

Accurate Spatial Gene Expression Prediction by Integrating Multi-Resolution Features

Youngmin Chung, Ji Hun Ha, Kyeong Chan Im, Joo Sang Lee*

Sungkyunkwan University, South Korea

ymblue@g.skku.edu, joosang.lee@skku.edu

Abstract

Recent advancements in Spatial Transcriptomics (ST) technology have facilitated detailed gene expression analysis within tissue contexts. However, the high costs and methodological limitations of ST necessitate a more robust predictive model. In response, this paper introduces TRIPLEX, a novel deep learning framework designed to predict spatial gene expression from Whole Slide Images (WSIs). TRIPLEX uniquely harnesses multi-resolution features, capturing cellular morphology at individual spots, the local context around these spots, and the global tissue organization. By integrating these features through an effective fusion strategy, TRIPLEX achieves accurate gene expression prediction. Our comprehensive benchmark study, conducted on three public ST datasets and supplemented with Visium data from 10X Genomics, demonstrates that TRIPLEX outperforms current state-of-the-art models in Mean Squared Error (MSE), Mean Absolute Error (MAE), and Pearson Correlation Coefficient (PCC). The model's predictions align closely with ground truth gene expression profiles and tumor annotations, underscoring TRIPLEX's potential in advancing cancer diagnosis and treatment.

1. Introduction

The emergence of large-scale Spatial Transcriptomics (ST) technology has facilitated the quantification of mRNA expression across a multitude of genes within the spatial context of tissue samples [24]. ST technology segments centimeter-scale Whole Slide Images (WSIs) into hundreds of thousands of small spots, each providing its gene expression profile. Considering the substantial cost associated with ST sequencing technology, coupled with the widespread availability of WSIs, a pressing question is how to best predict spatial gene expression based on WSIs using rapidly evolving computer vision techniques.

A number of studies have endeavored to address this

challenge [8, 20, 26, 28, 29]. Approaches vary, with some predicting gene expression strictly from the tissue image confined within the spot's boundaries [8], while others also take into account spatial dependencies between spot images [20, 29], or consider similarities to reference spots [26, 28]. However, we have noted several limitations inherent to these existing methodologies. Firstly, current methods primarily focus on spot images, neglecting the wealth of biological information available in the wider image context. By integrating both the specific spot and its surrounding environment, along with the holistic view of the entire histology image, we can access richer information, encompassing varied biological contexts. Secondly, models that consider interactions between spots [20, 29] face a limitation in processing the embedding of all patches in a WSI simultaneously. This approach, common in handling hundreds to thousands of patches within a WSI, limits the scalability of the patch embedding model due to resource constraints. Such limitations significantly impede the extraction of fine-grained, rich representations from each spot, thereby affecting the model's ability to perform detailed analysis of WSIs. Thirdly, model performance is frequently overestimated because of inadequate validation, such as using the limited size of dataset [8] sometimes without cross-validation [26] and training/testing with replicates from the same patient [20, 28, 29]. The limited size of ST datasets means that exclusive reliance on a single dataset for model evaluation can hinder an accurate assessment of the model's capabilities, thereby emphasizing the necessity for cross-validation. The issue is compounded when replicate data from the same patient, often featuring nearly identical image-gene expression pairs, are used in both training and testing phases. This can lead to an inflated perception of a model's effectiveness, as it may not accurately reflect the model's ability to generalize to new, unseen data. Lastly, the use of disparate datasets, diverse normalization methods, and varied evaluation techniques in existing research studies compounds the challenge of conducting fair comparisons of the models.

Addressing these limitations, we present TRIPLEX, an innovative deep learning framework designed to leverage

*Corresponding author.

multi-resolution features from histology images for robustly predicting spatial gene expression levels. TRIPLEX extracts three distinct types of features corresponding to different resolutions: the target spot image (representing the specific spot, whose gene expression to be predicted), the neighbor view (encompassing a wider area around the spot), and the global view (comprising the aggregate of all spot images). These features capture varying levels of biological information—ranging from the detailed cell morphology in the target spot image, to the surrounding tissue phenotype, and the overall tissue microenvironment in the WSI. Each is integral to understanding the spatial gene expression levels of the given spot. TRIPLEX employs separate encoders to extract these features from WSIs, each focusing on its assigned resolution to efficiently capture relevant details. For neighbour or global view with larger resolution, pre-extracted features are used to reduce the burden of computational cost, while for target spot images, encoders are fully updated to extract fine-grained information. These features are then integrated via a fusion layer for effective gene expression prediction. This approach allows TRIPLEX to utilize resolution-specific information, thereby enhancing prediction accuracy while avoiding significant increases in computational costs.

Our study sets a new benchmark in spatial gene expression prediction, comparing our model, TRIPLEX, against five prior studies [8, 20, 26, 28, 29] under uniform experimental conditions. We conduct internal evaluations using three public Spatial Transcriptomics (ST) datasets [1, 8, 11] and external validations using higher-resolution Visium data from 10X Genomics. Our validation procedure strictly avoids mixing patient sample replicates between training and testing datasets, a significant departure from previous methods [20, 28, 29], and employs rigorous cross-validation. Our results indicate that TRIPLEX surpasses existing models in terms of Mean Squared Error (MSE), Mean Absolute Error (MAE), and Pearson Correlation Coefficient (PCC) in both internal and external evaluations. Furthermore, we provide visualizations of the expression distributions for a specific gene commonly associated with cancer. These visualizations reveal that our model’s predictions align more closely with actual gene expression data and tumor annotations, demonstrating its enhanced predictive accuracy.

Our key contributions can be summarized as follows:

- We introduce an innovative approach to predict spatial gene expression levels from WSIs by integrating multiple biological contexts.
- Our proposed framework seamlessly integrates multi-resolution features. This integration is facilitated by a feature extraction strategy, the use of various types of transformers, and a fusion loss technique, all while keeping the additional computational costs to a minimum.

- Through comprehensive experiments on three public ST datasets and additional external evaluations using three Visium data, our study establishes a new benchmark in the field of spatial gene expression prediction. The results consistently show that our proposed method outperforms all existing models included in our comparative analysis.

2. Related Work

In this section, we delve into studies pertinent to our research. For clarity, the term ‘spot’ will be used to denote a predefined unit region within a WSI where gene expression is quantified. Moreover, we will use ‘target spot’ to specifically refer to the spot within a WSI for which we seek to predict gene expression.

Spatial gene expression prediction from WSIs via deep learning We review the pioneering works in this field, which aim to predict spatial gene expression from WSIs. ST-Net [8] utilizes a standard transfer learning strategy, training a Densenet121 model [9]—pretrained on ImageNet—using histology images as input and gene expression as labels. Following this, HisToGene [20] leverages Vision Transformers (ViT) [7] to account for correlations among patches in a WSI, thereby predicting gene expression from global-context aware features. Further developing this concept, Hist2ST [29] enhances the approach by emphasizing patch embedding using ConvMixer [25] and aggregating neighborhood information through graph convolution network [15]. While these previous works [20, 29] share similarities with our methodology, they predominantly process patch and global embedding sequentially, often overlooking the neighboring information around the target spot. In contrast, our method sets itself apart by concurrently extracting critical features at three distinct resolutions, including the neighbor view, and integrating them for gene expression prediction. In a different vein, EGN [28] adopts exemplar learning for predicting gene expression from histology images. This method dynamically selects the most analogous exemplars from a target spot within a WSI to enhance prediction accuracy. Additionally, BLEEP [26] introduces a bi-modal embedding framework similar to CLIP [21] to co-embed spot images and gene expression. After training, this model imputes the gene expression of a query spot using the retrieved gene expression set from a reference dataset.

Deep learning for WSIs Due to their gigapixel resolution, WSIs present a significant challenge for conventional deep learning frameworks in computer vision. To tackle this, Multiple Instance Learning (MIL) has been employed, enabling the handling of high-resolution data with sparse local annotations. In the context of WSIs, MIL approaches typically predict bag labels, such as distinguishing slides from cancer patients versus healthy individuals, by aggregating information from numerous small patches within the

WSIs [3, 13, 17, 22]. Recently, attention-based networks have been employed in MIL to aggregate all patches in WSIs, achieving state-of-the-art performance [17, 18, 22]. Moreover, Chen et al. [4] introduced a Hierarchical Image Pyramid Transformer (HIPT) to adapt Vision Transformers (ViT) for WSIs. They effectively captured the hierarchical structures of WSIs by sequentially training ViTs across images of various resolutions, employing a self-supervised learning approach. This method has shown superior results in cancer subtyping and survival prediction, surpassing previous models. Drawing inspiration from this, our proposed method is specifically designed to simultaneously handle information from these multi-resolution features for predicting spatial gene expression.

3. Method

3.1. Preliminary

In this section, we present our problem formulation and detail our proposed method, concurrently assessing it in contrast to previous methodologies. Our task is a multi-output regression problem, where we input a set of spot images from a WSI, denoted as $\mathbf{X} \in \mathbb{R}^{n \times H \times W \times 3}$, and aim to predict the gene expression levels of individual spots, represented as $\mathbf{Y} \in \mathbb{R}^{n \times m}$. Here, n denotes the number of spot images in a WSI, m represents the number of genes whose expression levels to be predicted, and H and W signify the height and width of each spot image, respectively.

ST-Net [8] approaches the prediction task by estimating $\hat{Y}_i \in \mathbb{R}^m$ based solely on the information from a single spot image $X_i \in \mathbb{R}^{H \times W \times 3}$. This is represented as:

$$\hat{Y}_i = f(X_i) \in \mathbb{R}^m \quad (1)$$

where i indexes a target spot within a WSI. ST-Net is formulated under general supervised learning principles, where each input corresponds to a unique label, and each input is treated independently.

On the other side, methods employing ViT [20, 29] predict the gene expression of all spot images concurrently, expressed as:

$$\hat{\mathbf{Y}} = f(X_1, X_2, \dots, X_n) \in \mathbb{R}^{n \times m} \quad (2)$$

In this case, the problem is defined within the context of supervised learning, but with a key difference: the inputs are interdependent, affecting the prediction outcome collectively.

EGN [28] approaches the problem from a different perspective and predicts \hat{Y}_i using X_i together with its global view G_i and its exemplar set $\{g_j, y_j\}_{j=1}^k \in K_i$. This can be formulated as:

$$\hat{Y}_i = f(X_i, G_i, K_i) \in \mathbb{R}^m \quad (3)$$

where G_i represents the features extracted from the target spot image X_i using a pretrained model. The set K_i comprises the k -nearest global views to G_i and their associated gene expression levels.

In BLEEP [26], a bi-modal pretraining phase is leveraged, employing contrastive loss [21] to generate embeddings for both images and gene expression levels. During the inference phase, a given query image X_i is processed through an image encoder, Enc^{img} , to produce its d -dimensional embedding vector v_i :

$$v_i = Enc^{img}(X_i) \in \mathbb{R}^d \quad (4)$$

Simultaneously, an expression encoder, Enc^{exp} , is applied to the gene expression levels of the reference dataset, yielding an d -dimensional embedding vectors e^{ref} represented as:

$$e^{ref} = Enc^{exp}(Y^{ref}) \in \mathbb{R}^{l \times d} \quad (5)$$

Here, l signifies the number of gene expression levels in the reference dataset. The process continues by identifying the top- k closest embeddings to v_i , denoted as $e^{top} \in \mathbb{R}^{k \times d}$. The corresponding gene expression values for these top- k embeddings, $Y^{top} \in \mathbb{R}^{k \times m}$, are then retrieved. The gene expression level for the query spot is estimated by computing the average of Y^{top} :

$$\hat{Y}_i = Average(Y^{top}) \in \mathbb{R}^m \quad (6)$$

Our proposed method diverges from the previously mentioned approaches by employing a unique combination of three input types to predict \hat{Y}_i . These are the target spot image X_i^{Ta} , the local neighbor views X_i^{Ne} , and a collection of all the global views G : g_1, g_2, \dots, g_n .

$$z_i^{Ta} = Enc^{Ta}(X_i^{Ta}) \in \mathbb{R}^{n^{Ta} \times d} \quad (7)$$

$$z_i^{Ne} = Enc^{Ne}(X_i^{Ne}) \in \mathbb{R}^{n^{Ne} \times d} \quad (8)$$

$$z^{Gl} = Enc^{Gl}(G) \in \mathbb{R}^{n \times d} \quad (9)$$

$$\hat{Y}_i = f(z_i^{Ta}, z_i^{Ne}, z_i^{Gl}) \in \mathbb{R}^m \quad (10)$$

Here, Enc^{Ta} , Enc^{Ne} , and Enc^{Gl} represent models that independently embed each type of input. The dimension of each embedded token is denoted by d . The local neighbor views X_i^{Ne} consist of the n^{Ne} adjacent patches around the target spot image X_i^{Ta} , and n^{Ta} signifies the number of tokens derived from X_i^{Ta} . In all cases, f refers to a neural network function outputting gene expression levels.

