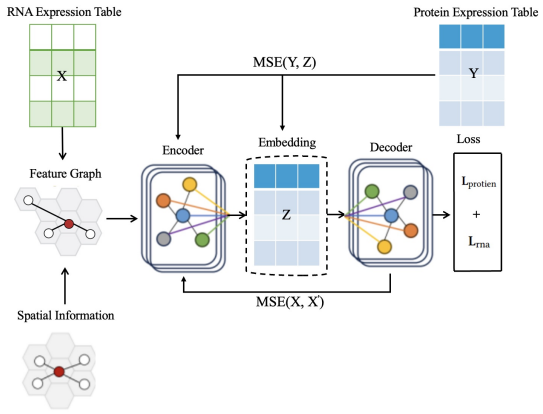

Figure 1: Training Framework of STProtein

Graph neural networks typically operate on graph-structured data, consisting of node features and edge index matrices. Therefore, the first part of STProtein is about construction of feature graph from the original spatial omics data. The feature graph then serves as the input for the second part: graph attention autoencoder block. The direct output also called protein embedding from this block can subsequently be utilized in the upstream task such as protein expression prediction. Finally, after applying clustering to the direct output (protein embedding), it can be used in the downstream task to observe and identify protein spatial domains. The above parts will be introduced in detail in the following sections.

Data Preprocessing

For all datasets, STProtein employs standard data preprocessing steps for transcriptomic and protein data. Specifically, for the transcriptomic data, gene expression counts are log-transformed and normalized by library size using the SCANPY package (Wolf, Angerer, and Theis 2018). The top 4,000 highly variable genes (HVGs) by using Seurat version 3 (Satija et al. 2015) method are selected as inputs for PCA to reduce dimensionality. To maintain a consistent input dimension with the protein data, the first k principal components (where k represents the number of proteins) are retained and used as model inputs. For protein data, protein expression counts are normalized using the Centered Log Ratio (CLR) (Townes et al. 2019). All principal components obtained after PCA dimensionality reduction are utilized as inputs for the model.

Construction of Feature Graph

Typically, the construction of feature graph for spatial omcis data can be divided into two mainstream methods with different assumptions. 1): Spatial Neighbor Feature Graph (Palla et al. 2022): assume the similar spots are spatially adjacent; and 2): KNN Feature Graph (Dann et al. 2022): assume the similar spots may not be spatially adjacent. In our project, we use the second method to construct feature graph with more reasonable and trustworthy assumption based on domain knowledge in biology and previous work (Dong and Zhang 2022).

Spatial Neighbor Feature Graph: With the assumption that spots with similarly cell types or states are spatially adjacent, we can convert spatial information into an undirected feature graph $G^{sn}_{feature} = (V^{sn}, E^{sn})$. V^{sn} denotes the set of N spots. E^{sn} denotes the set of connected edges between spots. We use the adjacency matrix $A^{sn}_{feature} \in R^{N\times N}$ to denote the feature graph $G^{sn}_{feature} = (V^{sn}, E^{sn})$. If the Euclidean distance between spot $j \in V^{sn}$ and spot $i \in V^{sn}$ is less than the specific radius number r, then $A^{sn}_{feature}(i, j) = 1$, otherwise $A^{sn}_{feature}(i, j) = 0$. The default value for r is 2, which is the same as default value in scikit-learn (Pedregosa 2011).

KNN Feature Graph: In complex tissue samples, spots with identical cell types or states may not be spatially adjacent or could be distantly located in spatial context. To capture the relationship of such spots in a latent space, we explicitly model their relationships using a feature graph. Specifically, we apply a k-nearest neighbours (KNN) algorithm to PCA embeddings and construct the feature graph $G^{knn}_{feature} = (V^{knn}, E^{knn})$. V^{knn} denotes the set of N spots. E^{knn} denotes the set of connected edges between spots. For each spot, we select the top k nearest spots as neighbours. The default value for k is 3. And experiment and discussion of k value's impact for STProtein framework can be seen in Section . We use the adjacency matrix $A^{knn}_{feature} \in R^{N\times N}$ to denote the feature graph $G^{knn}_{feature} = (V^{knn}, E^{knn})$. If spot $j \in V^{knn}$ is the neighbour of $i \in V^{knn}$, then $A^{knn}_{feature}(i, j) = 1$, otherwise $A^{knn}_{feature}(i, j) = 0$.

Graph Attention Autoencoder Block

The graph attention autoencoder block can be divied into four main parts: graph attention layer, encoder, decoder and loss function. And graph attention layer is implemented by using PyG (PyTorch Geometric) (Fey and Lenssen 2019). The comparison experiments of other graph convolutional layers can be seen in Appendix . And the details of graph attention autoencoder block can be seen in the following section.

Graph Attention Layer: Our graph attention layer is based on the GATv2 (Brody, Alon, and Yahav 2021) layer. And experiment and discussion for choosing GATv2 can be seen in Appendix . Normally, the inputs for graph neural network can be divided into two parts: 1): node feature; and 2): edge index. The normalized RNA expression is taken as the node feature matrix as input. The edge index representing the adjacency structure of the graph is generated in Section construction of feature graph by using KNN methods. Then it generates the spots embedding by involving parallel multi-head attention mechanisms followed by aggregation. Let x_i be the normalized RNA expression of spot i (node is the spot) and L be the number of layer for graph attention layer, H is heads number, F_{in} is the input feature dimensions and F_{out} is the output feature dimensions. The formula derivation is as follows:

Linear Transformation: Graph attention layer first applies a linear transformation to the input feature $\mathbf{h}_i$ to generate an intermediate representation for each head.

$$\mathbf{z}_i^{(k)} = \mathbf{W}^{(k)}\mathbf{h}_i^{(k)}, \tag{1}$$

where $\mathbf{h}_i^{(0)} = x_i, \forall i \in \{1, 2, 3, .., N\}$ for spot i and the same representation for spot j is $\mathbf{h}_j^{(0)} = x_j, \forall j \in \{1, 2, 3, .., N\}$. $\mathbf{W}^{(k)} \in R^{F_{out}\times F}$ is the weight matrix and $k-$th ($k \in \{1, 2, 3, ..., L-1\}$.

Attention Score: uses a dynamic attention mechanism, first concatenating the raw features of the query node and its neighbors, then computing the attention score $e_{ij}^{(k)}$.

$$e_{ij}^{(k)} = \mathbf{a}^{(k)\top} \cdot \text{LeakyReLU}\left(\mathbf{W}_a^{(k)}[\mathbf{h}_i^{(k)}||\mathbf{h}_j^{(k)}]\right), \tag{2}$$

where $\mathbf{W_a}^{(k)} \in R^{F_{out}\times H*F_{in}}$ is the linear transformation matrix for attention in the k-th head and $\mathbf{a}^{(k)} \in F_{out}$ is the attention vector for the k-th head.

Attention Normalization: normalizes the attention scores across the neighbor set N (including the node itself if self-loops exist) using softmax to get normalized attention coefficient $\alpha_{ij}^{(k)}$:

$$\alpha_{ij}^{(k)} = \text{softmax}_j(e_{ij}^{(k)}) = \frac{\exp(e_{ij}^{(k)})}{\sum_{m\in\mathcal{N}_i}\exp(e_{im}^{(k)})} \tag{3}$$

Feature Aggregation: uses the normalized attention coefficients to weight and aggregate the neighbor features, producing the output $\bar{\mathbf{h}}_i^{(k)} \in F_{out}$ for each head.

$$\bar{\mathbf{h}}_i^{(k)} = \sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(k)}\mathbf{W}^{(k)}\mathbf{h}_j^{(k)} \tag{4}$$

Multi-Head Aggregation: outputs of the multi-head are averaged rather than concatenated.

$$\mathbf{h}_i^{(k)} = \frac{1}{H}\sum_{k=1}^{H}\bar{\mathbf{h}}_i^{(k)} = \frac{1}{H}\sum_{k=1}^{H}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(k)}\mathbf{W}^{(k)}\mathbf{h}_j^{(k)}, \tag{5}$$

where $\mathbf{h}_i^{(k)} \in F_{out}$ is the final embedding of spot i.

Encoder: The encoder in STProtein framework consists of two graph attention layer in sequence with respective non-linear activation function ReLU with one linear layer in the final layer, which can be called one graph attention block. $\mathbf{h}_{1,j}^{(k)}$ and $\mathbf{h}_{2,i}^{(k)}$ represent the first and second outputs of graph attention layer for spot j and spot i in the graph attention

block respectively. In encoder , input $\mathbf{h}_j^{(0)} = x_j, \forall j \in \{1, 2, 3, .., N\}$, the output in $k-$th ($k \in \{1, 2, 3, ..., L-1\}$ layer can be defined as follows:

$$\mathbf{h}_{1,j}^{(k)} = \text{ReLU}\left(\frac{1}{H}\sum_{k=1}^{H}\sum_{j\in\mathcal{N}_i}\alpha_{1,ij}^{(k)}\mathbf{W}_1^{(k)}\mathbf{h}_j^{(k-1)}\right) \quad (6)$$

$$\mathbf{h}_{2,i}^{(k)} = \text{ReLU}\left(\frac{1}{H}\sum_{k=1}^{H}\sum_{j\in\mathcal{N}_i}\alpha_{2,ij}^{(k)}\mathbf{W}_2^{(k)}\mathbf{h}_{1,j}^{(k)}\right), \quad (7)$$

where $\alpha_{2,ij}^{(k)}$ and $\alpha_{1,ij}^{(k)}$ are the edge weight between spot i and spot j in the output of the k-th graph attention block in encoder respectively.

After two graph attention layer with nonlinear activation function ReLU. The output of encoder after the linear layer is $\mathbf{h}_{\text{enc},i}^{(k)}$, which can be defined as follows:

$$\mathbf{h}_{\text{enc},i}^{(k)} = \mathbf{W}_{\text{fc}}\mathbf{h}_{2,i}^{(k)} + \mathbf{b}_{\text{fc}} \quad (8)$$

The output of encoder is considered as the reconstructed protein normalized expression.

Decoder: Compared with encoder, decoder reverses the embedding for reconstructed protein normalized expression back into the RNA reconstructed normalized expression profile. The input of the decoder is represented as $\hat{\mathbf{h}}_j^{(k)}$ for spot j, $\hat{\mathbf{h}}_j^{(L)} = \mathbf{h}_j^{(L)}$, and output in k-th $k \in \{1, 2, .., L-1, L\}$ layer can be formulated as follows:

$$\hat{\mathbf{h}}_{1,j}^{(k-1)} = \text{ReLU}\left(\frac{1}{H}\sum_{k=1}^{H}\sum_{j\in\mathcal{N}_i}\hat{\alpha}_{1,ij}^{(k-1)}\hat{\mathbf{W}}_1^{(k)}\hat{\mathbf{h}}_j^{(k)}\right) \quad (9)$$

$$\hat{\mathbf{h}}_{2,i}^{(k-1)} = \text{ReLU}\left(\frac{1}{H}\sum_{k=1}^{H}\sum_{j\in\mathcal{N}_i}\hat{\alpha}_{2,ij}^{(k-1)}\hat{\mathbf{W}}_2^{(k)}\hat{\mathbf{h}}_{1,j}^{(k-1)}\right), \quad (10)$$

where $\hat{\alpha}_{2,ij}^{(k)}$ and $\hat{\alpha}_{1,ij}^{(k)}$ are the edge weight between spot i and spot j in the output of the k-th graph attention block in decoder respectively.

After two graph attention layer with nonlinear activation function ReLU. The output of decoder after the linear layer is $\hat{\mathbf{h}}_{\text{enc},i}^{(k)}$, which can be defined as follows:

$$\hat{\mathbf{h}}_{\text{dec},i}^{(k-1)} = \bar{\mathbf{W}}_{\text{fc}}\hat{\mathbf{h}}_{2,i}^{(k-1)} + \bar{\mathbf{b}}_{\text{fc}} \quad (11)$$

To avoid the overfitting problem, STProtein sets $\hat{\mathbf{W}}_1^{(1)} = \mathbf{W}_1^{(1)}$ $\hat{\mathbf{W}}_2^{(1)} = \mathbf{W}_2^{(1)}$, $\hat{\alpha}_{1,ij}^{(k)} = \alpha_{1,ij}^{(k)}$ and $\hat{\alpha}_{2,ij}^{(k)} = \alpha_{2,ij}^{(k)}$ respectively.

The output of decoder is considered as the reconstructed RNA normalized expression.

Loss Function: The loss function design for STProtein uses the multi-task learning (MTL) strategy. And the its objective is to minimize the reconstruction loss of RNA normalized expression and corresponding protein normalized expression together. The two loss function for RNA and protein can be formulated as follows:

$$\mathbf{L}_{\text{rna}} = \sum_{i=1}^{N}\|x_i - \hat{x}_i\|^2 \quad (12)$$

$$\mathbf{L}_{\text{protien}} = \sum_{i=1}^{N}\|y_i - \hat{y}_i\|^2, \quad (13)$$

where $\hat{x}_i$ and $\hat{y}_i$ represent the predicted RNA and protein normalized expression for spot i. x_i and y_i represent the ground truth of RNA and protein normalized expression for spot i. β_1 and β_2 are two parameters, indicating the impact weights for RNA reconstruction impact factor and protein reconstruction impact factor. Thus, total loss function can be given as follows:

$$\mathbf{L}_{\text{total}} = \beta_1\mathbf{L}_{\text{rna}} + \beta_2\mathbf{L}_{\text{protien}} \quad (14)$$

Upstream and Downstream Tasks

After training of STProtein on known multi-omics dataset containing both spatial RNA expression data and corresponding protein expression data, we can use the pre-trained model to do the upstream and down stream tasks on another dataset that only contain RNA expression table and corresponding spatial information. The workflow for upstream and downstream tasks by using STProtein can be seen in Figure 2

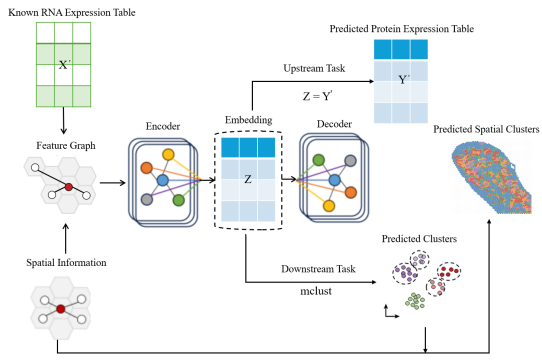

Figure 2: Workflow for Upstream and Downstream Tasks by Using STProtein

Upstream Task of Protein Expression Prediction: Due to the spatial and novel design of STProtein, the embedding represents the actual normalized protein expression table, which means $Z = Y$, Z is the embedding for STProtein and Y is the protein expression table. We can use the pre-trained model, which was trained on another dataset containing both transcriptomics and corresponding protein data. In order to predict unknown spatial protein expression in new dataset with known transcriptomics data. As shown in the Figure 2, the input to the pre-trained model is the known transcriptomics KNN feature graph that is constructed by known RNA expression table and relevant spatial information. The new embedding is the corresponding spatial protein expression table.

Downstream Task of Clustering: As shown in Figure 2, after protein expression prediction, we perform clustering analysis using the protein embedding to identify the protein spatial domains by observing the spatial clusters. There are three primary clustering tools commonly used in spatial omics: 1) mclust; 2) leiden; and 3) louvain. In our parameter sensitivity experiments in Appendix , we observed that the mclust algorithm generally outperforms both leiden and louvain when the number of labels is known to identify spatial domains in most cases.

Experiments

Benchmarking Prediction Methods

To benchmark the prediction performance, we compare STProtein against four state-of-the-art deep learning-based multi-omics prediction methods: totalVI (Gayoso et al. 2021), scArches (Lotfollahi et al. 2022), Dengkw (Lance et al. 2022) and cTp_net (Zhou et al. 2020) which serve as our baselines. A detailed introduction to these benchmarking methods is provided in Appendix .

Quantitative Evaluation Metrics

Our quantitative evaluation metrics includes in two aspects: 1): Quantitative upstream evaluation metrics assess the model's performance in terms of the prediction accuracy of protein expression; and 2): Quantitative downstream evaluation metrics on annotation datasets, which reveals the quality of prediction protein distribution in the spatial context. For quantitative upstream evaluation of protein expression prediction, the prediction accuracy of protein expression is assessed using the Root Mean Squared Error (RMSE) metric. For quantitative downstream evaluation of clustering, the quality of the predicted protein distribution within the spatial context is assessed through clustering analysis. Thus, we decided to employ a set of several metrics together related to evaluating accuracy of clustering (e.g. NMI (Danon et al. 2005), AMI (Vinh, Epps, and Bailey 2009), FMI (Walach et al. 2006), ARI (Hubert and Arabie 1985), Homogeneity (Rosenberg and Hirschberg 2007), V-measure (Rosenberg and Hirschberg 2007), F1-Score (Chicco and Jurman 2020) and Jaccard (Niwattanakul et al. 2013)), ensuring that results are both statistically significant and biologically relevant (Xu et al. 2023).