3.2. TRIPLEX

The overall workflow of our method is illustrated in Figure 1. Initially, we process the global view, derived from all spot images of a WSI, through a global encoder to produce global tokens. Although these global tokens capture

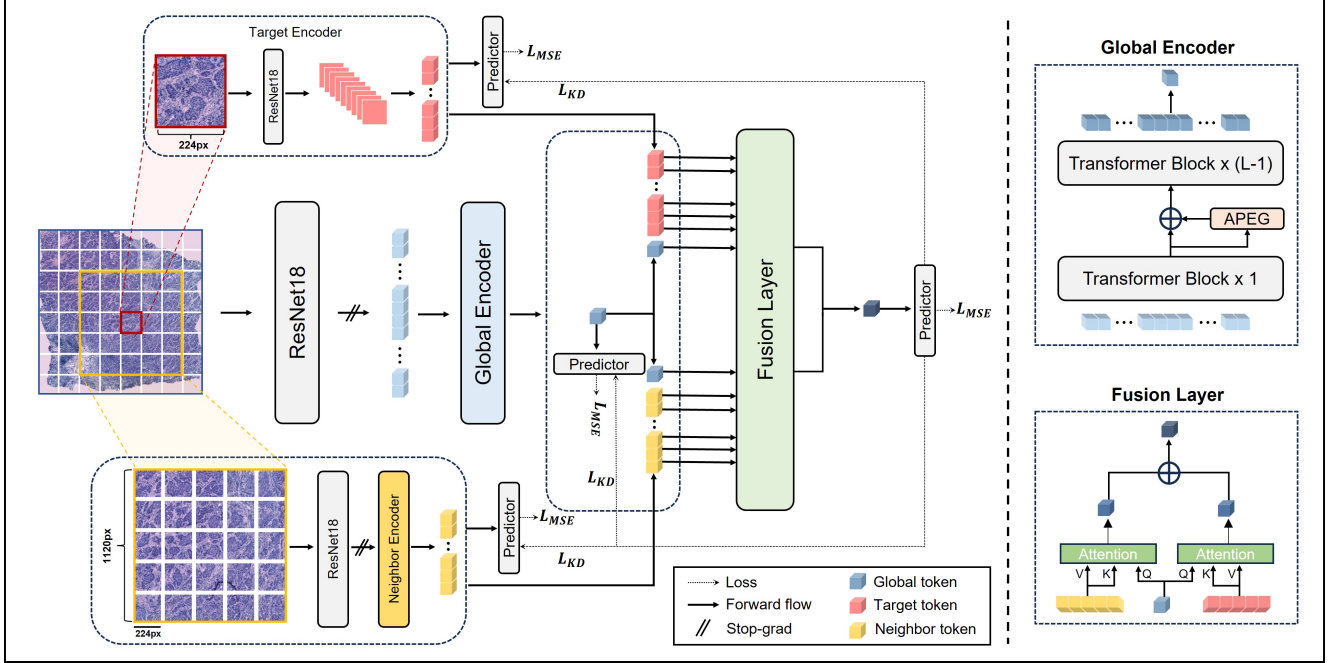


Figure 1. Schematic representation of the TRIPLEX. The global encoder processes the global view, while separate encoders handle the target spot image and neighbor view. A fusion layer, incorporated with fusion loss, facilitates the effective integration of these tokens to predict gene expression levels.

the macroscopic spatial distribution and inter-spot correlations, they might not adequately represent the detailed information specific to each target spot. To address this, we independently encode the target spot image and its neighboring views, generating tokens that encapsulate finer details. These are then integrated with the global tokens via fusion layers, enriching the global representation with specific target-related information. Additionally, considering the diverse contextual information provided by different input sources, we have implemented a fusion loss mechanism, inspired by knowledge distillation, to enhance the efficacy of the fusion process. Detailed descriptions of the individual components involved in our proposed model will be provided in the following sections.

3.3. Embedding Global Information

Processing all 224x224-sized images of spots starting from patch embedding is computationally intensive since the number of spots typically ranges from hundreds to thousands. To address this, we employ a feature extraction strategy commonly used in the MIL approach for WSIs. Specifically, we utilize a ResNet18 model pretrained on large-scale histology images for feature extraction [6]. The features thus obtained serve as input to the global encoder, which generates global tokens. This encoder comprises a series of transformer blocks and a Position Encoding Generator (PEG) [5], adept at encoding positional information for a

variable number of spots. Given that PEG was originally designed for natural images with regular, square shapes, we modify it for our specific use-case where the image shape is irregular. Our adaptation, termed the Atypical Position Encoding Generator (APEG), imbues tokens with absolute positional information pertinent to all spots in a WSI. This modification is crucial for effectively capturing the spatial distribution within the WSI.

Atypical Position Encoding Generator (APEG) In our APEG framework, after the initial transformation of tokens through one transformer block, we employ a technique to re-establish their relative positional context. This involves reshaping the global tokens $z^{Gl} \in \mathbb{R}^{n \times d}$ into a spatial format $\hat{z}^{Gl} \in \mathbb{R}^{h \times w \times d}$. Here, h and w represent the maximum values of the x - and y -coordinates, respectively. During this process, any voids in the spatial arrangement are temporarily filled with zeros. Subsequently, we apply convolutional layers to these reshaped tokens. This step includes refilling the areas that were previously vacant with zeros. After this convolutional processing, we revert the tokens back to their original format. This method enables us to effectively incorporate the relative spatial information of the tokens, enhancing the positional encoding within the WSI framework.

3.4. Embedding Target/Neighbor Information

We independently encode the image of the target spot and its neighboring view, generating tokens for both target

and neighbor that are rich in contextual information.

Embedding target information To encode the target spot image, we utilize a ResNet18 architecture, excluding the global average pooling and the fully-connected layer. This encoder processes each 224x224 target spot image by embedding it into 49 distinct features, with each feature having a dimension of 512. Notably, while this instance of ResNet18 is initialized with the same weights as before, it undergoes unique updates during the training process, ensuring tailored feature extraction for each target spot image.

Embedding neighbor information For the neighbor view, we use the surrounding 1120x1120 image of each target spot. This choice is based on images directly adjacent to the target spot, rather than a group of neighboring spot images. This approach addresses the issue of non-uniform alignment and spacing between spots in ST data (refer to supplementary Figure 1 for details). In cases where a target spot is at the edge of the slide, zero padding is applied to maintain the required image size. This 1120x1120 neighbor view is then embedded into twenty-five 512-dimensional feature vectors using the ResNet18. These vectors then serve as input for the neighbor encoder. The embedding process within the neighbor encoder involves a sequence of self-attention blocks, each integrated with relative position encoding [23]. Although the ResNet18 weights used for feature extraction remain fixed, the weights of the neighbor encoder are dynamically updated during training. Further details about the neighbor encoder are provided in the supplementary material.

3.5. Integrating global, neighbor, and target information

To achieve an effective exchange of information and enhanced contextual understanding between different token types, we implement cross-attention layers. In this setup, the global token acts as the query (Q), with the target and neighboring tokens within the WSI forming the key (K) and value (V) pairs. These Q, K, and V components are utilized in a dot-product attention mechanism, which is crucial for assessing the relative importance of each target and neighbor token in comparison to the global token. This process is instrumental in developing a comprehensive understanding of the local and global contexts at each spot, thereby augmenting the model’s performance in complex gene expression prediction tasks. The integration of these insights is realized by summing the resultant tokens from the cross-attention layers:

$$z_i^{GT} = \text{CrossAttn}(z_i^{Gl}, z_i^{Ta}) \in \mathbb{R}^d \quad (11)$$

$$z_i^{GN} = \text{CrossAttn}(z_i^{Gl}, z_i^{Ne}) \in \mathbb{R}^d \quad (12)$$

$$z_i^{GTN} = \text{Sum}(z_i^{GT}, z_i^{GN}) \in \mathbb{R}^d \quad (13)$$

In these equations, z_i^{GT} and z_i^{GN} represent the cross-attention results for the target and neighbor tokens, respectively, relative to the global token. z_i^{GTN} is the summation of these two tokens, which is used to estimate the gene expression levels. During the process, information exchange between the neighbor token and the target token occurs only through the global tokens. In this manner, we can efficiently integrate the three pieces of information at a minimal additional cost.

Fusion Loss and Objective Function To optimize the integration of information from multiple tokens, we have introduced a fusion loss mechanism. This approach capitalizes on the rich, gene expression-relevant information inherent in the fusion token, which synthesizes data from the target, neighbor, and global tokens. By transferring knowledge from this fusion token to other individual tokens, we significantly enhance the model’s predictive accuracy for gene expression levels. In practice, fully-connected layers, tasked with predicting gene expression levels, are attached to each of the target, neighbor, and global tokens, with average pooling applied beforehand to the target and neighbor tokens. The optimization process involves two key components: 1) minimizing Mean Squared Error (MSE) loss between individual predictors’ outputs and ground-truth values, 2) reducing the MSE loss between each predictor’s output and the ‘soft target,’ which are the predictions derived from the fusion token.

The loss for a given j_{th} token (target, neighbor, or global) is computed as follows:

$$L^j = (1 - \alpha) \frac{1}{m} \sum_{k=1}^m \|q_k^j - y_k\|_2^2 + \alpha \frac{1}{m} \sum_{k=1}^m \|q_k^j - q_k^F\|_2^2, \quad (14)$$

where q_k^j is the prediction for the k_{th} gene by the j_{th} token, q_k^F is the fusion token’s prediction for the k_{th} gene, and α is a hyperparameter balancing the two aspects of the loss.

For the fusion token, we calculate the MSE loss in relation to the actual labels:

$$L^F = \frac{1}{m} \sum_{k=1}^m \|q_k^F - y_k\|_2^2 \quad (15)$$

Ultimately, we optimize the following object function:

$$L = \sum_j^3 L^j + L^F. \quad (16)$$

4. Experiments

In this section, we outline the specifics of the ST data employed in our model’s training, detail our experimental setup and evaluation metrics, and provide implementation details. For more details on experiment settings, please refer to Section 2 of the supplementary material.

Source	Model	BC1			BC2			SCC		
		MSE	PCC(M)	PCC(H)	MSE	PCC(M)	PCC(H)	MSE	PCC(M)	PCC(H)
Local	ST-Net [8]	0.260 ± 0.04	0.194 ± 0.11	0.345 ± 0.16	0.209 ± 0.02	0.116 ± 0.06	0.223 ± 0.10	0.294 ± 0.07	0.274 ± 0.08	0.382 ± 0.08
	EGN [28]	0.241 ± 0.06	0.197 ± 0.11	0.328 ± 0.17	0.192 ± 0.02	0.111 ± 0.05	0.203 ± 0.09	0.281 ± 0.08	0.281 ± 0.06	0.388 ± 0.06
	BLEEP [26]	0.277 ± 0.05	0.151 ± 0.11	0.277 ± 0.16	0.235 ± 0.02	0.095 ± 0.05	0.193 ± 0.10	0.297 ± 0.05	0.269 ± 0.07	0.396 ± 0.08
	TEM	0.252 ± 0.04	0.208 ± 0.11	0.365 ± 0.15	0.190 ± 0.02	0.119 ± 0.06	0.227 ± 0.10	0.290 ± 0.06	0.296 ± 0.07	0.402 ± 0.08
	NEM	0.278 ± 0.08	0.255 ± 0.13	0.424 ± 0.18	0.193 ± 0.03	0.152 ± 0.05	0.277 ± 0.09	0.373 ± 0.14	0.308 ± 0.05	0.444 ± 0.06
Global	HistoGene [20]	0.314 ± 0.09	0.168 ± 0.12	0.302 ± 0.19	0.194 ± 0.05	0.100 ± 0.05	0.219 ± 0.12	0.270 ± 0.09	0.133 ± 0.06	0.261 ± 0.13
	GEM	0.253 ± 0.06	0.295 ± 0.14	0.491 ± 0.17	0.221 ± 0.03	0.193 ± 0.06	0.341 ± 0.08	0.317 ± 0.16	0.276 ± 0.09	0.392 ± 0.08
Local+Global	Hist2ST [29]	0.285 ± 0.08	0.118 ± 0.10	0.248 ± 0.17	0.181 ± 0.02	0.044 ± 0.02	0.099 ± 0.03	1.291 ± 0.65	0.004 ± 0.01	0.053 ± 0.01
Multiple	TRIPLEX	0.228 ± 0.07	0.314 ± 0.14	0.497 ± 0.17	0.202 ± 0.02	0.206 ± 0.07	0.352 ± 0.10	0.268 ± 0.09	0.374 ± 0.07	0.490 ± 0.07

Table 1. Cross validation result on each ST dataset. PCC(M) and PCC(H) denote the mean PCC for all genes and the mean PCC for highly predictive genes, respectively. The mean and standard deviation of cross-validation results are displayed. MAEs are excluded due to space limitations and can be found in the supplementary.

ST dataset Spatial Transcriptomics (ST) data is characterized by its compilation of numerous spot images within a single slide, each accompanied by corresponding gene expression values. A typical ST dataset contains several hundred spatially resolved spots, with each spot representing the expression values of around 20,000 genes. Visium, the next iteration of ST data, expands this scope to include thousands of spots, each still characterized by the expressions of a similar number of genes. For our internal validation, we utilize two breast cancer ST datasets [1, 8] and one skin cancer ST dataset [11]. External validation is conducted using Visium data from three independent breast cancer patients. We refer to the breast cancer ST dataset from [1] as the BC1 dataset, the one from [8] as the BC2 dataset, and the skin cancer dataset from [11] as the SCC dataset.

Experiment Setup and Evaluation Metrics To mitigate potential overfitting due to the limited size of our datasets, we employ cross-validation for model performance evaluation across the three ST datasets. Consistent with our earlier mention, we ensure that samples from the same patients are exclusively allocated to either the training or the test dataset, avoiding any overlap. Specifically, we adopt a leave-one-patient-out cross-validation approach for the BC1 dataset (n sample=36, n patient=8) and the SCC dataset (n sample=12, n patient=4), using samples from a single patient for sequential validation. For the BC2 dataset, which has a larger sample size (n sample=68, n patient=23), we conduct 8-fold cross-validation, with careful consideration to keep samples from the same patient within the same data partition. To extend our model’s evaluation to independent datasets, we use three breast cancer Visium datasets from 10x Genomics, training on the BC1 dataset and testing on each Visium dataset. Our evaluation metrics include the Pearson correlation coefficient (PCC), mean squared error (MSE), and mean absolute error (MAE). PCC is computed for each gene across all spots in a sample, and we report both the mean PCC for all genes (PCC(M)) and the mean PCC for highly predictive genes (PCC(H)). The highly predictive genes are identified through a cross-validation ranking process, where the top 50 genes are determined based

on their average rank across all folds.

Implementation Details For preprocessing, we crop each spot image to 224x224 pixels using the center coordinates of each spot. Neighbor views are obtained by capturing a 1120x1120 image centered on the spot, which is then subdivided into 25 equal-sized sub-images. We select 250 genes for each dataset following the criteria in [8]. The gene expression values are normalized by dividing by the sum of expressions in each spot, followed by a log transformation. To mitigate experimental noise, we adopt the smoothing approach from [8], averaging the gene expression values of each spot with those of its adjacent neighbors. Our model is optimized using the Adam optimizer [14] with an initial learning rate of 0.0001. The learning rate is adjusted dynamically using a Step LR scheduler, with a step size of 50 and a decay rate of 0.9. During training, we use a batch size of 128. In testing, the model’s performance is evaluated on all spots in each WSI per batch, and we report the mean of all validation performances.

4.1. Cross-validation Performance of TRIPLEX

We conduct cross-validation using the three ST datasets to assess TRIPLEX’s performance in comparison to baseline models. The total counts of spot images in the BC1, BC2, and SCC datasets are 13,620, 68,050, and 23,205, respectively.

Baselines Our model’s performance was benchmarked against existing models, including 1) local-based models (ST-Net[8], EGN[28], BLEEP[26]) and 2) global-based models (HisToGene[20], Hist2ST[29]). For a consistent evaluation, the same ResNet18 used in TRIPLEX was applied as the feature extractor in EGN and the image encoder in BLEEP. We also compared TRIPLEX with simpler models focusing on single information types: Target Encoding Model (TEM), Neighbor Encoding Model (NEM), and Global Encoding Model (GEM). Implementation details for all baseline models are available in Section 2 of the supplementary material.

Result Comparison As shown in Table 1, TRIPLEX outperforms all previous models and demonstrates superior

Source	Model	10X Visium-1				10X Visium-2				10X Visium-3			
		MSE	MAE	PCC(M)	PCC(H)	MSE	MAE	PCC(M)	PCC(H)	MSE	MAE	PCC(M)	PCC(H)
Local	ST-Net [8]	0.423	0.505	-0.026	-0.000	0.395	0.492	0.091	0.193	0.424	0.508	-0.032	0.008
	EGN [28]	0.421	0.512	0.003	0.024	0.328	0.443	0.102	0.157	0.303	0.425	0.106	0.220
	BLEEP [26]	0.367	0.470	0.106	0.221	0.289	0.406	0.104	0.260	0.298	0.415	0.114	0.229
	TEM	0.339	0.453	0.024	0.093	0.278	0.402	0.106	0.218	0.290	0.412	0.078	0.193
	NEM	0.444	0.515	0.089	0.259	0.391	0.482	0.105	0.290	0.393	0.483	0.036	0.175
Global	GEM	0.392	0.494	0.132	0.269	0.397	0.482	0.056	0.166	0.394	0.488	0.082	0.191
Multiple	TRIPLEX	0.351	0.464	0.136	0.241	0.282	0.407	0.155	0.356	0.285	0.410	0.118	0.282

Table 2. Generalization performance comparison between other models and ours by PCC(M), PCC(H), MSE, and MAE.

performance over the individual modules within TRIPLEX across most evaluation metrics. Notably, GEM demonstrates substantial effectiveness, underscoring the crucial role of global interactions in accurately predicting gene expression. However, TRIPLEX, which integrates the target and neighbor view information with the global view, yields the most notable improvement. Compared to EGN, one of the best-performing existing models, TRIPLEX achieves substantial increases in PCC(M) and PCC(H) across all datasets. Specifically, in the BC1 dataset, there is an improvement of 0.117 in PCC(M) and 0.169 in PCC(H); in the BC2 dataset, an increase of 0.095 in PCC(M) and 0.149 in PCC(H); and in the SCC dataset, a rise of 0.093 in PCC(M) and 0.102 in PCC(H). Despite integrating three types of information, TRIPLEX maintains a parameter count comparable to other top-performing models (see supplementary table 1). The variances between our results and those reported in the original works of the baseline models can be attributed to differences in 1) cross-validation strategies, 2) normalization methods, and 3) metric calculations. For an in-depth explanation of these differences, refer to Section 3 of the supplementary material.

4.2. Generalization performance of TRIPLEX

For external validation, we preprocess Visium data similarly to the ST data and evaluate model performance on three individual Visium samples.

Result comparison In this set of experiments, TRIPLEX is benchmarked against the three best-performing models identified in prior tests. According to Table 2, TRIPLEX shows robust performance on unseen Visium data, which is formatted differently from the training data. It outperforms existing models (ST-Net, EGN, BLEEP) across MSE, MAE, PCC (M), and PCC (H). Notably, TRIPLEX consistently achieves impressive PCC (H) scores (average 0.293), indicating its potential applicability in clinical settings.

4.3. Visualization of Cancer Marker Genes

Among the highly predictive genes identified by our model are known breast cancer markers such as CLDN4 and GNAS [12, 19], which could aid pathological diagnosis. In the cross-validation for the GNAS gene, TRIPLEX

significantly outperforms all existing models in both BC1 and BC2 datasets. For instance, in the BC1 dataset, the PCCs are 0.359 (HisToGene), 0.286 (Hist2ST), 0.411 (ST-Net), 0.374 (EGN), 0.338 (BLEEP), and 0.583 (TRIPLEX). Similarly, in the BC2 dataset, the PCCs are 0.282 (HisToGene), 0.138 (Hist2ST), 0.371 (ST-Net), 0.341 (EGN), 0.0669 (BLEEP), and 0.554 (TRIPLEX). We further provide visualizations of the predicted values for GNAS alongside its ground truth values for each dataset. As depicted in Figure 2, we notice that TRIPLEX not only exhibits a higher PCC with the actual ground truth values but also demonstrates a greater visual congruence with the actual gene expression patterns. Consequently, it appears to be more effective in assisting pathologists in clinical diagnostics. More visualizations are available in the supplementary material.

4.4. Ablation study

In this section, we demonstrate the contributions of each method to gene expression prediction through an ablation study of our proposed model. Specifically, we aim to observe the contributions of the three modules in our model and investigate the extent to which the position encoding generator and our proposed fusion approach contribute to gene expression prediction. Here we only present experimental results on the SCC dataset, but we have observed similar outcomes on other datasets as well. Additional experiment results can be found in supplementary material.

Individual Modules We assess how the fusion of information from TEM, NEM, and GEM enhances gene expression prediction accuracy. According to Table 3, it is clear that incorporating features from all three modules achieves the best performance. Notably, the exclusion of the NEM module results in a substantial decrease in PCC(M), highlighting the critical influence of neighboring interactions in gene expression prediction in TRIPLEX. While the absence of the GEM module does not significantly impact performance metrics such as MSE and MAE in this specific dataset, ablation studies conducted on alternative datasets reveal the significant contribution of global interactions to the accuracy of gene expression level predictions.

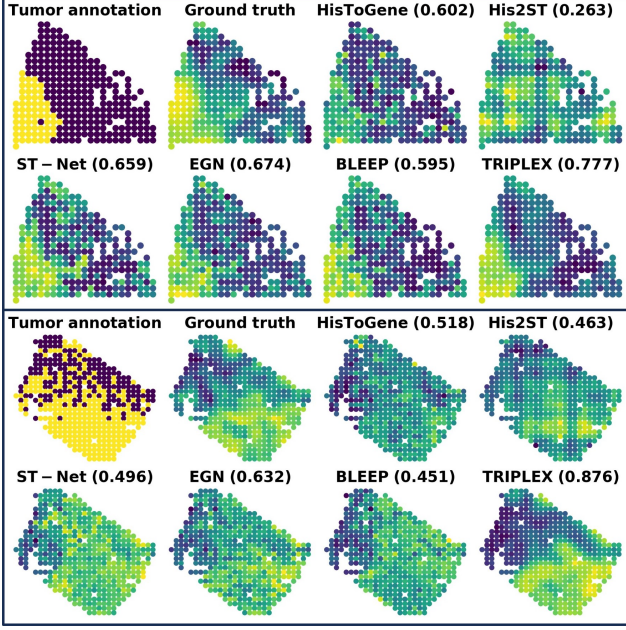


Figure 2. The visualization includes tumor region annotations by pathologists, ground truth for GNAS expression levels, and predicted GNAS expression levels from HisToGene, Hist2ST, ST-Net, EGN, BLEEP, and TRIPLEX, in samples from datasets BC1 and BC2. The PCC between the ground truth and predicted values is displayed for each model.

	MSE	MAE	PCC(M)	PCC(T)
w/o TEM	0.289	0.419	0.352	0.471
w/o NEM	0.271	0.408	0.330	0.439
w/o GEM	0.263	0.402	0.358	0.481
TRIPLEX	0.268	0.404	0.374	0.490

Table 3. Ablation study for the individual modules

Position Encoding Generator (PEG) We further compare our APEG to different methods of positional encoding, including (1) without PEG and (2) with PEG [5], as shown in Table 4. In the conventional PEG implementation, we add zero padding to the global token to make the number of tokens a perfect square, followed by the standard squaring procedure. The results indicate that APEG not only outperforms the model without any PEG but also surpasses the conventional PEG. This implies that APEG more effectively encodes spatial distribution within the global tokens, enhancing the overall model performance.

Fusion Method and Fusion Loss We also investigate the impact of different fusion methods for multi-resolution features, comparing our model’s fusion layer with traditional feature fusion techniques like summation, concatenation, and attentional pooling [10]. Additionally, we assess the

	MSE	MAE	PCC(M)	PCC(T)
w/o PEG	0.280	0.413	0.360	0.480
PEG [5]	0.276	0.411	0.364	0.479
APEG	0.268	0.404	0.374	0.490

Table 4. Ablation study PEG

contribution of fusion loss to the model’s performance. As shown in Table 5, integrating multi-resolution features using our fusion layer yields superior results compared to these conventional techniques. Moreover, the marked improvement in performance when incorporating fusion loss indicates its effectiveness in integrating the three types of features. This result highlights the significance of our fusion approach in achieving high accuracy in gene expression prediction.

Fusion method	MSE	MAE	PCC(M)	PCC(T)
Summation	0.297	0.425	0.341	0.464
Concatenation	0.293	0.422	0.348	0.474
Attentional pooling	0.293	0.423	0.353	0.473
fusion layer	0.268	0.404	0.374	0.490
Fusion loss	MSE	MAE	PCC(M)	PCC(T)
w/o fusion loss	0.292	0.423	0.358	0.469
w/ fusion loss	0.268	0.404	0.374	0.490

Table 5. Ablation studies for fusion method and fusion loss

5. Conclusion

We demonstrate a novel approach for predicting spatial gene expression patterns from WSIs. By incorporating multiple sources of information utilizing various types of transformer and our proposed fusion method, TRIPLEX achieves superior performance compared to all existing approaches in both internal and external evaluations. TRIPLEX has the potential to improve the accuracy and robustness of the predictions for spatial gene expression distribution, paving the way for new discoveries at the interface of WSIs and sequencing.

Acknowledgments This work was partly supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2019-0-00421, AI Graduate School Support Program(Sungkyunkwan University)) and the Samsung Research Funding & Incubation Center of Samsung Electronics under Project SRFC-MA2102-05. This work is a study partly supported by domestic scholarships funded by the Kwanjeong Educational Foundation (KEF1464).

References

- [1] Alma Andersson, Ludvig Larsson, Linnea Stenbeck, Fredrik Salmén, Anna Ehinger, Sunny Wu, Ghamdan Al-Eryani, Daniel Roden, Alex Swarbrick, Åke Borg, et al. Spatial deconvolution of her2-positive breast tumors reveals novel intercellular relationships. *bioRxiv*, 2020. 2, 6
- [2] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. 6
- [3] Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Mirafior, Vitor Werneck Krauss Silva, Klaus J Busam, Edi Brogi, Victor E Reuter, David S Klimstra, and Thomas J Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8):1301–1309, 2019. 3
- [4] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16144–16155, 2022. 3
- [5] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 4, 8
- [6] Ozan Ciga, Tony Xu, and Anne Louise Martel. Self supervised contrastive learning for digital histopathology. *Machine Learning with Applications*, 7:100198, 2022. 4, 2, 3
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3
- [8] Bryan He, Ludvig Bergenstråhle, Linnea Stenbeck, Abubakar Abid, Alma Andersson, Åke Borg, Jonas Maaskola, Joakim Lundberg, and James Zou. Integrating spatial gene expression and breast tumour morphology via deep learning. *Nature biomedical engineering*, 4(8): 827–834, 2020. 1, 2, 3, 6, 7, 4, 5
- [9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2, 4
- [10] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. 8, 5
- [11] Andrew L Ji, Adam J Rubin, Kim Thrane, Sizun Jiang, David L Reynolds, Robin M Meyers, Margaret G Guo, Benson M George, Annelie Mollbrink, Joseph Bergenstråhle, et al. Multimodal analysis of composition and spatial architecture in human squamous cell carcinoma. *Cell*, 182(2): 497–514, 2020. 2, 6
- [12] X Jin, L Zhu, Z Cui, J Tang, M Xie, and G Ren. Elevated expression of gnas promotes breast cancer cell proliferation and migration via the pi3k/akt/snail1/e-cadherin axis. *Clinical and Translational Oncology*, 21:1207–1219, 2019. 7
- [13] Fahdi Kanavati, Gouji Toyokawa, Seiya Momosaki, Michael Rambeau, Yuka Kozuma, Fumihiko Shoji, Koji Yamazaki, Sadanori Takeo, Osamu Iizuka, and Masayuki Tsuneki. Weakly-supervised learning for lung carcinoma classification using deep learning. *Scientific reports*, 10(1):1–11, 2020. 3
- [14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2
- [16] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019. 4
- [17] Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2021. 3
- [18] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6):555–570, 2021. 3
- [19] Patrice J Morin. Claudin proteins in human cancer: promising new targets for diagnosis and therapy. *Cancer research*, 65(21):9603–9606, 2005. 7
- [20] Minxing Pang, Kenong Su, and Mingyao Li. Leveraging information in spatial transcriptomics to predict super-resolution gene expression from histology images in tumors. *bioRxiv*, 2021. 1, 2, 3, 6, 4, 5, 7
- [21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3
- [22] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in Neural Information Processing Systems*, 34:2136–2147, 2021. 3
- [23] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 5
- [24] Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016. 1
- [25] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022. 2

- [26] Ronald Xie, Kuan Pang, Gary D. Bader, and Bo Wang. Spatially resolved gene expression prediction from h&e histology images via bi-modal contrastive learning, 2023. [1](#), [2](#), [3](#), [6](#), [7](#), [4](#), [5](#)
- [27] Yan Yang, Md Zakir Hossain, Tom Gedeon, and Shafin Rahman. S2fgan: semantically aware interactive sketch-to-face translation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1269–1278, 2022. [4](#)
- [28] Yan Yang, Md Zakir Hossain, Eric A Stone, and Shafin Rahman. Exemplar guided deep neural network for spatial transcriptomics analysis of gene expression prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5039–5048, 2023. [1](#), [2](#), [3](#), [6](#), [7](#), [4](#), [5](#)
- [29] Yuansong Zeng, Zhuoyi Wei, Weijiang Yu, Rui Yin, Yuchen Yuan, Bingling Li, Zhonghui Tang, Yutong Lu, and Yuedong Yang. Spatial transcriptomics prediction from histology jointly through transformer and graph neural networks. *Briefings in Bioinformatics*, 23(5):bbac297, 2022. [1](#), [2](#), [3](#), [6](#), [4](#), [5](#), [7](#)

Accurate Spatial Gene Expression Prediction by Integrating Multi-Resolution Features

Supplementary Material

F. Evaluation methodology for multi-output regression problem

In this study, our objective is to address the multi-output regression problem in n distinct spots within a Whole Slide Image (WSI). We formally define this problem as:

$$f : X \rightarrow Y \in \mathbb{R}^{n \times m}$$

where X represents the set of n input images, and Y denotes the expression levels of m genes across these n spots. To tackle this problem, we employ three evaluation metrics: the Pearson Correlation Coefficient (PCC), Mean Squared Error (MSE), and Mean Absolute Error (MAE). Our choice of these metrics is grounded in their distinct advantages. Firstly, the PCC offers insights into the linear relationship between predicted and actual target values, both in strength and direction. Secondly, the MSE is a robust measure of the average squared discrepancies between predicted and actual targets, reflecting the model's accuracy and sensitivity to errors. Lastly, the MAE provides an interpretable measure of the average absolute differences between predictions and actual targets, advantageous for its lower sensitivity to outliers compared to MSE.