Implementation details of STProrptein

In all experiments, the encoder of STProrptein is set as a two-layer neural network with the graph attention layer, and the decoder is set as the same number of layers as the encoder. Adam optimizer is used to minimize the reconstruction loss with an initial learning rate of 1e-4. The weight decay is set as 1e-4. The epoaches for training is 12000. The activation function is set as the ReLU. The weights β_1 and β_2 for RNA and protein reconstruction loss are 5 and 3, respectively. The parameter sensitivity experiment about weights β_1 and β_2 and support for this kind of weights' setting $\beta_1 = 5$ and $\beta_2 = 3$ can seen in Appendix . The input feature graph is based on KNN methods to construct.

Results

Quantitative Upstream Evaluation of Protein Expression Prediction

We first performed quantitative upstream evaluation of protein expression prediction on three datasets: 1): Mouse spleen (SPOTS); 2): Mouse Thymus (Stereo-CITE-seq); and 3): Human Lymph Node (10x Genomics Visium). We compare with four state-of-the-art methods that mentioned in Section and use RMSE metrics mentioned in Section to evaluate the performance of the prediction results. The Table 1 summarizes the quantitative results of prediction outcomes on these three datasets. STProtein outperforms other four benchmarking methods at all three datasets: For RMSE value, it is 0.04 higher on Mouse Spleen Dataset; 0.07 higher on Mouse Thymus Dataset and 0.17 higher on Human Lymph Node Dataset. The results demonstrate that our method, STProtein, can more effectively learn important features and generate reliable, comprehensive embeddings and protein expression predictions from complex spatial multi-omics data.

Quantitative Downstream Evaluation of Clustering

We then performed quantitative evaluation of clustering on three tasets: 1): Mouse spleen (SPOTS); 2): Mouse Thymus (Stereo-CITE-seq); and 3): Human Lymph Node (10x Genomics Visium). Similar to the quantitative upstream evaluation, we compare STProtein with four state-of-the-art methods and use metrics mentioned before for evaluation. The Tables 2 - 4 summarize the quantitative results of quantitative downstream evaluation on three datasets. Similar to its prediction performance, the STProtein method also outperforms the four benchmark methods across all three datasets and almost all seven metrics, consistently. According to the results shown in Table 3, although the NMI and AMI metrics of STProtein are lower than those of Dengkw and scArches, it outperforms both methods on the remaining five metrics. Overall, STProtein demonstrates superior performance across comprehensive evaluation metrics on downstream clustering task on three platforms with different resolutions, which further proves the effectiveness of STProtein.

Furthermore, both upstream and downstream quantitative evaluation results demonstrate that the STProtein method exhibits superior capability for multi-omics learning across different datasets and tasks.

Ablation Study

In the ablation study, we evaluate the effectiveness and contribution of STProtein's graph construction and loss function on the Mouse Spleen Dataset (SPOTS). For STProtein, it use KNN methods to feature graph rather than use spatial neighbor (SN) feature graph as the graph neural network's input. $\mathbf{L}_{\text{protien}}$ represents the loss function only use the protein loss function. Similarly, $\mathbf{L}_{\text{rna}}$ represents the loss function only use the RNA loss function. Whereas, STProtein's loss function combine the protein loss function and RNA loss function together.

Table 1: Quantitative Upstream Evaluation on Three Datasets by Using RMSE Metric

Methods	Mouse Spleen Dataset	Mouse Thymus Dataset	Human Lymph Node Dataset
totalVI	1.05	1.42	1.22
scArches	1.03	1.38	1.20
Dengkw	0.99	1.05	1.17
cTp_net	1.27	1.47	1.27
STProtein	**0.95**	**0.98**	**1.00**

Table 2: Quantitative Downstream Evaluation on Mouse Spleen Dataset (SPOTS)

Methods	NMI(%)	AMI(%)	FMI(%)	ARI(%)	V-Measure(%)	F1-Score(%)	Jaccard(%)
totalVI	25.58	25.39	41.49	4.84	25.58	39.42	24.55
scArches	26.40	26.21	43.28	5.28	26.40	40.61	25.47
Dengkw	28.69	28.52	39.66	8.19	28.69	39.46	24.58
cTp_net	17.22	17.04	42.33	18.39	17.22	42.27	26.80
STProtein	**30.91**	**30.75**	**60.35**	**40.43**	**30.91**	**59.74**	**42.59**

Table 3: Quantitative Downstream Evaluation on Mouse Thymus Dataset (Stero-CITE-seq)

Methods	NMI(%)	AMI(%)	FMI(%)	ARI(%)	V-Measure(%)	F1-score(%)	Jaccard(%)
totalVI	39.03	38.90	53.64	32.77	39.03	51.55	34.72
scArches	45.01	44.89	56.45	36.51	45.96	54.25	37.22
Dengkw	**45.04**	**45.32**	58.79	38.94	47.04	57.37	40.22
cTp_net	43.35	43.23	57.98	38.18	46.35	56.23	39.11
STProtein	43.85	43.71	**63.25**	**40.35**	**49.85**	**59.19**	**47.72**

Table 4: Quantitative Downstream Evaluation on Human Lymph Node Dataset (10x Genomics Visium)

Methods	NMI(%)	AMI(%)	FMI(%)	ARI(%)	V-Measure(%)	F1-score(%)	Jaccard(%)
totalVI	29.90	29.70	51.56	21.63	29.90	50.63	34.10
scArches	41.71	41.55	52.80	29.12	41.71	52.77	35.84
Dengkw	41.18	41.00	55.75	35.67	41.18	56.73	40.57
cTp_net	35.66	35.48	48.13	24.42	35.66	48.11	31.67
STProtein	**42.31**	**42.12**	**56.82**	**35.97**	**42.31**	**57.22**	**41.26**

Table 5: Ablation study for feature graph construction and loss function design on prediction task

Methods	RMSE
STProtein	**0.95**
SN Feature Graph	0.96
$\mathbf{L}_{\text{protien}}$	0.97
$\mathbf{L}_{\text{rna}}$	1.02

According to Table 5 and Table 6, KNN feature graph construction that used in STProtein has better performance than the way of SN feature graph construction. Additionally, using either the RNA loss function or the protein loss function alone results in worse performance compared to incorporating both RNA and protein terms together in the loss function. The design of this combined loss function is inspired by the MTL strategy, which provides multi-perspective constraints on the learning process for both prediction and clustering tasks, ultimately leading to improved performance.

Scientific Discovery by Using STProtein

In this part, we conduct more analytical experiments and a case study on the Mouse Spleen Dataset (SPOTS) to support the fact that the power of STProtein for scientific discovery. First, we explore the ability for STProtein to accurately predict the tissue structures and marker gene distribution on different sequencing platforms. Then, we conduct the case study to explore the great power for STProtein in observing and resolving unknown mouse spleen structure and spleen macrophage subsets.

STProtein Accurately Enables the Protein Prediction of Tissue Structures and Maker Gene Distribution: Differ-

Table 6: Ablation study for feature graph construction and loss function design on clustering task

Methods	NMI(%)	AMI(%)	FMI(%)	ARI(%)	V-Measure(%)	F1-Score(%)	Jaccard(%)
STProtein	**30.91**	**30.75**	**60.35**	**40.43**	**30.91**	**59.74**	**42.59**
SN Feature Graph	27.05	26.89	44.49	21.31	27.05	42.72	27.16
$\mathbf{L}_{\text{protien}}$	29.10	28.94	43.43	19.12	29.10	42.06	26.63
$\mathbf{L}_{\text{rna}}$	20.56	20.38	37.85	9.23	20.56	37.20	22.85

ent spatial multi-omics sequencing platforms have different resolutions. The three dataset used in this research are sequenced on different platforms: 1): Mouse Spleen Dataset (SPOTS); 2): Mouse Thymus Datset (Stero-CITE-seq); and 3): Human Lymph Node (10x Genomics Visium). Building a robust model can adapt different sequencing platforms with different resolutions is of great importance. Next, we show clustering results about comparison of benchmarking prediction methods and STProtein with original Ground Truth on three dataset: Mouse Spleen Dataset (SPOTS) in Appendix Figure 5, Mouse Thymus Dataset (Stero-CITE-seq) in Appendix Figure 6 and Human Lymph Node (10x Genomics Visium) in Appendix Figure 7. The quantitative benchmarking results using metrics have been mentioned in Section . We discover that compared with benchmarking prediction methods, STProtein can better predict marker gene distribution with different resolutions.

STProtein Accurately Resolves Unknown Mouse Spleen Structures and Spleen Macrophage Subsets: We use the Mouse Spleen Dataset (SPOTS) to make a meaningful case study to support the idea that STProtein can empower the scientific discovery for life science research. For domain knowledge in biology, spleen plays a vital role in the lymphatic and immunity system, which a complex structure with an array of immune cells like T cells and B cells. The mouse spleen's histological image can be seen in Figure 3a.