We assess each model's performance on a per-slide basis. For the j_{th} gene, the PCC, MAE, and MSE are calculated as follows:

$$PCC_j = \frac{\sum_{i=1}^n (\hat{y}_{i,j} - \bar{\hat{y}}_{\cdot,j})(y_{i,j} - \bar{y}_{\cdot,j})}{\sqrt{\sum_{i=1}^n (\hat{y}_{i,j} - \bar{\hat{y}}_{\cdot,j})^2} \sqrt{\sum_{i=1}^n (y_{i,j} - \bar{y}_{\cdot,j})^2}}$$

$$MAE(Y, \hat{Y}) = \frac{1}{n \times m} \sum_{i=1}^n \sum_{j=1}^m |y_{i,j} - \hat{y}_{i,j}|$$

$$MSE(Y, \hat{Y}) = \frac{1}{n \times m} \sum_{i=1}^n \sum_{j=1}^m (y_{i,j} - \hat{y}_{i,j})^2$$

Here, $\hat{y}_{i,j}$ represents the predicted expression level of the j_{th} gene in the i_{th} spot, and m and n are numbers of genes to be predicted and spots in a WSI, respectively. For PCC, the average value across m number of genes is computed as:

$$PCC = \frac{1}{m} \sum_{j=1}^m PCC_j$$

When multiple slides are involved in testing, we calculate PCC, MAE, and MSE for each slide using the above methodology and then report the average values across all slides.

G. Method Details

G.1. Processing of input data

Target spot image For the extraction of target spot images, we employ pre-defined center coordinates to obtain images of dimensions 224x224. Subsequently, these images undergo a normalization process where pixel values are adjusted to fall within the range of 0 to 1, prior to their input into the model. During the training phase, we enhance the robustness of our model by applying image augmentation techniques. These techniques encompass random horizontal and vertical flips, and random rotations of the input images by 90 degrees.

Neighbor view In processing images sized 1,120x1,120, we commence by segmenting a centrally located 1,120x1,120 patch from the target spot image into 25 uniform sub-patches, each measuring 224x224. It is important to note that these sub-patches differ from the 25 spot images nearest to the target spot, a distinction necessitated by the non-uniform alignment of center coordinates in ST data and the observable gaps between spot images, as depicted in Figure 3.

Feature extraction is carried out using a ResNet18 model that has been pre-trained. The significant dissimilarity between histology images and conventional image types presents a challenge, as models trained on datasets like ImageNet might not be directly suitable for WSI analysis. To address this, we utilize a version of ResNet18 that has undergone training on an integrated, multi-organ dataset through self-supervised learning. This training strategy ensures the extraction of features that are robust to variations in staining and resolution, as detailed in [6]. The extracted features are then employed as inputs for the neighbor encoder.

Global view We engage all spot images contained within a WSI. It's important to clarify that the aggregation of all spot images does not represent the full extent of the WSI. Nonetheless, this comprehensive inclusion of spot images enables us to effectively map the interconnections between these images and approximate the spatial information, as illustrated in Figure 3.

For feature extraction, we apply the same methodology used in the neighbor view. This involves cropping spot images to a uniform size of 224x224 and processing them through the pre-trained ResNet18 model. The features extracted from this process are then channeled as inputs into the global encoder for further processes.

Visium data for external test We evaluate our model on spatial gene expression data from breast cancer tissues, sourced from 10x Genomics. The datasets employed are as follows:

- 10X Visium-1: Breast cancer tissue from human (v1), Spatial Gene Expression Dataset by Space Ranger v1.3.0 (2022, Jul 02).
- 10X Visium-2: Breast cancer tissue from human (v1, Section 1), Spatial Gene Expression Dataset by Space Ranger v1.1.0 (2020, Jun 12).
- 10X Visium-3: Breast cancer tissue from human (v1, Section 2), Spatial Gene Expression Dataset by Space Ranger v1.1.0 (2020, Jun 12).

This dataset represents an enhancement over the ST data used in the training phase, providing high-resolution gene expression profiles with thousands of spots per sample. We apply consistent pre-processing methods across all Visium datasets. The model, initially trained on the BC1 dataset, is subsequently applied to predict 250 genes, initially selected based on their representation in the BC1 dataset. In cases where any of the 250 genes are not present in a Visium dataset, we exclude those genes from our evaluation. This approach leads to the consistent exclusion of 7 genes across all datasets, a detail we elaborate upon in Figure 8.

G.2. Method details for Target Encoder

As detailed in the Methods section of the main text, the target encoder embeds target spot images using the pre-trained ResNet18 model [6]. This model is fine-tuned to specifically capture fine-grained, target-specific information from the target spot images. In particular, an image with dimensions of 224x224x3 undergoes a transformation into a 7x7x512 feature map after processing through all layers of the ResNet18 model. The resulting features are then reshaped into 49x512 tokens. These tokens are integrated with other tokens -neighbor and global tokens- to form the input for the fusion layer. Concurrently, a separate fully connected layer is linked to the average-pooled token of the target tokens, facilitating independent gene expression prediction. During the training phase, the weights of the target encoder are comprehensively updated to enhance the capture of fine-grained spot information.

G.3. Method details for Neighbor Encoder

The neighbor encoder is designed to embed local information surrounding the target spot. It processes features extracted from 25 images, each of size 224x224, representing the neighbor view. This approach closely aligns with the Vision Transformer (ViT) [7] architecture, incorporating self-attention mechanisms with relative position encoding and fully-connected layers applied to the input tokens. A key deviation from the standard ViT model is the exclusion of the patch embedding module for 2D images. In-

stead, we directly utilize pre-extracted features as our input values. Details regarding specific hyperparameters and their settings will be discussed in the forthcoming section, "Additional Implementation Details and Experimental Results." In a manner similar to the target layer, an additional fully-connected layer is attached to the pooled token of the neighbor tokens. This layer is also specifically tasked with the prediction of gene expression.

G.4. Method details for Global Encoder

The global encoder is composed of transformer blocks integrated with the Atypical Position Encoding Generator (APEG). The operational flow of APEG is illustrated in Figure 4.

Implementation of APEG APEG's process begins with the rearrangement of spot features based on their relative coordinates. This involves constructing a sparse matrix in the Coordinate Format (COO) using Pytorch. In this matrix, indices represent adjusted normalized coordinates, starting from a minimum value of 0, and feature values correspond to the non-zero elements of the matrix. This sparse matrix is subsequently converted into a dense format to facilitate the application of convolutional layers, after which it is restored to its original sparse format.

For the global layer, as opposed to employing a pooling operation, the approach involves retrieving tokens that correspond to each target spot. These tokens are then connected to a fully connected layer, which is tailored to independently predict gene expression.

G.5. Method details for Fusion Layer

The fusion layer is specifically designed to integrate information from the global token with corresponding neighbor and target tokens. In this process, the global token actively exchanges information with all other tokens. However, the target and neighbor tokens, which collectively form a larger set of tokens, are structured to avoid direct interaction or information exchange among themselves. In a more detailed mechanism, for each spot, a global token functions as the query, while neighbor and target tokens are utilized as key and value elements. The overall time complexity of the fusion layer is represented as $O(n^{Ta} + n^{Ne})$, where n^{Ta} and n^{Ne} denote the numbers of target and neighbor tokens, respectively. Following this interactive process, the aggregated tokens are processed through a fully connected layer to yield the final prediction output. This approach is identified as more computationally efficient than applying attention mechanisms across the complete set of tokens, which would result in time complexity of $O((n^{Ta} + n^{Ne} + 1)^2)$. Furthermore, it has shown superior performance compared to traditional feature fusion methods, a claim substantiated by our experimental results.

G.6. Implementation Details for Baselines

This subsection details the implementation nuances of our baseline models, ensuring a consistent comparison framework with our proposed TRIPLEX model. In preprocessing the input data for all baseline models, we adhere to the same steps as outlined in Section 4 in the main text, which are also employed in TRIPLEX.

ST-Net ST-Net [8] utilizes a DenseNet121 [9] model, pretrained on ImageNet, with minimal modifications (only replacing the final output layer) and is fine-tuned on ST data using transfer learning. Our implementation strictly adheres to this scheme.

HisToGene and Hist2ST These models employ spot images of size 112x112, as used in their respective studies [20, 29]. We maintain their overall architectural framework while adapting data normalization and image augmentation techniques to align with our approach. Details on hyperparameter selection can be found in Section 3.6.

EGN For EGN [28], in addition to following the preprocessing steps of TRIPLEX, we implement the method proposed by [28] to obtain k exemplars for all spots. We replace the unsupervised SF2GAN-based model from [27] with a pretrained ResNet18 model used in TRIPLEX. This decision, informed by the results in [28], ensures a fair comparison by maintaining consistent feature extraction strategies across models.

BLEEP BLEEP [26] is a bi-modal learning model designed to co-embed histology images and gene expression levels. In our implementation, we adhere to the core architecture of BLEEP, with minor hyperparameter adjustments as detailed in Section 3.6. For inference, BLEEP utilizes the top k nearest gene expression levels to predict gene expression for query spot images. Among the three methods suggested - "simple" (using the top 1 value), "simple average" (averaging the top k values), and "weighted average" (using a weighted average of the top k values) - we employ the "simple average" approach, specifically using the top-50 nearest gene expression levels. This choice was based on its superior performance in our tests. Additionally, we opted not to use Harmony [16] for batch correction of gene expression levels, as Harmony is geared towards correcting PCA embeddings rather than raw gene expression levels. Given our dataset's limited slide range (12 to 68), such batch correction might inadvertently reduce the training data's diversity, thereby increasing the risk of overfitting.

TEM, NEM, GEM We replicate the three derivative models from TRIPLEX: the Target Encoding Model (TEM), Neighbor Encoding Model (NEM), and Global Encoding Model (GEM). Each model is specialized to process distinct views: TEM utilizes the target spot image, NEM focuses on the neighbor view, and GEM deals with the global view. Their primary objective is to predict gene expression levels based on their respective input data. Consistency with TRIPLEX

is maintained in terms of architecture and hyperparameters for these models.

H. Additional implementation details and experimental results

H.1. Description of datasets

We evaluate our model on three distinct datasets: BC1 and BC2 (breast cancer datasets) and SCC (a skin cancer dataset). We focus on the top 250 genes with high gene expression levels in each dataset as labels for prediction. Furthermore, we calculate the average ranks of well-predicted genes during cross-validation to identify the top 50 "highly predictive genes." These genes are then used to compute the PCC (H). For detailed summaries of each dataset and the specific genes selected, please refer to Figures 5, 6, and 7.

H.2. Implementation Details for Experiments

Our approach is implemented using PyTorch (version 1.13.0) and pytorch-lightning (version 1.8.0), and models are trained on a Nvidia RTX A5000 GPU. We employ mixed precision training, utilizing PyTorch native Automatic Mixed Precision (AMP) for efficiency. To ensure reproducibility, the random seed is consistently set at 2021 across all experiments. The training process is capped at a maximum of 200 epochs, with an early stopping mechanism triggered if there is no improvement in the PCC(M) (MSE in case of BLEEP) after 20 epochs.

H.3. Implementation Details for Ablation Studies

We assess the impact of omitting individual components and comparing the resulting model performance with the complete TRIPLEX model. Key components of TRIPLEX include: individual modules (TEM, NEM, GEM), each predicting gene expression levels using distinct input data; the Position Encoding Generator (PEG), which infuses positional information into WSIs; and a fusion strategy designed to integrate various types of tokens effectively.

Individual Modules In our experimental setup, each module (TEM, NEM, GEM) is individually omitted while maintaining the other components as per the original TRIPLEX configuration. Notably, in scenarios where the GEM is excluded, we introduce a dimensionally equivalent, randomly initialized token in place of the global token. This approach is necessary because, without GEM, there is no medium for information exchange in the fusion layer.

Position Encoding Generator (PEG) We evaluate the significance of our Atypical PEG (APEG), which is engineered to encapsulate positional information within a WSI, on the model's ability to predict gene expression levels. This evaluation involves either removing APEG or substituting it with a traditional PEG as detailed in [5] and Section 4.4 in the main text.

Fusion Method To ascertain the efficacy of our proposed fusion layer in amalgamating different token types for the prediction of gene expression levels, we compare TRIPLEX’s performance when the fusion layer is replaced with alternative methods: element-wise summation, concatenation, and attentional pooling. In the case of attentional pooling, we dynamically compute feature weights using a neural network, subsequently deriving a weighted sum of the features, as illustrated in [10].

H.4. Computational Cost Comparison

Table 6 provides a detailed comparison of the computational costs between TRIPLEX and the baseline models, calculated using a single slide sample for each fold in each dataset. This table includes average values for Multiply-Accumulate Operations (MACs), the number of parameters for each model, and both training and testing times across all folds. It’s important to note that the training time is gauged based on the duration required to complete 10 epochs.

While TRIPLEX, with its additional inputs, has higher training time compared to other baselines, two observations particularly highlight its efficiency: 1) TRIPLEX’s feature extraction technique and integration method efficiently limit its parameters to approximately 20 million, which, while comparable to the baselines, allows TRIPLEX to still achieve state-of-the-art performance. This demonstrates the model’s ability to balance complexity with high performance. 2) Additionally, TRIPLEX shows comparable testing times to other leading models (ST-Net, EGN, BLEEP), indicating that its speed remains competitive for practical applications after training. This balance between training complexity and testing efficiency underscores the model’s practical applicability in real-world scenarios.

H.5. Comparison of MAE in the cross-validation experiments

Table 7 shows the evaluation of the MAE of the cross-validation results on ST data, which is not included in the main text due to space limitations.

H.6. Contribution of the Neighbor View size

We examine how the performance of TRIPLEX is influenced by the expansion of the neighbor view size. Here, the term “number of neighbors” refers to the count of 224x224 patches along an axis. Results are illustrated in Figure 9. When evaluated in terms of MES and PCC, it is observed that enlarging the neighbor view size does not consistently result in performance gains. In fact, as the number of neighbors increases, a decrease in performance is noted. This pattern suggests that 1,120x1,120 sized neighbor view (number of neighbors: 5) is an efficient configuration, striking a balance between capturing detailed neighboring informa-

Dataset	BC1			
	MACs(G)	# Param(M)	Training Time(s)	Testing Time(ms)
ST-Net	1002	7	244.2	201.7
HistoGene	52	153	291.5	2.55
Hist2ST	110	107	254.9	7.62
EGN	1823	162	407.6	52.67
BLEEP	631	11	119.7	109.5
TRIPLEX	657	22	410.9	53.21
Dataset	BC2			
	MACs(G)	# Param(M)	Training Time(s)	Testing Time(ms)
ST-Net	1465	7	508.7	295.0
HistoGene	77	153	329.5	2.33
Hist2ST	20	37	193.0	5.54
EGN	865	39	722.1	31.58
BLEEP	923	11	207.2	72.93
TRIPLEX	960	20	1117.3	76.27
Dataset	SCC			
	MACs(G)	# Param(M)	Training Time(s)	Testing Time(ms)
ST-Net	1928	7	153.9	385.6
HistoGene	18	27	93.70	3.12
Hist2ST	13	14	57.5	5.37
EGN	5563	223	368.8	157.83
BLEEP	1215	11	86.4	95.41
TRIPLEX	1263	19	340.3	99.90

Table 6. Computational cost comparison

Source	Model	BC1 MAE	BC2 MAE	SCC MAE
Local	ST-Net [8]	0.389 ± 0.03	0.349 ± 0.02	0.428 ± 0.05
	EGN [28]	0.377 ± 0.04	0.337 ± 0.02	0.418 ± 0.06
	BLEEP [26]	0.401 ± 0.03	0.369 ± 0.02	0.430 ± 0.04
	TEM	0.385 ± 0.03	0.336 ± 0.02	0.433 ± 0.05
	NEM	0.403 ± 0.06	0.375 ± 0.03	0.481 ± 0.10
Global	HistoGene [20]	0.428 ± 0.07	0.335 ± 0.04	0.415 ± 0.07
	Hist2ST [29]	0.413 ± 0.07	0.333 ± 0.02	0.924 ± 0.29
	GEM	0.383 ± 0.05	0.352 ± 0.02	0.434 ± 0.12
Multiple	TRIPLEX	0.362 ± 0.05	0.343 ± 0.02	0.404 ± 0.07

Table 7. Cross validation result on each ST dataset. The mean and standard deviation of MAE from the cross-validation results are displayed.

tion relevant to the target and maintaining manageable computational costs.

H.7. Performance Discrepancy Between Our Experimental Results and Existing Implementations

Our experimental results show notable deviations from those reported in the original publications of the baseline models. We attribute this discrepancy primarily to three factors, as detailed in Section 4.1 in the main text: 1) the use of an alternative cross-validation strategy, 2) a different approach to normalization, and 3) variations in how metrics are calculated.

Specifically, the performance gap observed for HistoGene and Hist2ST can be largely traced back to the first factor. In the original studies, these models are tested on the BC1 and SCC datasets using Leave-one-out-cross-validation (LOOCV), where each sample is treated independently. This approach potentially skews the evaluation,

as it allows for the possibility of using replicates from the same sample in both training and testing phases. In contrast, our study employs Leave-one-patient-out-cross-validation (LOPCV), which we believe offers a stricter and more realistic assessment of model performance by ensuring no overlap between training and testing sets for a given patient. Table 8 compares the results obtained from these two cross-validation methods. As hypothesized, the change to LOPCV significantly affects the performance of both models, reinforcing our assertion about the importance of the rigorous cross-validation approach in model evaluation.

Model	HisToGene			
	MSE	MAE	PCC(M)	PCC(H)
LOPCV (ours)	0.314	0.428	0.168	0.302
LOOCV [20, 29]	0.223	0.364	0.186	0.315
Model	HisT2ST			
	MSE	MAE	PCC(M)	PCC(H)
LOPCV (ours)	0.285	0.413	0.118	0.248
LOOCV [20, 29]	0.163	0.313	0.251	0.416

Table 8. Result comparison for different cross-validation method in BC1 dataset

In the case of EGN, factors 2) and 3) — different normalization methods and variations in metric calculations — significantly contribute to the performance gap observed. Our approach to normalization for ST data involves dividing each gene’s count by the total expression count of each spot and applying a log transformation, complemented by the expression smoothing method proposed by ST-Net [8]. Conversely, EGN’s methodology adds a pseudo count of 1, applies a log transformation, and then conducts min-max normalization using each gene’s max and min count values from all training data. We hypothesize that EGN’s approach may inadvertently amplify technical variations due to batch effects, diminishing the focus on biologically relevant variations, which is central to our study. Therefore, we opt for a normalization method that we believe better preserves these biological variations. Regarding evaluation methods, as detailed in Section 1, our approach involves predicting gene expression levels for each slide and averaging the outcomes across multiple slides. In contrast, EGN evaluates all spots of the validation data in a single assessment. Given the clinical context where a WSI is typically provided for gene expression prediction, we find our method more aligned with real-world applications. To further explore these methodological differences, we conduct experiments substituting our methods with those of EGN (s/ norm, s/ eval, and both combined as s/ norm&eval). The experiment results, depicted in Table 9, confirm that the original results reported in EGN’s literature are reproducible when adopting their specific normalization and evaluation strategies.

Model	EGN			
	MSE	MAE	PCC(M)	PCC(H)
Ours	0.1923	0.3366	0.1112	0.2025
s/ eval	0.1930	0.3365	0.1494	0.3056
s/ norm	0.0005	0.0173	0.1595	0.2193
s/ eval&norm [28]	0.0003	0.0134	0.2003	0.3011

Table 9. Result comparison for different evaluation and normalization method in BC2 dataset.

In summary, our findings demonstrate that the discrepancies observed between our experimental results and those reported in existing literature arise primarily from differences in methodological approaches, particularly in cross-validation, normalization, and evaluation metrics, rather than from a lack of extensive hyperparameter tuning. These results highlight the critical impact of methodological choices in computational biology and the need for meticulous methodological reporting to ensure accurate comparisons and reproducibility.

H.8. Additional Ablation Studies

We conduct further ablation studies on the BC1, BC2, and Visium datasets, with the results detailed in Tables 10, 11, and 12 for each dataset respectively. In these studies, we examine the impact of different components of our model to understand their individual contributions to performance. A consistent trend observed across all datasets, aligning with findings from the SCC dataset, is the pronounced significance of the GEM. The GEM, central to our model’s architecture, has shown to be particularly influential in enhancing performance.

Dataset	BC1			
	MSE	MAE	PCC(M)	PCC(T)
w/o TEM	0.229	0.363	0.315	0.501
w/o NEM	0.240	0.372	0.295	0.478
w/o GEM	0.228	0.362	0.266	0.448
w/o PEG	0.227	0.363	0.294	0.466
PEG	0.230	0.365	0.304	0.485
Summation	0.241	0.375	0.293	0.475
Concatenation	0.239	0.372	0.297	0.484
Attentional fusion	0.237	0.370	0.311	0.502
w/o fusion loss	0.246	0.377	0.295	0.481
Ours	0.228	0.362	0.314	0.497

Table 10. Ablation studies in BC1 dataset

H.9. Detailed Hyperparameter Settings

In our study, hyperparameter tuning for each dataset is meticulously conducted using the WanDB platform [2]. We

Dataset	BC2			
	MSE	MAE	PCC(M)	PCC(T)
w/o TEM	0.203	0.346	0.208	0.356
w/o NEM	0.202	0.344	0.199	0.347
w/o GEM	0.193	0.336	0.159	0.291
w/o PEG	0.192	0.335	0.194	0.341
PEG	0.196	0.338	0.201	0.350
Summation	0.203	0.346	0.186	0.335
Concatenation	0.205	0.346	0.190	0.337
Attentional fusion	0.198	0.340	0.198	0.354
w/o fusion loss	0.211	0.349	0.203	0.355
Ours	0.202	0.343	0.206	0.352

Table 11. Ablation studies in BC2 dataset

Dataset	10X Visium			
	MSE	MAE	PCC(M)	PCC(T)
w/o TEM	0.338	0.453	0.099	0.250
w/o NEM	0.322	0.443	0.107	0.238
w/o GEM	0.325	0.439	0.087	0.237
w/o PEG	0.339	0.455	0.087	0.264
PEG	0.371	0.475	0.109	0.248
Summation	0.342	0.451	0.106	0.225
Concatenation	0.332	0.446	0.089	0.232
Attentional fusion	0.327	0.447	0.110	0.267
w/o fusion loss	0.328	0.442	0.050	0.206
Ours	0.306	0.427	0.136	0.293

Table 12. Ablation studies in Visium dataset

set the range of hyperparameters based on the defaults reported in relevant literature, as shown in Table 13. For each model and dataset combination, we undertake a minimum of 100 experiments to determine the optimal settings.

The hyperparameters for each baseline model are detailed in their respective publications [20, 26, 28, 29]. Regarding TRIPLEX, 'depth1', 'depth2', and 'depth3' refer to the depths of the transformer blocks in the Fusion Layer, Global Encoder, and Neighbor Encoder, respectively. 'num_heads' denotes the number of heads in the multi-head self-attention mechanism of each transformer block, the 'mlp_ratio' is the ratio of the MLP dimension to the embedding dimension within the transformer's Feed-Forward network, and 'dropout' represents the dropout probability in Transformer block. These hyperparameters are fine-tuned to maximize the PCC(M). The final hyperparameters, as determined through our extensive experiments, are presented in Table 14.

I. Additional Visualizations

In this section, we present additional visualizations focusing on the spatial expression distribution prediction of the

Model	HisToGene		
Parameter	Distribution	Min/Values	Max
n_layers	int_uniform	2	8
dim	categorical	512,1024,2048	
num_heads	categorical	4,8,16,32	
dropout	categorical	0.1,0.2,0.3,0.4	
Model	HisT2ST		
Parameter	Distribution	Min/Values	Max
depth1	int_uniform	1	4
depth2	int_uniform	2	8
depth3	int_uniform	1	4
heads	categorical	4,8,16	
channel	categorical	16,32,64,128	
bake	categorical	3,5,7	
kernel_size	categorical	3,5,7	
Model	EGN		
Parameter	Distribution	Min/Values	Max
dim	categorical	512,1024,2048	
mlp_dim	categorical	1024,2048,4096	
depth	categorical	2,4,6,8	
heads	categorical	4,8,16	
bhead	categorical	4,8,16	
bdim	categorical	32,64,128	
Model	BLEEP		
Parameter	Distribution	Values	
projection_dim	categorical	128,256,512,1024,2048	
dropout	categorical	0.1,0.15,0.2,0.25,0.3,0.35,0.4	
Model	TRIPLEX		
Parameter	Distribution	Min/Values	Max
depth1	int_uniform	1	4
depth2	int_uniform	2	4
depth3	int_uniform	1	4
dropout	categorical	0.1,0.2,0.3,0.4	
mlp_ratio	categorical	1,2,4	
num_heads	categorical	4,8,16	

Table 13. Hyperparameters to be tuned

GNAS gene, as shown in Figures 10 to 13. These visualizations include four additional samples from the BC1 dataset and 20 samples from the BC2 dataset. In analyzing these visualizations, we observe a high degree of consistency between the GNAS expression distribution and the annotations provided by pathologists. Notably, the predictions made by TRIPLEX demonstrate a markedly high accuracy, as quantitatively assessed against benchmark metrics, and align closely with tumor annotations. This consistency is particularly evident when compared to the predictions from other baseline models [8, 20, 26, 28, 29].