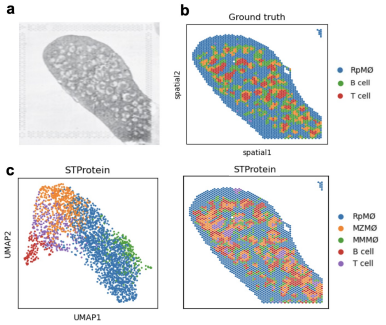

Figure 3: a): H&E Histological Image for Mouse Spleen Structure; b): Ground Truth of Clustering Results for Mouse Spleen with its Orignial Annotation (RpMZΦ, B Cell and T Cell) Shown in the Right.; c): UMAP Picture and clustering Visualization Picture for STProtein with its Annotation (RpMZΦ, MZMΦ, MMMΦ, B Cell and T Cell) Shown in the Right.

As shown in Figure 3b, original study's ground truth only annotates three kinds of cells (RpMZΦ, B cell and T cell). However, STProtein's final annotated result shown in Figure 3c can generate more clusters (MZMΦ and MMMΦ), which is the "Dark Matter" discovered by STProtein.

Thus, based on above analysis and detailed instruments in Appendix we can re-annotate the clusters on Mouse Spleen Dataset (SPOTS). The new annotation of marker genes and visualization of individual maker gene can both be seen in Appendix Figure 10. From new annotation figure by using STProtien, it can show that B cells and T cells highly coexisted, MMMΦ was distributed around the white pulp, which can be seen in Figure 3a. And the positive correlation between macrophage subsets (RpMZΦ and MZMΦ, MZMΦ and MMMΦ) reflected the hierarchical structure of red pulp-marginal zone around white pulp.

Discussion

STProtein is a novel deep learning framework based on graph neural network with multi-task learning strategy. STProtein can leverage great power in spatial protein expression prediction form spatial multi-omics data. It also can be a powerful tool to address the problem for sarcity of spatial proteomics data, the critical bottleneck in spatial multi-omics research. From a practical view, STProtein is designed to be computationally efficient. It only requires only resources such as an NVIDIA RTX 4090 GPU for training and inference. The case study on the Mouse Spleen Dataset (SPOTS) reveals STProtein's great power in scientific research in life science filed, It can uncover previously undetected macrophage subsets and providing new annotations for marker genes.

Limitation and Future Work

Despite its advantages, STProtein still has limitations that should take into consideration. The model may not accurately capture subtle structural similarities within tissues, particularly in cases where spatial adjacency plays a more dominant role than latent feature similarity. This stems from its reliance on KNN-based feature graph construction, which prioritizes global relationships over local spatial constraints. Our Future work could explore the possibility of integrating H&E image information alongside spatial transcriptomics data could enhance STProtein's ability to predict spatial protein expression by providing visual cues about tissue structure and cellular shape. This multimodal approach could be implemented by extending the current GNN framework to include image-derived features' block: potentially using the foundation model for H&E histological image like UNI (Chen et al. 2024), CHIEF (Wang et al. 2024) and TITAN (Ding et al. 2024).

References

Abdi, H.; and Williams, L. J. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4): 433–459.

Abe, K.; and Shimamura, T. 2023. UNMF: a unified nonnegative matrix factorization for multi-dimensional omics data. *Briefings in bioinformatics*, 24(5): bbad253.

Argelaguet, R.; Arnol, D.; Bredikhin, D.; Deloro, Y.; Velten, B.; Marioni, J. C.; and Stegle, O. 2020. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. *Genome biology*, 21: 1–17.

Ben-Chetrit, N.; Niu, X.; Swett, A. D.; Sotelo, J.; Jiao, M. S.; Stewart, C. M.; Potenski, C.; Mielinis, P.; Roelli, P.; Stoeckius, M.; et al. 2023. Integration of whole transcriptome spatial profiling with protein markers. *Nature biotechnology*, 41(6): 788–793.

Brody, S.; Alon, U.; and Yahav, E. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

Chen, R. J.; Ding, T.; Lu, M. Y.; Williamson, D. F.; Jaume, G.; Chen, B.; Zhang, A.; Shao, D.; Song, A. H.; Shaban, M.; et al. 2024. Towards a General-Purpose Foundation Model for Computational Pathology. *Nature Medicine*.

Chicco, D.; and Jurman, G. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21: 1–13.

Dann, E.; Henderson, N. C.; Teichmann, S. A.; Morgan, M. D.; and Marioni, J. C. 2022. Differential abundance testing on single-cell data using k-nearest neighbor graphs. *Nature biotechnology*, 40(2): 245–253.

Danon, L.; Diaz-Guilera, A.; Duch, J.; and Arenas, A. 2005. Comparing community structure identification. *Journal of statistical mechanics: Theory and experiment*, 2005(09): P09008.

Ding, T.; Wagner, S. J.; Song, A. H.; Chen, R. J.; Lu, M. Y.; Zhang, A.; Vaidya, A. J.; Jaume, G.; Shaban, M.; Kim, A.; Williamson, D. F. K.; Chen, B.; Almagro-Perez, C.; Doucet, P.; Sahai, S.; Chen, C.; Komura, D.; Kawabe, A.; Ishikawa, S.; Gerber, G.; Peng, T.; Le, L. P.; and Mahmood, F. 2024. Multimodal Whole Slide Foundation Model for Pathology. arXiv:2411.19666.

Dong, K.; and Zhang, S. 2022. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nature communications*, 13(1): 1739.

Fey, M.; and Lenssen, J. E. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Gayoso, A.; Steier, Z.; Lopez, R.; Regier, J.; Nazor, K. L.; Streets, A.; and Yosef, N. 2021. Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nature methods*, 18(3): 272–282.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Han, J.-R.; Li, S.; Li, W.-J.; and Dong, L. 2024. Mining microbial and metabolic dark matter in extreme environments: a roadmap for harnessing the power of multi-omics data. *Advanced Biotechnology*, 2(3): 26.

Hubert, L.; and Arabie, P. 1985. Comparing partitions. *Journal of classification*, 2: 193–218.

Janesick, A.; Shelansky, R.; Gottscho, A. D.; Wagner, F.; Williams, S. R.; Rouault, M.; Beliakoff, G.; Morrison, C. A.; Oliveira, M. F.; Sicherman, J. T.; et al. 2023. High resolution mapping of the tumor microenvironment using integrated single-cell, spatial and in situ analysis. *Nature Communications*, 14(1): 8353.

Kang, S. 2021. K-nearest neighbor learning with graph neural networks. *Mathematics*, 9(8): 830.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lance, C.; Luecken, M. D.; Burkhardt, D. B.; Cannoodt, R.; Rautenstrauch, P.; Laddach, A.; Ubingazhibov, A.; Cao, Z.-J.; Deng, K.; Khan, S.; et al. 2022. Multimodal single cell data integration challenge: results and lessons learned. *BioRxiv*, 2022–04.

Liao, S.; Heng, Y.; Liu, W.; Xiang, J.; Ma, Y.; Chen, L.; Feng, X.; Jia, D.; Liang, D.; Huang, C.; et al. 2023. Integrated spatial transcriptomic and proteomic analysis of fresh frozen tissue based on stereo-seq. *bioRxiv*, 2023–04.

Liu, Y.; DiStasio, M.; Su, G.; Asashima, H.; Enninful, A.; Qin, X.; Deng, Y.; Nam, J.; Gao, F.; Bordignon, P.; et al. 2023. High-plex protein and whole transcriptome co-mapping at cellular resolution with spatial CITE-seq. *Nature Biotechnology*, 41(10): 1405–1409.

Long, Y.; Ang, K. S.; Sethi, R.; Liao, S.; Heng, Y.; van Olst, L.; Ye, S.; Zhong, C.; Xu, H.; Zhang, D.; et al. 2024. Deciphering spatial domains from spatial multi-omics with SpatialGlue. *Nature Methods*, 1–10.

Lotfollahi, M.; Naghipourfar, M.; Luecken, M. D.; Khajavi, M.; Büttner, M.; Wagenstetter, M.; Avsec, Ž.; Gayoso, A.; Yosef, N.; Interlandi, M.; et al. 2022. Mapping single-cell data to reference atlases by transfer learning. *Nature biotechnology*, 40(1): 121–130.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Niwattanakul, S.; Singthongchai, J.; Naenudorn, E.; and Wanapu, S. 2013. Using of Jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, 380–384.

Palla, G.; Spitzer, H.; Klein, M.; Fischer, D.; Schaar, A. C.; Kuemmerle, L. B.; Rybakov, S.; Ibarra, I. L.; Holmberg, O.; Virshup, I.; et al. 2022. Squidpy: a scalable framework for spatial omics analysis. *Nature methods*, 19(2): 171–178.

Pedregosa, F. 2011. Scikit-learn: Machine learning in python Fabian. *Journal of machine learning research*, 12: 2825.