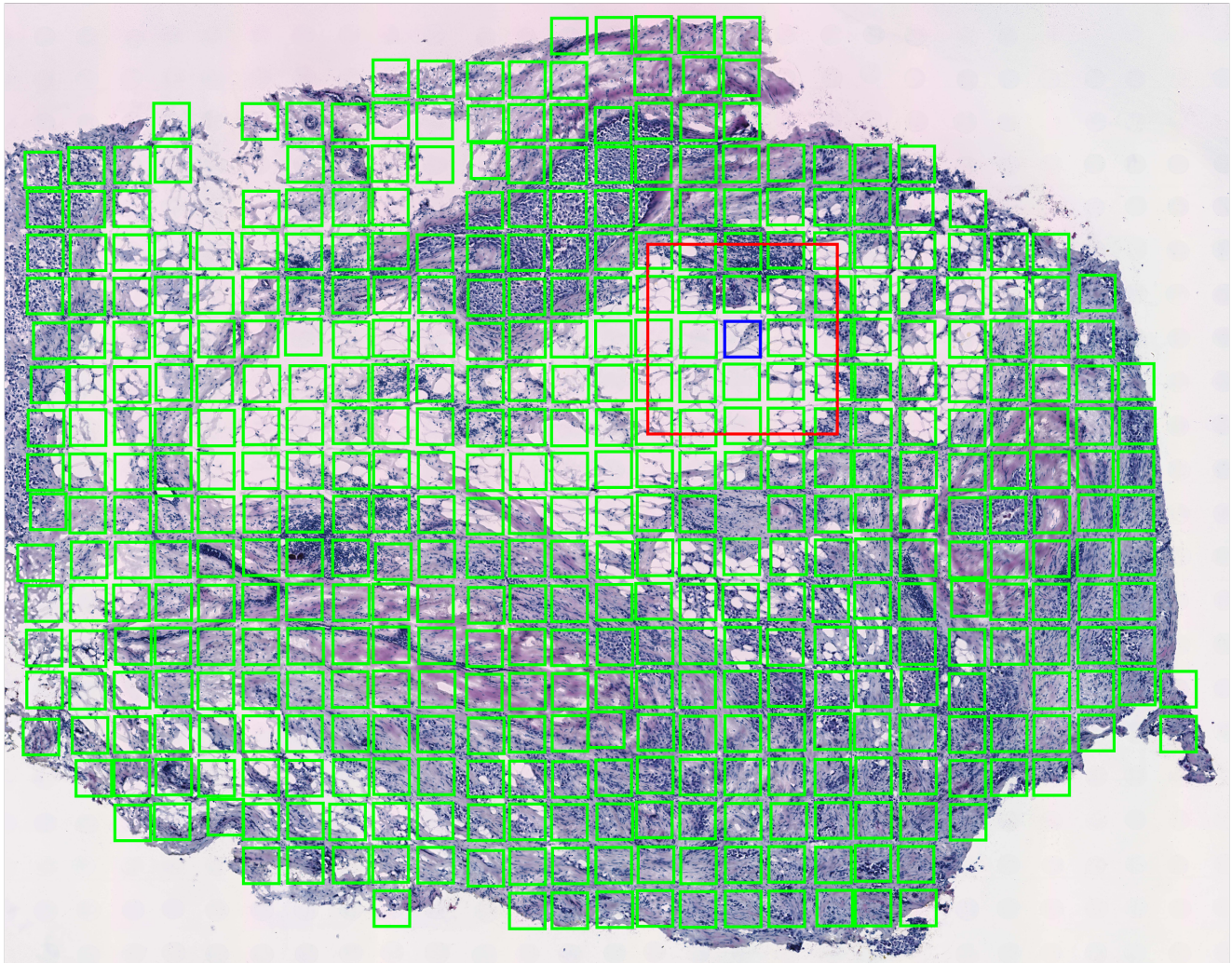
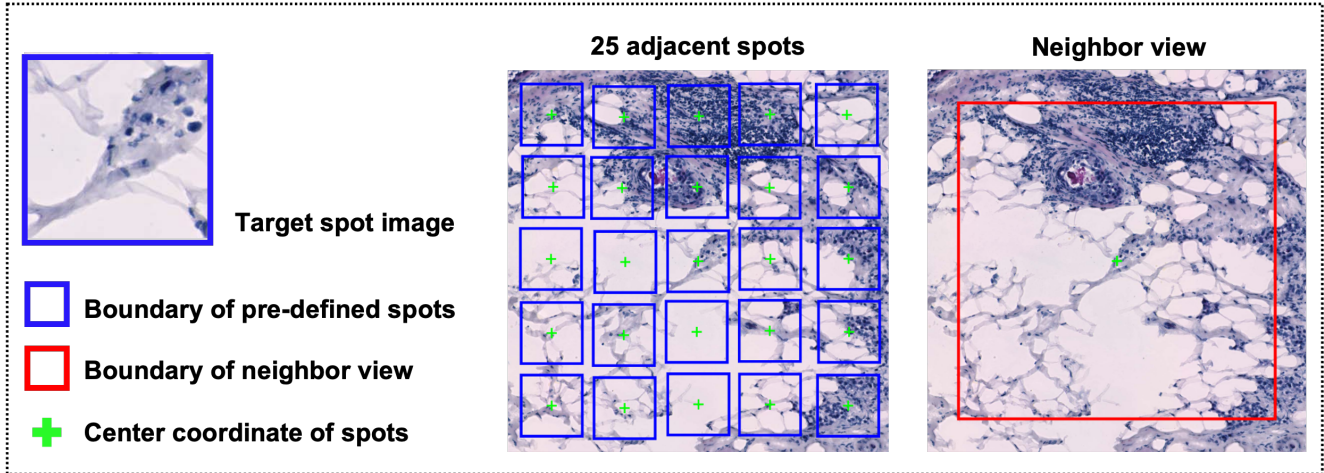


Figure 3. An example of input data for TRIPLEX from BC1 dataset. (Top) Difference between the input data used in NEM and the 25 adjacent spot images around the target spot image. The pre-defined spot image is marked with a blue boundary, while the input data for the NEM model is marked with a red boundary. The '+' within each image indicates the center coordinates. (Bottom) All input data for the same sample. The input data for TEM is marked with a blue boundary, the input data for NEM is marked with a red boundary, and the input data for GEM is marked with a green boundary. (The spot marked with the blue boundary is the target spot image.)

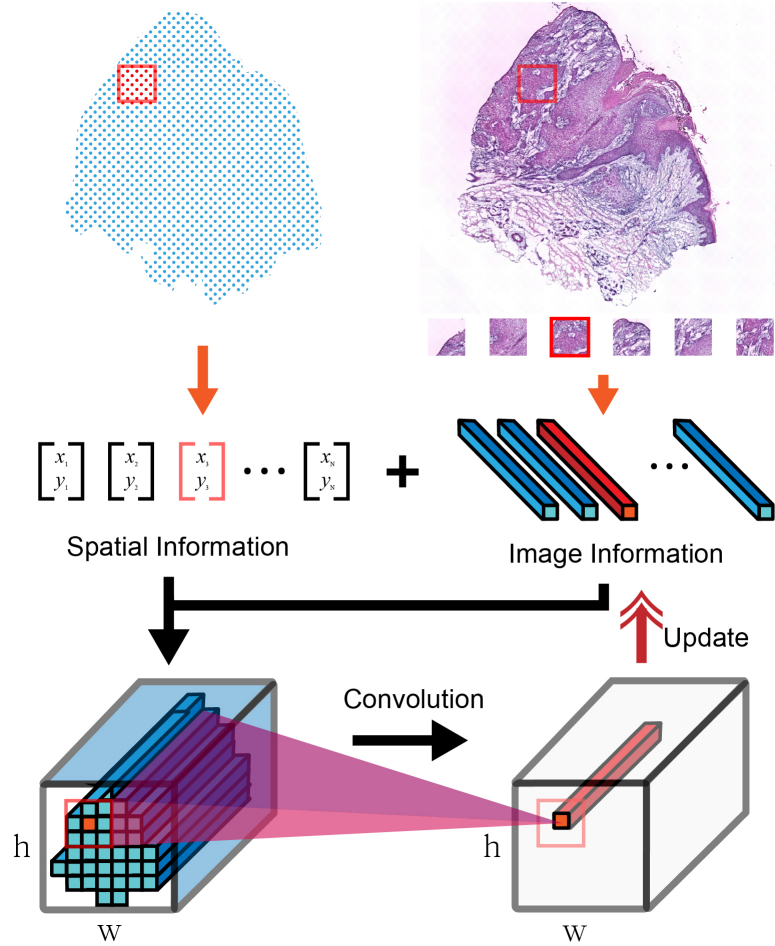


Figure 4. Overview of proposed positional encoding for histology images (APEG). We utilize the coordinates of each spot to reposition the feature token to its original location, apply convolution, and then restore it to its original shape.

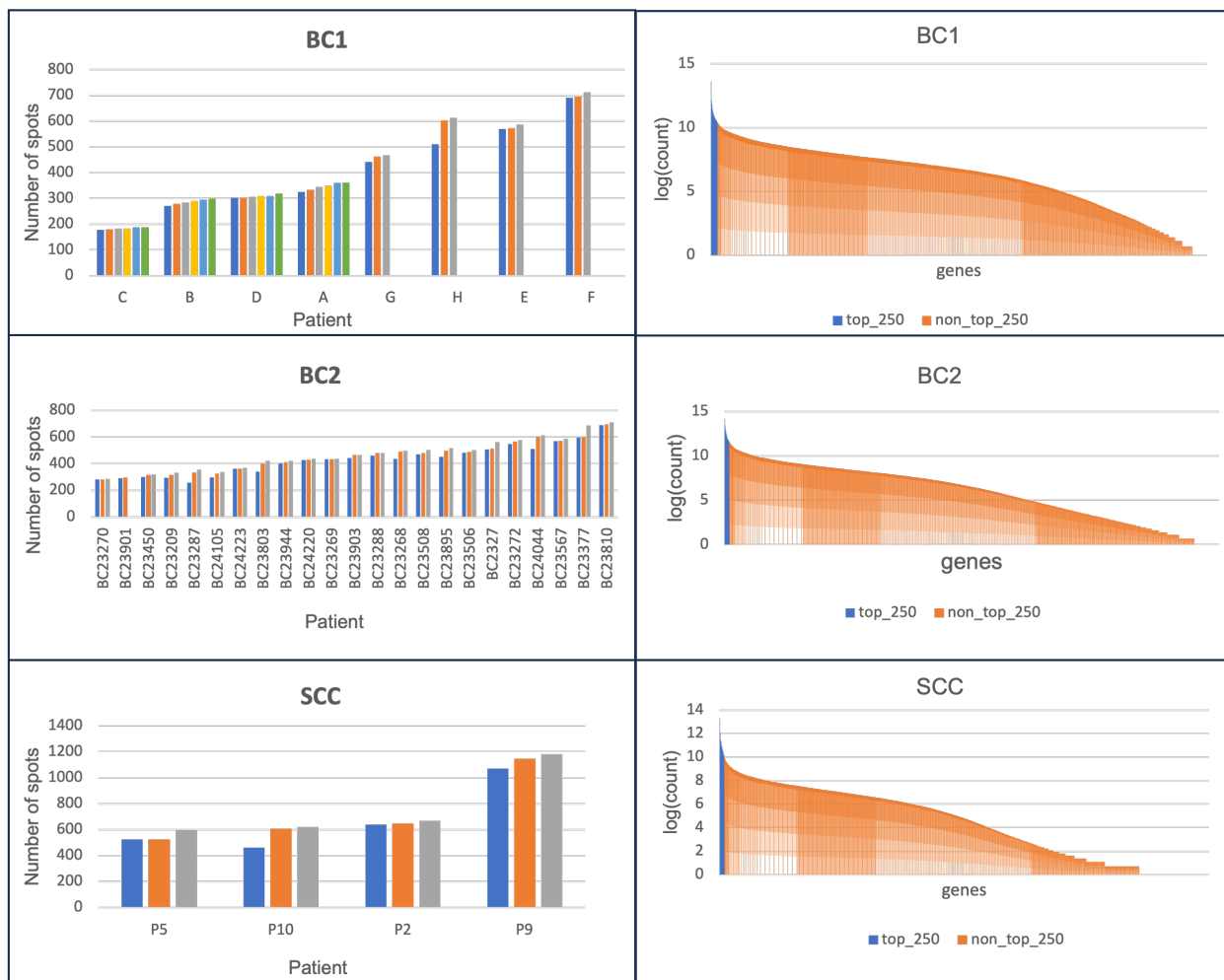


Figure 5. Dataset summary of ST data used for cross-validation. (Left) Number of spots per sample in each dataset. The x-axis label represents each patient, with multiple samples existing for every patient. (Right) Log-transformed count values for each gene in the datasets. The 250 genes utilized in this study correspond to the top genes within the blue region.