Rosenberg, A.; and Hirschberg, J. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 410–420.

Satija, R.; Farrell, J. A.; Gennert, D.; Schier, A. F.; and Regev, A. 2015. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5): 495–502.

Shi, Y.; Huang, Z.; Feng, S.; Zhong, H.; Wang, W.; and Sun, Y. 2020. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*.

Townes, F. W.; Hicks, S. C.; Aryee, M. J.; and Irizarry, R. A. 2019. Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome biology*, 20: 1–16.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y.; et al. 2017. Graph attention networks. *stat*, 1050(20): 10–48550.

Vinh, N.; Epps, J.; and Bailey, J. 2009. Information Theoretic Measures for Clusterings Comparison: Variants. *Properties, Normalization and Correction for Chance*, 18.

Walach, H.; Buchheld, N.; Buttenmüller, V.; Kleinknecht, N.; and Schmidt, S. 2006. Measuring mindfulness—the Freiburg mindfulness inventory (FMI). *Personality and individual differences*, 40(8): 1543–1555.

Wang, X.; Zhao, J.; Marostica, E.; Yuan, W.; Jin, J.; Zhang, J.; Li, R.; Tang, H.; Wang, K.; Li, Y.; et al. 2024. A pathology foundation model for cancer diagnosis and prognosis prediction. *Nature*, 634(8035): 970–978.

Wolf, F. A.; Angerer, P.; and Theis, F. J. 2018. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology*, 19: 1–5.

Xu, H.; Wang, S.; Fang, M.; Luo, S.; Chen, C.; Wan, S.; Wang, R.; Tang, M.; Xue, T.; Li, B.; et al. 2023. SPACEL: deep learning-based characterization of spatial transcriptome architectures. *Nature Communications*, 14(1): 7603.

Zeng, H.; Huang, J.; Zhou, H.; Meilandt, W. J.; Dejanovic, B.; Zhou, Y.; Bohlen, C. J.; Lee, S.-H.; Ren, J.; Liu, A.; et al. 2023. Integrative in situ mapping of single-cell transcriptional states and tissue histopathology in a mouse model of Alzheimer's disease. *Nature Neuroscience*, 26(3): 430–446.

Zeng, Z.; Ma, Y.; Hu, L.; Tan, B.; Liu, P.; Wang, Y.; Xing, C.; Xiong, Y.; and Du, H. 2024. OmicVerse: a framework for bridging and deepening insights across bulk and single-cell sequencing. *Nature Communications*, 15(1): 5983.

Zhou, Z.; Ye, C.; Wang, J.; and Zhang, N. R. 2020. Surface protein imputation from single cell transcriptomes by deep neural networks. *Nature communications*, 11(1): 651.

Dataset

Table 7: Dataset's detialed information for STProtein

Name	Platform	Size(spots x genes/proteins)
Mouse spleen replicate 1	SPOTS (RNA-protein)	2,568x32,285 2,568x21
Mouse spleen replicate 2	SPOTS (RNA-protein)	2,768x32,285 2,768x21
Mouse Thymus 1	Stereo-CITE-seq (RNA-protein)	4,697x23,622 4,697x51
Mouse Thymus 2	Stereo-CITE-seq (RNA-protein)	4,253x23,529 4,253x19
Mouse Thymus 3	Stereo-CITE-seq (RNA-protein)	4,646x23,960 4,646x19
Mouse Thymus 4	Stereo-CITE-seq (RNA-protein)	4,228x23,221 4,228x19
Human Lymph Node A1	10x Genomics Visium (RNA-protein)	3,484x18,085 3,484x31
Human Lymph Node D1	10x Genomics Visium (RNA-protein)	3,359x18,085 3,359x31

According to the above table, it is clear that three types of dataset are from different sequencing platforms (SPOTS, Stero-CITE-seq and 10x Genomics Visium) with different resolutions. For clustering task, we have acquired the annotation by biological experiments, the ground truth for clustering, from original authors. Thus, we have known the cluster numbers according to the annotation ground truth.

Benchmarking Prediction Methods

totalVI: A variational autoencoder-based tool that jointly models single-cell RNA and protein data. By mapping both modalities into a shared low-dimensional latent space, it enables cross-modal predictions.

scArches: A transfer learning based fine-tuning pre-trained models to adapt to new datasets, reducing training time significantly. In protein abundance prediction, scArches can apply a model trained on large-scale single-cell data to new experiments.

Dengkw: A model uses kernel ridge regression to predict protein levels from single-cell RNA-seq data. It performs exceptionally well in intra-dataset evaluations, particularly in capturing cell-cell correlation metrics.

cTp_net: A neural network-based transfer learning framework designed for cell-type-specific protein expression prediction. When applied to diverse immune cell subsets, cTP-net uses existing multi-omics data to train models and predict protein levels in new single-cell datasets.

Quantitative Evaluation Metrics

Our quantitative evaluation metrics includes in two aspects: 1): Quantitative upstream evaluation metrics, which is the model performance evaluation about the prediction accuracy of protein expression; and 2): Quantitative downstream evaluation metrics on annotation datasets, which reveals the quality of prediction protein distribution in the spatial context.

Quantitative Upstream Evaluation Metrics of Protein Expression Prediction: Prediction accuracy of protein expression can be evaluated in the metric: Root Mean Squared Error (RMSE). The metric indicate the accuracy of our predictions and the degree of deviation from the actual values. The metric equation has been given as follows:

$$\text{RMSE} = \sqrt{\frac{\sum (y_i - y_p)^2}{n}}, \tag{15}$$

where y_i is the true protein expression, y_p is the predicted protein expression and n is the number of prediction instances. The lower values for RMSE and MAE indicate better prediction accuracy of protein expression.

Quantitative Downstream Evaluation Metrics of Clustering: The quality of prediction protein distribution in the spatial context can be evaluated by clustering. Thus, we decided to utilize combination of several metrics together related to evaluating

accuracy of clustering (e.g. NMI (Danon et al. 2005), AMI (Vinh, Epps, and Bailey 2009), FMI (Walach et al. 2006), ARI (Hubert and Arabie 1985), Homogeneity (Rosenberg and Hirschberg 2007), V-measure (Rosenberg and Hirschberg 2007), F1-Score (Chicco and Jurman 2020) and Jaccard (Niwattanakul et al. 2013)) can provide a robust system for evaluating the performance of model, ensuring that results are both statistically significant and biologically relevant (Xu et al. 2023). The seven detailed metrics significances can be clearly explained in the following Table 8.

Table 8: Significance of clustering evaluation metrics

Metrics	Significance
NMI	Indicates perfect correlation.
AMI	Adjusted for chance
FMI	Similarity between two clustering results
ARI	Measures similarity with truth table
V-measure	Combines homogeneity and completeness
F1-Score	Harmonic mean of precision and recall
Jaccard	Similarity between prediction and ground truth

The seven metrics in Table 8 all range from 0 to 1, where 1 indicates perfect clustering and 0 means poor clustering. Nine quantitative metrics (NMI (Danon et al. 2005), AMI (Vinh, Epps, and Bailey 2009), FMI (Walach et al. 2006), ARI (Hubert and Arabie 1985), V-measure (Rosenberg and Hirschberg 2007), F1-Score (Chicco and Jurman 2020) and Jaccard (Niwattanakul et al. 2013)) can be computed by using the scikit-learn (Pedregosa 2011) package in Python. As for the equations for the aforementioned six metrics, they are given in following part.

Define the cluster label of embedding predicted STProtein as X and the ground truth label as Y. $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y. And $p(x, y)$ is the joint probability distribution of X and Y. And MI, Msutual information, can be defined as following equation.

$$\mathrm{MI}(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \tag{16}$$

NMI, Normalized mutual information can be defined as the standardization of MI:

$$\mathrm{NMI}(X, Y) = \frac{\mathrm{MI}(X, Y)}{\mathrm{H}(X) * \mathrm{H}(Y)}, \tag{17}$$

where $H(X)$ and $H(Y)$ represents the entropy of predicted clusters and ground truth clusters, respectively.

AMI, Adjusted mutual information, modifies the mutual information and adjusts the expected value of MI for random clustering to minimize the influence of chance:

$$\mathrm{AMI} = \frac{\mathrm{MI} - \mathrm{E}[\mathrm{MI}]}{\max(\mathrm{MI}) - \mathrm{E}[\mathrm{MI}]}, \tag{18}$$

where $\mathrm{E}[\mathrm{MI}]$ is the average value for MI.

RI, the rand index, FMI, Fowlkes-Mallows index that measuring the similarity between two clustering outcomes by taking into account the proportion of intra-class pairs (data points within the same class) to inter-class pairs (data points across different classes) can be both defined as:

$$\mathrm{RI} = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

$$\mathrm{FMI} = \sqrt{\frac{TP^2}{(TP + FP) * (TP + FN)}}, \tag{20}$$

where TP, TN, FP and FN represent true positives, true negatives, false positives and false negtives, respectively.