Dataset	250 genes (Total genes to be predicted)
SCC	S100A8, KRT6A, KRT14, S100A9, KRT5, KRT6B, KRT16, KRT6C, KRT17, MT-CO3, S100A7, MT-CO2, SFN, S100A2, MT-CO1, ACTB, PERP, SPRR18, KRT10, KRT1, EEF1A1, RPLP1, LGALS7B, LGALS7, COL1A1, FABP5, RPS12, HLA-B, MT-ND4, RPLP2, ACTG1, GJB2, B2M, TPT1, RPL13, MT-ATP6, RPS24, PFN1, KRTDAP, RPS6, DMKN, RPLP0, MT-ND3, RPL37A, DSP, CXCL14, RPS18, RPS17, RPS8, RPL13A, MT-CYB, RPL11, RPL27A, RPL28, MT-ND1, RPS27, RPL32, CSTA, RPL34, RPL31, COL1A2, RPL8, SBSN, TMSB10, ENO1, RPS14, RPL36, SPRR2A, RPL39, GSTP1, RPS27A, JUP, RPS19, RPL37, RPL27, RPL3, RPS29, COL3A1, RPS11, CSTB, RPL9, RACK1, ANXA2, RPL7A, RPL23, RPL19, S100A11, RPS2, RPS28, EEF2, ANXA1, CD74, PABPC1, LDHA, RPS3, RPL35A, DSC2, AQP3, RPS25, IFI27, CALML5, YWHAZ, RPL6, TMSB4X, RPS23, RPL12, S100A14, RPS4X, UBA52, SLPI, PKP1, RPL38, HLA-A, RPS13, LY6D, RPL24, ATP1B3, MYL6, GJB6, S100A6, HSPB1, RPL18, MT-ND2, SDC1, IVL, FTL, RPS3A, RPL10, RPS15A, PI3, RPL18A, S100A10, RPS7, S100A7A, RPL29, RPL26, RPL41, RPL4, RPL7, SPARC, VIM, PTMA, RPS20, MMP1, SH3BGR13, RPL15, MYH9, GJA1, ITM2B, PPIA, RPL14, UBC, RPL5, CD44, AHNAK, RPL21, DSC3, CNFN, CD24, CFL1, COL17A1, HSP90AA1, RPS16, PKM, NACA, RPS5, ALDOA, H3F3B, S100A16, TAGLN2, HLA-C, TRIM29, LYPD3, FAU, LMNA, SPINK5, SPRR2E, RPL22, KRT2, CST3, DSG3, CLCA2, RPSA, DSG1, RPS9, NDRG1, AC090498.1, GRN, TXN, HSPA8, TGFBI, CTSB, SPRR2D, HLA-DRA, ACTN4, RPS21, EIF1, CTSD, ARPC2, CALML3, KLK7, CALM1, GNAS, DYNLL1, FLG, FLNA, DST, SLC2A1, PSAP, EIF4G2, EEF1B2, FGFBP1, LGALS1, ITGA6, MYL12B, TPI1, RPL10A, TMEM45A, BTF3, DSTN, RTN4, HNRNPA2B1, LAD1, ATP1A1, SERPINB3, PRDX1, COL6A1, ATP5E, PPDPF, TYMP, CD63, EIF5A, YWHAQ, PGK1, HLA-E, IFITM3, RPS26, IGFBP4, OAZ1, NPM1, LCE3D, FXYD3, MT2A, COL6A2, POLR2L, CD59, HNRNP, RPL35, TMBIM6, HSP90AB1
BC1	IGKC, TMSB10, ERBB2, IGHG3, IGLC2, IGHA1, GAPDH, ACTB, IGLC3, IGHM, SERF2, PSMB3, PFN1, ACTG1, KRT19, RACK1, MUC1, CISD3, APOE, MIEN1, SSR4, CALR, PSAP, CTSD, FTL, FTH1, TPT1, PTPRF, UBA52, P4HB, BEST1, HLAB, FAU, SLC9A3R1, FN1, COL1A1, EEF2, IGHG4, CALML5, CD74, B2M, FASN, S100A9, MGP, CFL1, PSMD3, IGHG1, HLAA, S100A6, MYL6, COL1A2, PHB, TAGLN2, HLAE, HLAC, KRT7, CD63, SYNGR2, STARD3, PABPC1, GPX4, GRB7, SLC25A6, AEBP1, GNAS, NDUFB9, EDF1, CRIP2, DDX5, OAZ1, EIF4G1, LMNA, GNB2, CST3, PCGF2, SDC1, S100A11, PRDX1, GRINA, ATP6V0B, TFF3, HLADRA, EEF1D, AZGP1, PPP1CA, FLNA, COL3A1, ATP5E, SPDEF, AP0007691, ALDOA, PLXNB2, TAGLN, TUBA1B, APOC1, PRRC2A, LAPTM5, PTMS, KRT18, IFI27, PLD3, ADAM15, C1QA, AES, TSPO, MLLT6, TAPBP, SCAND1, ATP1A1, CD81, SEC61A1, CLDN3, PPDPF, S100A14, BGN, C3, MZT2B, S100A8, MDK, PFDN5, H2AFJ, SH3BGR13, ENO1, XBP1, CYBA, COX6B1, TRAF4, CD24, PRSS8, MMP14, MUC1, VIM, MIDN, SPINT2, BST2, TIMP1, GUK1, ACTN4, CTSB, COX4I1, CCT3, HNRNPA2B1, SEPW1, LY6E, SCD, HSPB1, EIF4G2, BSG, ZYX, TUBB, LASP1, CD99, COL6A2, H1FX, RALY, UBE2M, SPARC, ATG10, HSP90AB1, ORMDL3, LMAN2, CHCHD2, COX7C, ARHGDI1A, VMP1, UBC, IGFBP2, COPE, NUPR1, PERP, KRT81, PPP1R1B, LGALS3BP, SSR2, KIAA0100, MYL9, CIB1, IDH2, STARD10, LGALS1, COX6C, GRN, MAPKAPK2, GNAI2, KDELR1, COL18A1, UQCRCQ, COX5B, ELOVL1, CHPF, CLDN4, C12orf57, LGALS3, HSP90AA1, JUP, A2M, NDUFB7, PGAP3, HSPA8, TCEB2, PEBP1, COPS9, ATP5G2, ATP6AP1, MYH9, LSM4, COX8A, UQCRC1, ATP5B, DHCR24, PTBP1, EIF3B, NDUFA3, FKBP2, MMACHC, RABAC1, ISG15, PTMA, RRBP1, POSTN, C1QB, BCAP31, PSMB4, LAPTM4A, INTS1, FNBP1L, JTB, NBL1, HM13, SLC2A4RG, ROMO1, SERINC2, NDUFA11, RHOC, TXNIP, TYMP, NACA, HSP90B1, SNRPB, PFKL, VCP, ERGIC1, NUCKS1, PSMD8, CALM2, AP2S1, DBI, C4orf48, SDF4, TPI1
BC2	RPS3, IGLL5, RPLP1, TFF3, RPS18, GAPDH, TMSB10, RPLP2, RPS14, RPL37A, RPS19, RPL28, KRT19, RPL8, RPL13, RPL19, ACTB, RPL36, RPL18A, RPL35, RPL18, RPS2, RPS12, RPS21, RACK1, RPL13A, CTSD, FTL, PFN1, MGP, RPS15, RPS11, RPS16, HLAB, UBA52, NHERF1, RPS17, PSAP, RPLP0, SERF2, RPS27, RPS8, RPL27A, MUC1, RPS28, H2AJ, RPL10, CALR, RPS29, RPL38, RPL11, P4HB, RPS6, CST3, FTH1, RPS4X, SSR4, RPL30, ERBB2, APOE, AZGP1, RPL3, COX6C, HLAC, FAU, RPS9, EEF2, B2M, RPS5, RPL12, ACTG1, RPS27A, RPL37, RPL23, HLAA, RPL31, RPL29, RPL7A, IFI27, PABPC1, CD74, BEST1, RPL32, FASN, S100A9, GPX4, RPL15, RPL27, MZT2B, RPL23A, HSPB1, MALAT1, RPS24, COL1A1, C4B, KRT18, CFL1, CD81, ALDOA, RPL35A, SYNGR2, PPP1CA, HLAE, TAGLN, RPL9, CD63, RPS3A, LGALS3BP, IGFBP2, BST2, TPT1, EDF1, RPS25, ATP6V0B, TAPBP, GRINA, XBP1, S100A11, NBEAL1, AEBP1, CCND1, OAZ1, RPL14, TAGLN2, FN1, PPDPF, BCAP31, IFITM3, PRDX1, BGN, GNAS, PTMA, UBC, MZT2A, SLC25A6, RPS20, HSP90AB1, RPS10, MYL6, CLDN3, ATP6AP1, PRDX2, RPL24, GNB2, RPL34, RPL4, LMNA, NDUFA13, HLADRA, SNHG25, TIMP1, H110, RPS23, COX8A, KRT8, LY6E, ENO1, GRN, PTPRF, RPL7, UBB, BSG, ELOB, COX6B1, TMSB4X, C1QA, PRSS8, RPL5, UQCRC1, RPS7, A2M, RPS15A, VIM, S100A6, NDUFA11, PSMD3, EVL, APOC1, H3B3, ATP5F1E, PLXNB2, MYL9, TUBA1B, CTSB, ISG15, FLNA, RPS13, NDUFB9, EIF4A1, POLR2L, CYBA, CRIP2, EEF1D, ATP1A1, ELF3, TUFM, SH3BGR13, STARD10, C3, GUK1, ZNF90, C12orf57, TLE5, SEC61A1, SDC1, PLD3, SPDEF, ARHGDI1A, IFI6, LAPTM5, RPL41, CLU, GNAI2, PFDN5, RPL39, SSR2, COX4I1, RHOC, JUP, EIF4G1, FXYD3, TSPO, UQCRCQ, COL1A2, RPL10A, S100A8, SELENOW, TPI1, ATP5MC2, PTMS, IGFBP5, LGALS1, SPINT2, RPSA, GSTP1, CHCHD2, EIF5A, COX5B, ATG10, RPL6, EEF1A1, CAPNS1, LMAN2, UBE2M, SPARC, EIF3C, GAS5, TUBB, ACTN4, IGFBP4

Figure 6

Dataset	Highly predictive genes (Top 50)
SCC	'TMSB10', 'MYL6', 'PTMA', 'EEF2', 'PFN1', 'TRIM29', 'PKP1', 'RPL8', 'NACA', 'TAGLN2', 'FXYD3', 'CD63', 'PRDX1', 'RPL9', 'UBA52', 'RPS3A', 'HLA-A', 'RPS3', 'HSPB1', 'RPL5', 'RPL18', 'TPI1', 'HLA-B', 'RPS5', 'MYL12B', 'RPL3', 'RPL22', 'LGALS1', 'RPL10', 'LAD1', 'RPS9', 'RPS4X', 'GJB6', 'RPL36', 'RPS20', 'COL6A2', 'FLNA', 'RPL28', 'RPL12', 'RPS17', 'FTL', 'RPS16', 'HNRNPA2B1', 'HSP90AB1', 'HLA-C', 'RPS11', 'ALDOA', 'S100A16', 'RPL29', 'MYH9'
BC1	'CD24', 'HNRNPA2B1', 'HSP90AB1', 'GNAS', 'FASN', 'MLLT6', 'CCT3', 'ERBB2', 'GRB7', 'HSP90AA1', 'NACA', 'SPINT2', 'ATP1A1', 'CLDN4', 'COX6C', 'PTPRF', 'CALM2', 'SCD', 'ATP5B', 'PERP', 'DBI', 'DDX5', 'ACTG1', 'PTMA', 'PEBP1', 'HSP90B1', 'PSMD3', 'LAPTM4A', 'COX7C', 'VMP1', 'FNBP1L', 'C3', 'EIF4G2', 'IGHA1', 'PLXNB2', 'PRDX1', 'FN1', 'PGAP3', 'PRSS8', 'KRT7', 'MMACHC', 'PABPC1', 'TUBA1B', 'JTB', 'VCP', 'S100A11', 'PCGF2', 'JUP', 'XBP1', 'CHCHD2'
BC2	'FASN', 'GNAS', 'ACTG1', 'HSP90AB1', 'ATP1A1', 'PTMA', 'PTPRF', 'H3-B3', 'GAS5', 'SPINT2', 'COX6C', 'HLA-DRA', 'ERBB2', 'APOE', 'RPL9', 'TUBA1B', 'EIF4A1', 'AEBP1', 'PLXNB2', 'TPT1', 'IGLL5', 'RPL31', 'RPS24', 'XBP1', 'COL1A2', 'SPARC', 'JUP', 'CD74', 'RPL5', 'COL1A1', 'PABPC1', 'PSMD3', 'CLDN3', 'CHCHD2', 'EVL', 'LAPTM5', 'C1QA', 'BGN', 'PRDX1', 'EEF1A1', 'RPL6', 'ACTN4', 'RPL15', 'C3', 'RPL13', 'RPS15A', 'RPLP1', 'CCND1', 'PRSS8', 'ATG10'

Figure 7

Dataset	Genes not available
Breast Visium data	'AES', 'AP000769.1', 'ATP5B', 'ATP5E', 'ATP5G2', 'SEPW1', 'TCEB2'

Figure 8

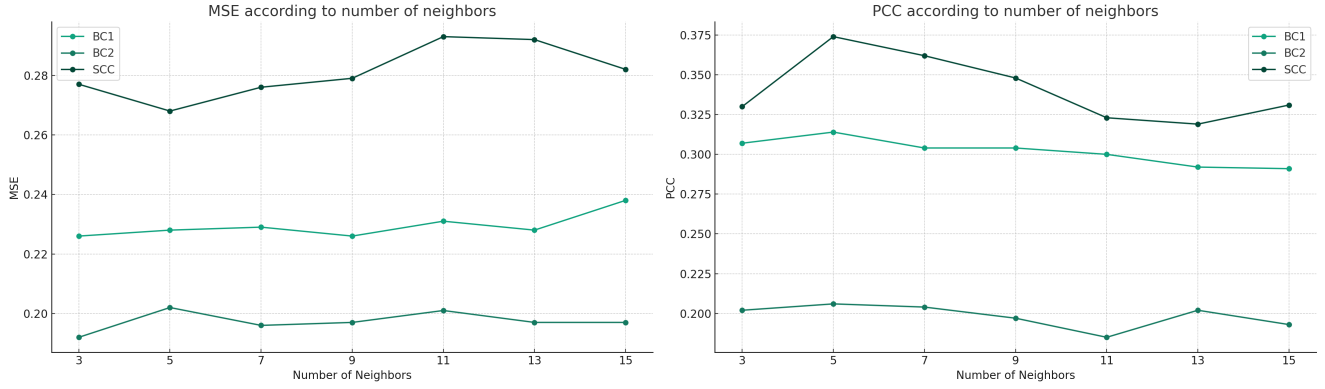


Figure 9. The performance varies with the size of the neighbor view. Variations in MSE (**Left**) and PCC (M) (**Right**) relative to the size. "Number of neighbors" represents the count of 224x224 patches along an axis

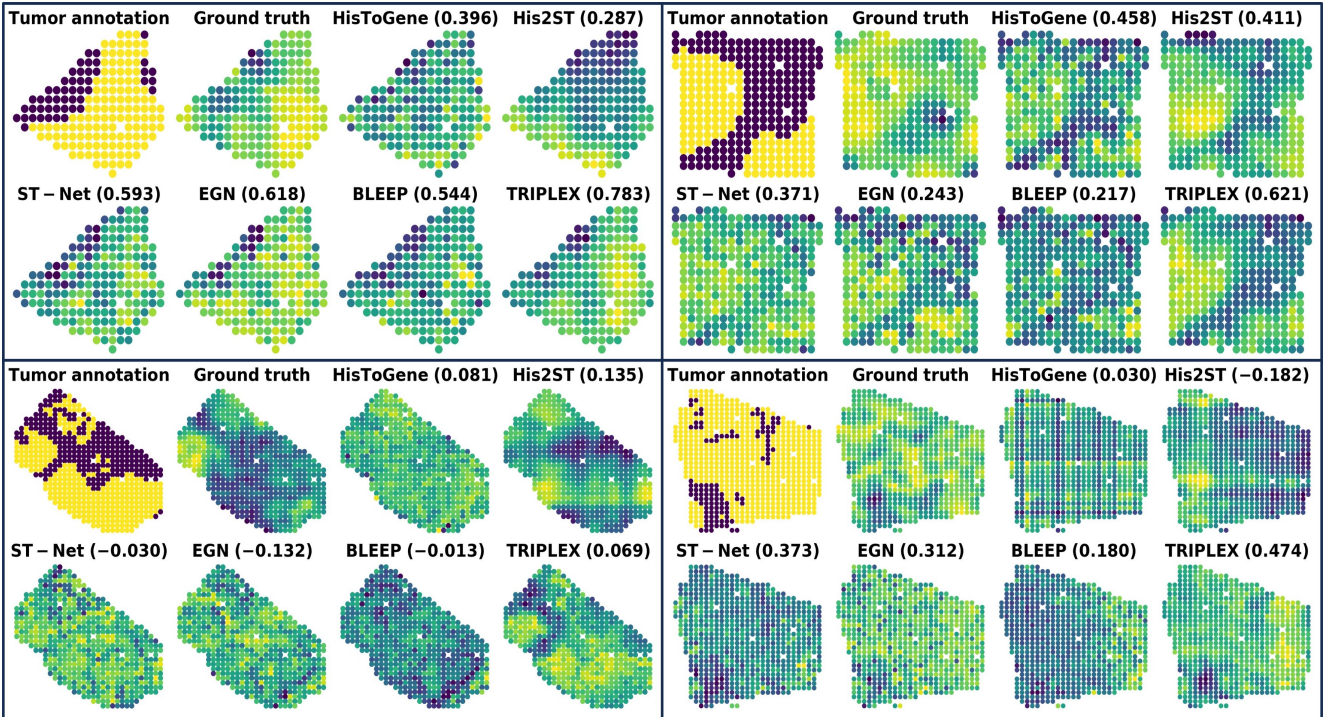


Figure 10. Additional visualization for predicting GNAS gene expression levels in BC1 dataset. We display the Pearson Correlation Coefficient (PCC) values between the ground truth and the prediction of the GNAS expression level estimated by each model.

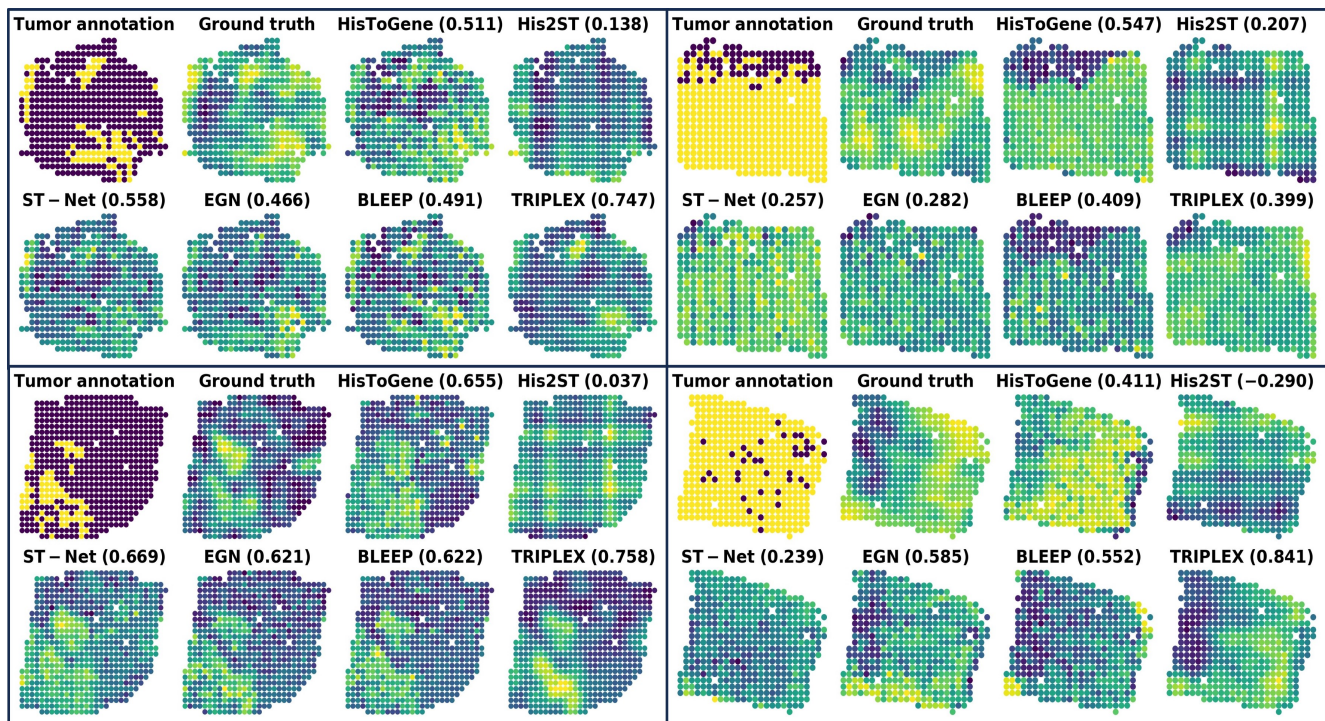


Figure 11. Additional visualization for predicting GNAS gene expression levels in BC2 dataset.

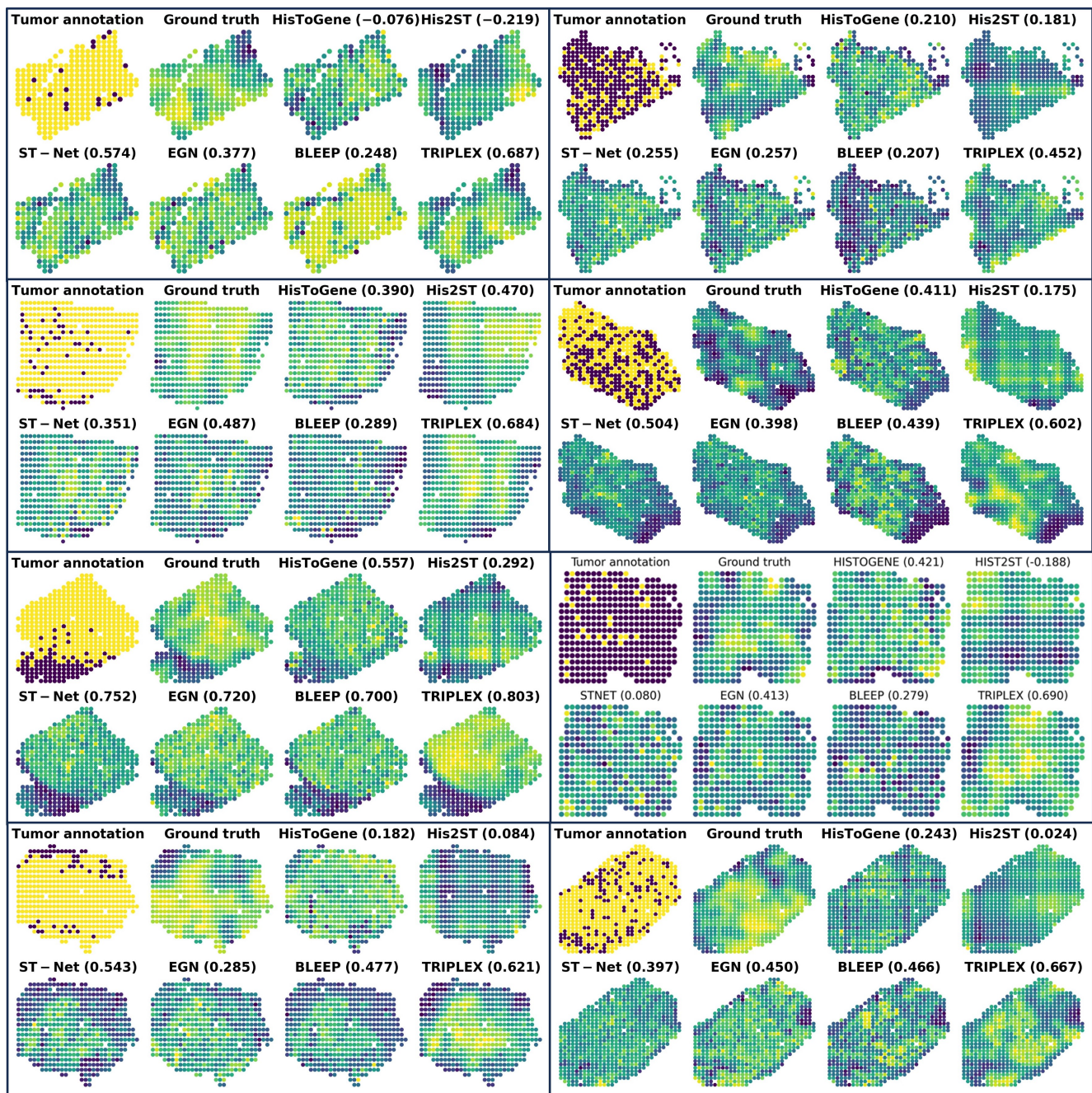


Figure 12. Additional visualization for predicting GNAS gene expression levels in BC2 dataset.

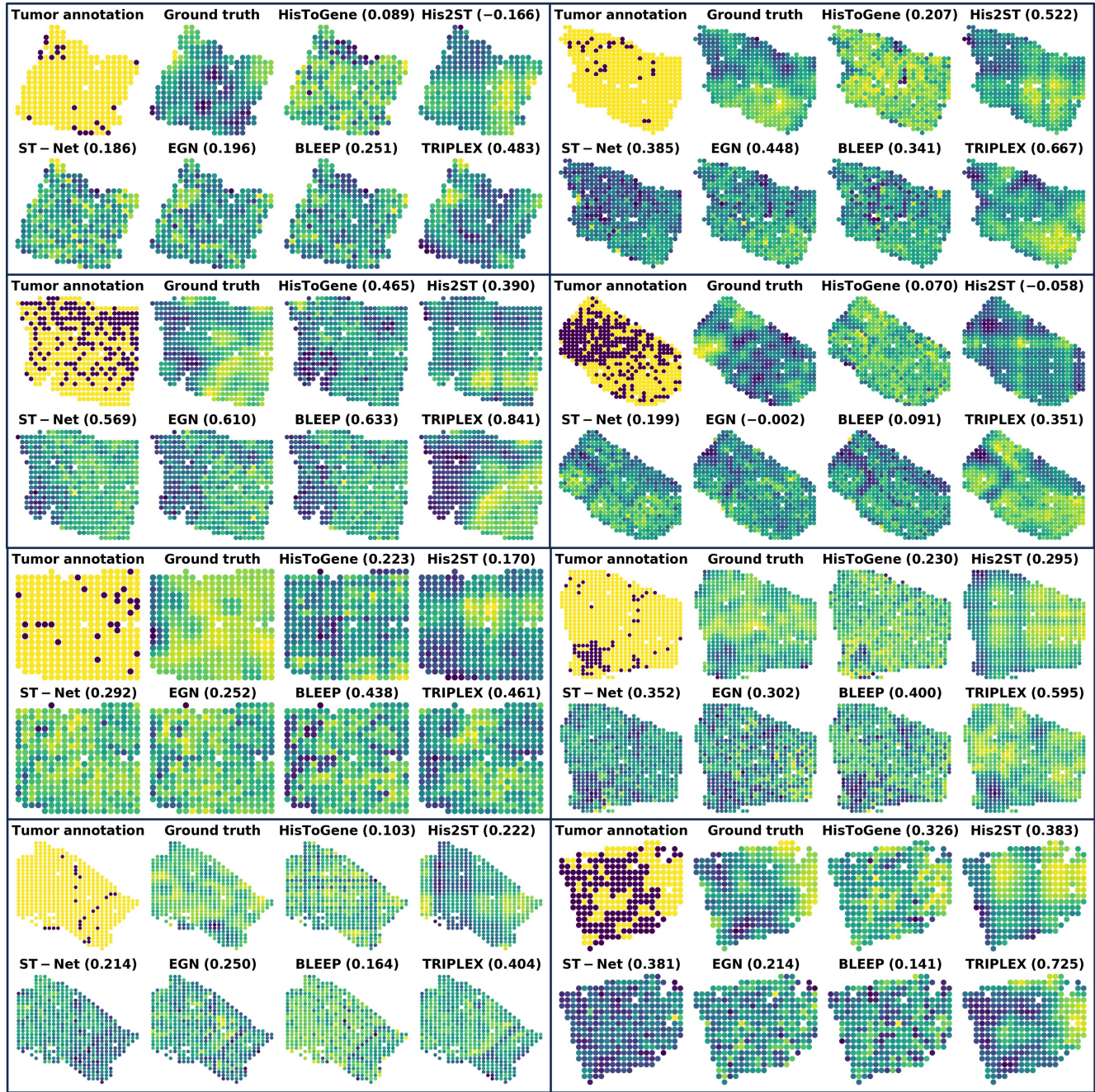


Figure 13. Additional visualization for predicting GNAS gene expression levels in BC2 dataset.

Model	HisToGene		
Parameter	BC1	BC2	SCC
n_layers	4	3	5
dim	2048	2048	512
num_heads	4	16	4
dropout	0.4	0.3	0.4

Model	HisT2ST		
Parameter	BC1	BC2	SCC
depth1	4	2	2
depth2	3	6	6
depth3	4	1	2
heads	8	4	8
channel	64	32	16
bake	3	5	7
kernel_size	3	3	7

Model	EGN		
Parameter	BC1	BC2	SCC
dim	2048	512	2048
mlp_dim	2048	4096	2048
depth	6	6	8
heads	4	8	16
bhead	16	4	16
bdim	128	64	64

Model	BLEEP		
Parameter	BC1	BC2	SCC
projection_dim	128	128	128
dropout	0.4	0.3	0.35

Model	TRIPLEX		
Parameter	BC1	BC2	SCC
depth1	1	3	2
depth2	3	3	2
depth3	3	4	4
dropout1	0.2	0.4	0.1
dropout2	0.1	0.1	0.1
dropout3	0.3	0.3	0.3
mlp_ratio1	4	4	4
mlp_ratio2	4	2	1
mlp_ratio3	1	4	1
num_heads1	4	16	8
num_heads2	16	8	16
num_heads3	16	8	16

Table 14. Selected hyperparameters in each dataset