ARI, Adjusted Rand Index, measures the consistency between the predicted clustering results and the reference ground truth labels:

$$\mathrm{ARI} = \frac{\mathrm{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathrm{E}[\mathrm{RI}]}, \tag{21}$$

where $\mathrm{E}[\mathrm{RI}]$ is the average value for RI.

V-measure represents the harmonic average of homogeneity and completeness:

$$\mathrm{H}(X) = -\sum_i p(x_i)\log(p(x_i)) \tag{22}$$

$$\mathrm{H}(Y) = -\sum_i p(y_i)\log(p(y_i)) \tag{23}$$

$$\text{Homogeneity} = 1 - \frac{\mathrm{H}(X|Y)}{\mathrm{H}(X)} \tag{24}$$

$$\text{Completeness} = 1 - \frac{\mathrm{H}(Y|X)}{\mathrm{H}(Y)} \tag{25}$$

$$\begin{aligned}\text{V-Measure} &= 2 * \frac{\text{Homogeneity} * \text{Completeness}}{\text{Homogeneity} + \text{Completeness}} \\ &= 2 * \frac{\mathrm{H}(X|Y) + \mathrm{H}(Y|X)}{\mathrm{H}(X) + \mathrm{H}(Y)}\end{aligned} \tag{26}$$

F1-Score represents the harmonic average of Precision and Recall:

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

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

$$\text{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{29}$$

Jaccard coefficient is employed to assess the degree of similarity between X and Y:

$$\text{Jaccard} = \frac{|X \cap Y|}{|X \cup Y|} \tag{30}$$

Experiments Protocol

The experiments for STProtein can be mainly divided into two parts: 1): Upstream experiments; and 2): Downstream experiments. For all quantitative experiments in upstream and downstream experiments, we conducted the statistics analysis by at least running the codes for 10 times and using the average values as our final results, ensuring variances within a reasonable range.

Upstream Experiments: The upstream experiments include four parts: 1): Quantitative evaluation of protein expression prediction; 2): Graph convolutional layer model comparison experiments; 3): Ablation study; and 4): Parameter sensitivity experiments. All above four experiments in upstream task are quantitative experiments. The first experiment in upstream task conducted in all three datasets: 1): Mouse spleen (SPOTS); 2): Mouse Thymus (Stereo-CITE-seq); and 3): Human Lymph Node (10x Genomics Visium). Other three experiments in upstream task only conducted in the Mouse Spleen Dataset (SPOTS).

Downstream Experiments: The downstream experiments include two parts: 1): Quantitative evaluation of clustering; and 2): Scientific discovery. The first experiment in downstream task is quantitative experiment. The second one is the case study on Mouse Spleen Dataset (SPOTS) to verify that STProtein can empower the scientific discovery in the life science research.

Graph Convolutional Layer Model Comparison Experiments

For graph convolution layer model comparison experiments, we chose five graph convolution layer: GCN (Kipf and Welling 2016), SAGE (Hamilton, Ying, and Leskovec 2017), GraphTransformer (Shi et al. 2020), GAT (Velickovic et al. 2017) and GATv2 (Brody, Alon, and Yahav 2021) and tested on both upstream and downstream tasks: 1): Protein expression prediction; and 2): Clustering on Mouse Spleen Dataset (SPOTS). The implementation and evaluation metrics are the same as the quantitative evaluation.

Table 9: Prediction Results of Different Graph Convolutional Layer on Mouse Spleen Dataset (SPOTS)

Methods	RMSE
GCN	0.98
SAGE	0.96
Transformer	0.96
GAT	0.97
GATv2	**0.95**

Table 10: Clustering Results of Different Graph Convolutional Layer on Mouse Spleen Dataset (SPOTS)

Methods	NMI(%)	AMI(%)	FMI(%)	ARI(%)	V-Measure(%)	F1-Score(%)	Jaccard(%)
GCN	21.46	21.24	38.69	12.12	29.04	37.65	23.19
SAGE	29.24	29.65	44.60	22.52	29.24	42.08	26.64
Transformer	24.13	23.91	41.20	11.37	24.13	40.91	25.71
GAT	21.95	21.73	41.70	15.30	21.95	40.85	25.66
GATv2	**30.91**	**30.75**	**60.35**	**40.43**	**30.91**	**59.74**	**42.59**

From the above Table 9 and Table 10, it is clear that GATv2 outperforms than other four graph convolutional layer for both upstream task and the downstream task's evaluation on comprehensive evaluation metrics. Thus, choosing GATv2, what is this dissertation used, can be better and Convincing choice to build STProtein framework.

Parameter Sensitivity Experiments

We conducted parameter sensitivity experiments on the Mouse Spleen Dataset (SPOTS) to verify the impact of different K values in the initial KNN feature graph, the influence of different clustering algorithm methods for STProtein clustering performance and modal reconstruction loss weights between protein loss and RNA loss on STProtein's performance.

K Value in the Initial KNN Graph: For feature graph construction, we use the KNN to construct feature graph as graph neural network's input. However, the process for construction needs the constant parameter K value. Thus, we designed the below experiments and evaluated the impact of different K values (ranging from 1, 2, 3, 4, to 5) on the performance of STProtein. As shown in Table 11 and Table 12, STProtein only has subtle fluctuations on K value's change and achieves the best results when K = 3, which is the value selected for STProtein experiments.

Table 11: Parameter sensitivity experiments for k value in KNN graph construction on prediction task

K value	RMSE
1	0.96
2	0.97
3	**0.95**
4	0.96
5	0.96

Table 12: Parameter sensitivity experiments for k value in KNN graph construction on clustering task

K value	NMI(%)	AMI(%)	FMI(%)	ARI(%)	V-Measure(%)	F1-Score(%)	Jaccard(%)
1	29.27	29.11	47.85	23.19	29.27	47.10	30.81
2	26.00	25.84	42.66	18.22	26.00	41.26	25.99
3	**30.91**	**30.75**	**60.35**	**40.43**	**30.91**	**59.74**	**42.59**
4	28.52	28.35	52.76	30.24	28.52	51.92	35.06
5	27.49	27.32	56.02	34.22	27.49	55.41	38.32

Clustering Algorithm Choices: After acquiring the embedding from the STProtein to get clusters, we need to choose a suitable clustering algorithm for use to get the better classification. Based on the previous work, we choose three main widely used three

clustering algorithm: 1): mclust; 2): leiden; and 3): louvain to make comparison. As illustrated in Table 13, mclust achieves the best performance compared with other clustering algorithms, which is the choice of clustering algorithm for STProtein experiments.

Table 13: Parameter sensitivity experiments for clustering algorithm choices on clustering task

Algorithms	NMI(%)	AMI(%)	FMI(%)	ARI(%)	V-Measure(%)	F1-Score(%)	Jaccard(%)
mclust	**30.91**	**30.75**	**60.35**	**40.43**	**30.91**	**59.74**	**42.59**
leiden	28.36	28.19	51.43	28.06	28.36	50.71	33.97
louvain	27.29	27.12	53.08	30.20	27.29	52.42	35.52

Reconstruction Loss Weights: We set the weights β_1 and β_2 for RNA and protein in ranging from 1 to 5 and test it on the Mouse Spleen dataset. As shown in Fig. 4, the performance of STProtein have some small fluctuations when weights for RNA and protein loss function items change. It can also demonstrate that the insensitivity of STProtein's performance to the reconstruction loss weights. However, the weights $\beta_1 = 5$ and $\beta_2 = 3$ for RNA and protein outperform others on comprehensive evaluation metrics, which are the values setted for STProtein experiments.

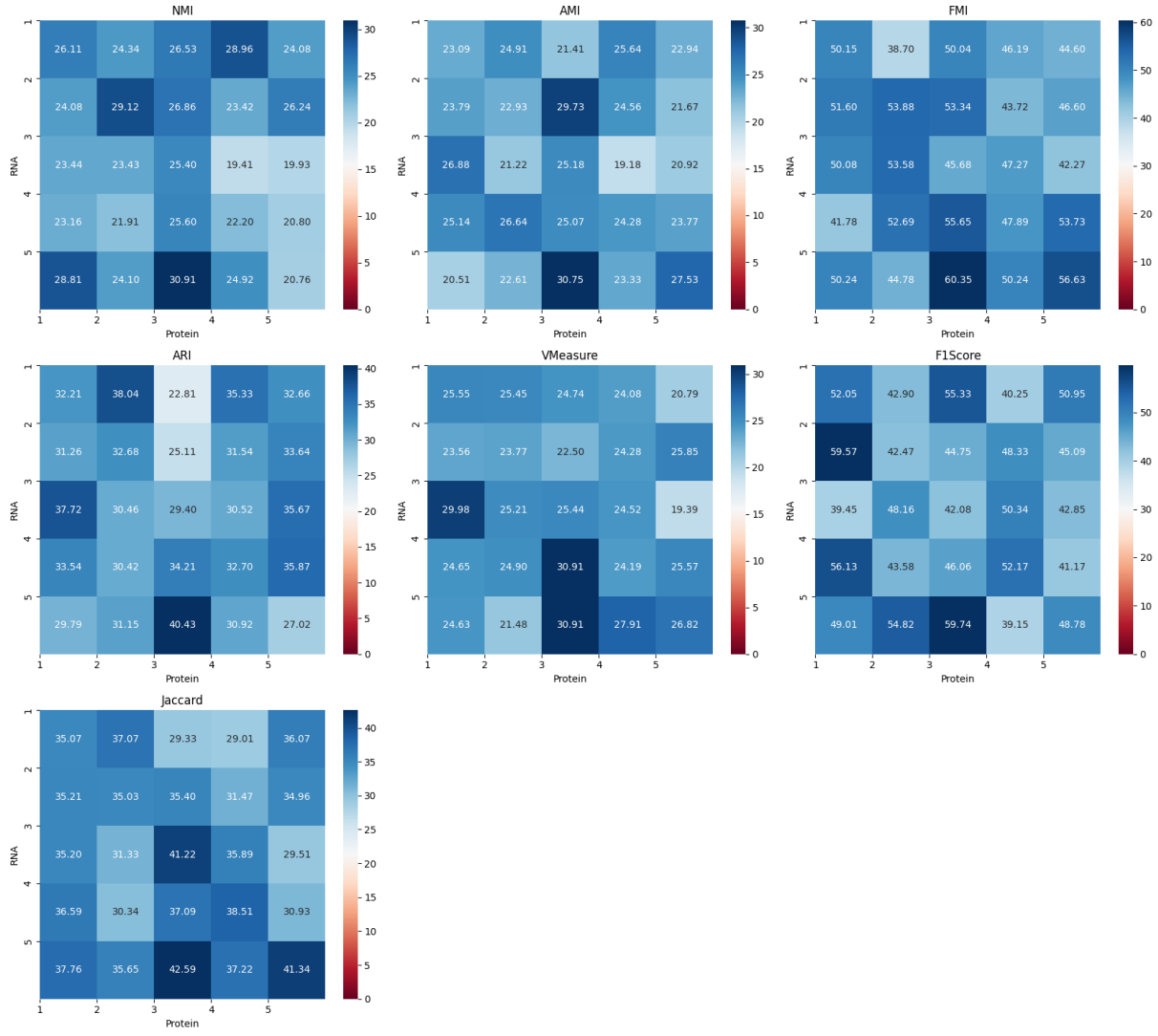

Figure 4: Parameter Sensitivity Experiments for Reconstruction Loss Weights for RNA and Protein Items on Mouse Spleen Dataset (SPOTS). The Data Presented in Heatmap are Presented in Percentages.

STProtein Accurately Enables the Protein Prediction of Tissue Structures and Maker Gene Distribution with Different Resolutions

As mentioned in Section **??** and Section , different spatial multi-omics sequencing platforms have different resolutions. And three dataset are sequenced on different platforms: 1): Mouse Spleen Dataset (SPOTS); 2): Mouse Thymus Datset (Stero-CITE-seq); and 3): Human Lymph Node (10x Genomics Visium). Building a robust model can adapt different sequencing platforms with different resolutions is of great importance. Next, we show clustering results about comparison of benchmarking prediction methods and STProtein with original Ground Truth on three dataset: Mouse Spleen Dataset (SPOTS) in Fig. 5, Mouse Thymus Dataset (Stero-CITE-seq) in Fig. 6 and Human Lymph Node (10x Genomics Visium) in Fig. 7. The quantitative benchmarking results using metrics have been mentioned in Section . We discover that compared with benchmarking prediction methods, STProtein can better predict marker gene distribution with different resolutions.

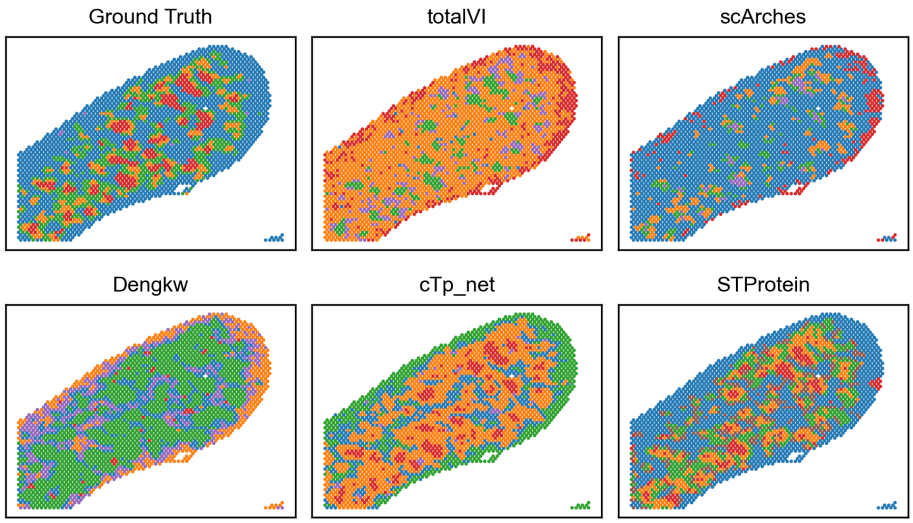

Figure 5: Visualization Results about Comparison of Benchmarking Prediction Methods and STProtein with Original Ground Truth on Mouse Spleen Dataset (SPOTS)

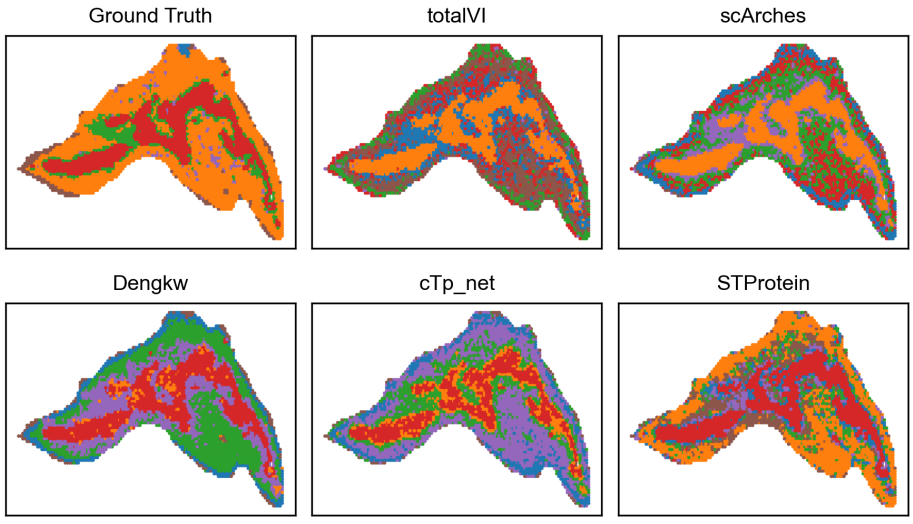

Figure 6: Visualization Results about Comparison of Benchmarking Prediction Methods and STProtein with Original Ground Truth for Mouse Thymus Dataset (Stero-CITE-seq)

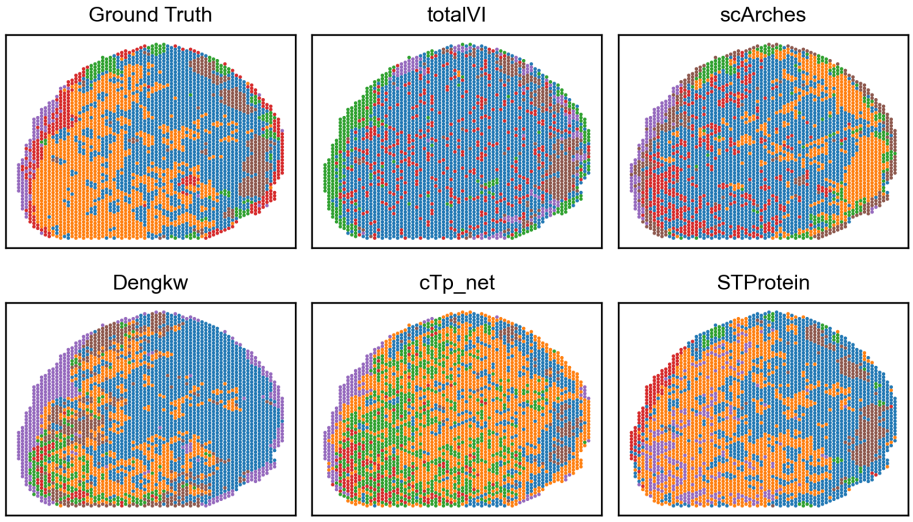

Figure 7: Visualization Results about Comparison of Benchmarking Prediction Methods and STProtein with Original Ground Truth on Human Lymph Node (10x Genomics Visium)

STProtein Accurately Resolves Unknown Mouse Spleen Structures and Spleen Macrophage Subsets

We use the Mouse Spleen Dataset (SPOTS) to make a meaningful case study to support the idea that STProtein can empower the scientific discovery for life science research. For domain knowledge in biology, spleen plays a vital role in the lymphatic and immunity system, which a complex structure with an array of immune cells like T cells and B cells.

In order to accurately annotate the clusters that discovered by STProtein, we first visualize all marker genes in Mouse Spleen Dataset (SPOTS) as shown in Fig. 9. Then based on the annotation in original study shown in Fig. 3b, we classified the some marker genes into three categories: 1): T cell; 2): B cell; and 3): RpMZΦ. And the result can be seen in Fig. 8.

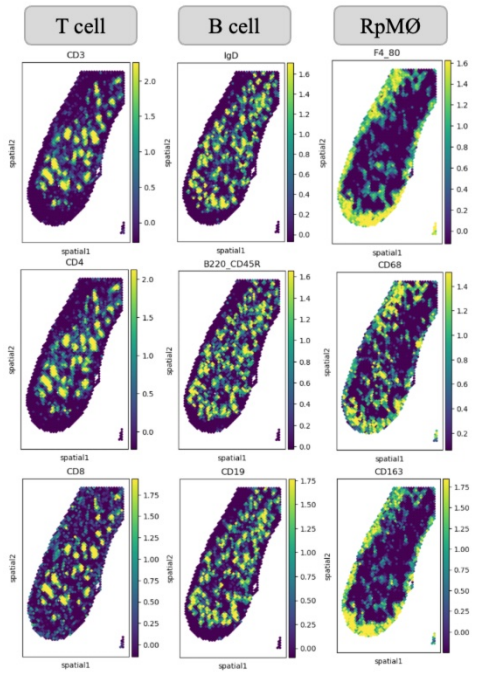

Figure 8: Spatial Visualization of RpMZΦ, B Cell and T Cell in Mouse Spleen Dataset (SPOTS)

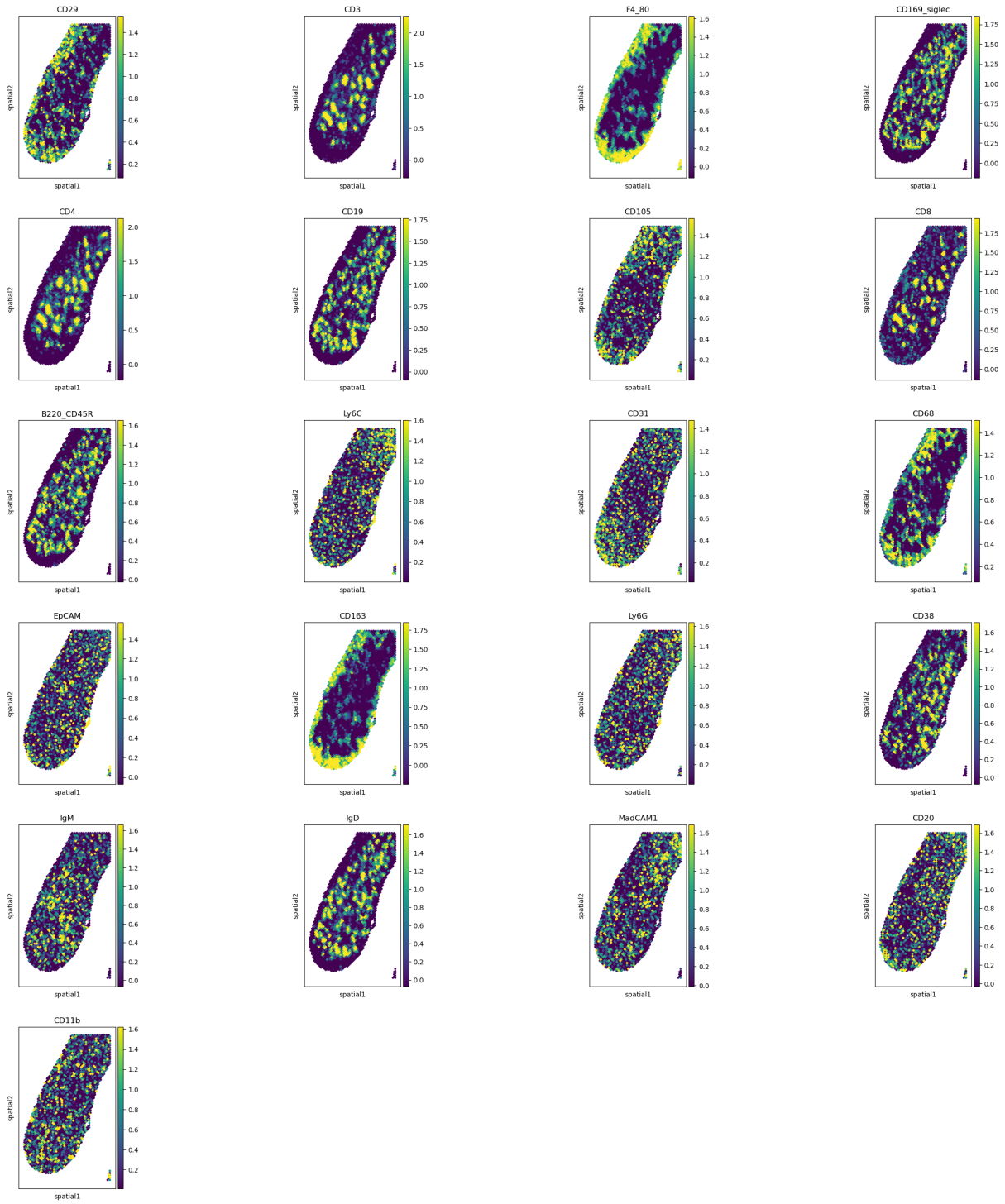

Figure 9: Spatial Visualization of all Maker Genes in Mouse Spleen Dataset (SPOTS)

According to Fig. 3 and Fig. 8, we can directly observe the T cell, B cell and RpMZΦ maker genes' distributions in mouse spleen. Spatial visualization of protein markers further revealed the distribution of cell types: B cells and T cells were concentrated in the germinal center and T cell zone, respectively, with obvious spatial proximity. RpMZΦ was clearly located by strong expression of F4_80 and CD163, while MZMΦ and MMMΦ were distinguished by specific markers such as CD29 (MZMΦ) and CD169_siglec (MMMΦ) shown in Fig. 9. They have shown different patterns of the heterogeneity of marginal zone macrophages.

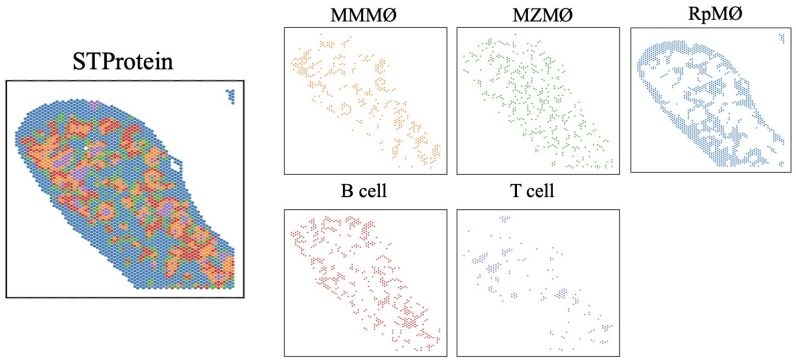

Figure 10: The New Annotation of Marker Genes on Mouse Spleen Dataset (SPOTS) by Using STProtein

In conclusion, we can use the STProtien to discover new cells (MZMΦ and MMMΦ) that are not annotated in the original study on Mouse Spleen Dataset (SPOTS) and identify macrophage subsets (RpMZΦ, MZMΦ, MMMΦ) with immue cells (T cell and B cell), which means STProtein can help scientists to find and uncover ”Dark Matter” and boost the speed of new scientific discovery. Similarly, we can continue use STProtein on different dataset like Mouse Thymus (Stero-CITE-seq) and Human Lymph Node (10x Genomics Visium) with the same analysis pipeline to uncover more ”Dark Matter” and observe more scientific discoveries in the life science filed.