A. 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:

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

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:

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

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

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

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.

B. Method Details

B.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 1.

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 [3]. 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 1.

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, Sec-

tion 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 6.

B.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 [3]. 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.

B.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) [4] 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. Instead, 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.

B.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 2.

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.

B.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.

B.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 [5] utilizes a DenseNet121 [6] model, pre-trained on ImageNet, with minimal modifications (only re-

placing 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 [9, 13]. 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 [12], in addition to following the preprocessing steps of TRIPLEX, we implement the method proposed by [12] to obtain k exemplars for all spots. We replace the unsupervised SF2GAN-based model from [11] with a pretrained ResNet18 model used in TRIPLEX. This decision, informed by the results in [12], ensures a fair comparison by maintaining consistent feature extraction strategies across models.

BLEEP BLEEP [10] 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 [8] 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.

C. Additional implementation details and experimental results

C.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 3, 4, and 5.

C.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.

C.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 [2] 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 [7].

C.4. Computational Cost Comparison

Table 1 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.

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 1. Computational cost comparison

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

Table 2 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.

Source	Model	BC1 MAE	BC2 MAE	SCC MAE
Local	ST-Net [5]	0.389 ± 0.03	0.349 ± 0.02	0.428 ± 0.05
	EGN [12]	0.377 ± 0.04	0.337 ± 0.02	0.418 ± 0.06
	BLEEP [10]	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 [9]	0.428 ± 0.07	0.335 ± 0.04	0.415 ± 0.07
	Hist2ST [13]	0.413 ± 0.07	$\mathbf{0.333} \pm 0.02$	0.924 ± 0.29
	GEM	0.383 ± 0.05	0.352 ± 0.02	0.434 ± 0.12
Multiple	TRIPLEX	$\mathbf{0.362} \pm 0.05$	0.343 ± 0.02	$\mathbf{0.404} \pm 0.07$

Table 2. Cross validation result on each ST dataset. The mean and standard deviation of MAE from the cross-validation results are displayed.

C.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 7. 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 information relevant to the target and maintaining manageable computational costs.

C.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 3 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 [9, 13]	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 [9, 13]	0.163	0.313	0.251	0.416

Table 3. 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 [5]. 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 4, confirm that the original results reported in EGN's literature are reproducible when adopting their specific normalization and evaluation strategies.

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.

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 [12]	0.0003	0.0134	0.2003	0.3011

Table 4. Result comparison for different evaluation and normalization method in BC2 dataset.

C.8. Additional Ablation Studies

We conduct further ablation studies on the BC1, BC2, and Visium datasets, with the results detailed in Tables 5, 6, and 7 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 5. Ablation studies in BC1 dataset

C.9. Detailed Hyperparameter Settings

In our study, hyperparameter tuning for each dataset is meticulously conducted using the WanDB platform [1]. We set the range of hyperparameters based on the defaults reported in relevant literature, as shown in Table 8. 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 [9, 10, 12, 13]. Regarding TRIPLEX, 'depth1', 'depth2', and 'depth3' refer to the depths of the transformer blocks in the Fusion

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 6. 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 7. Ablation studies in Visium dataset

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 9.

D. Additional Visualizations

In this section, we present additional visualizations focusing on the spatial expression distribution prediction of the GNAS gene, as shown in Figures 8 to 11. 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

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 8. Hyperparameters to be tuned

align closely with tumor annotations. This consistency is particularly evident when compared to the predictions from other baseline models [5, 9, 10, 12, 13].

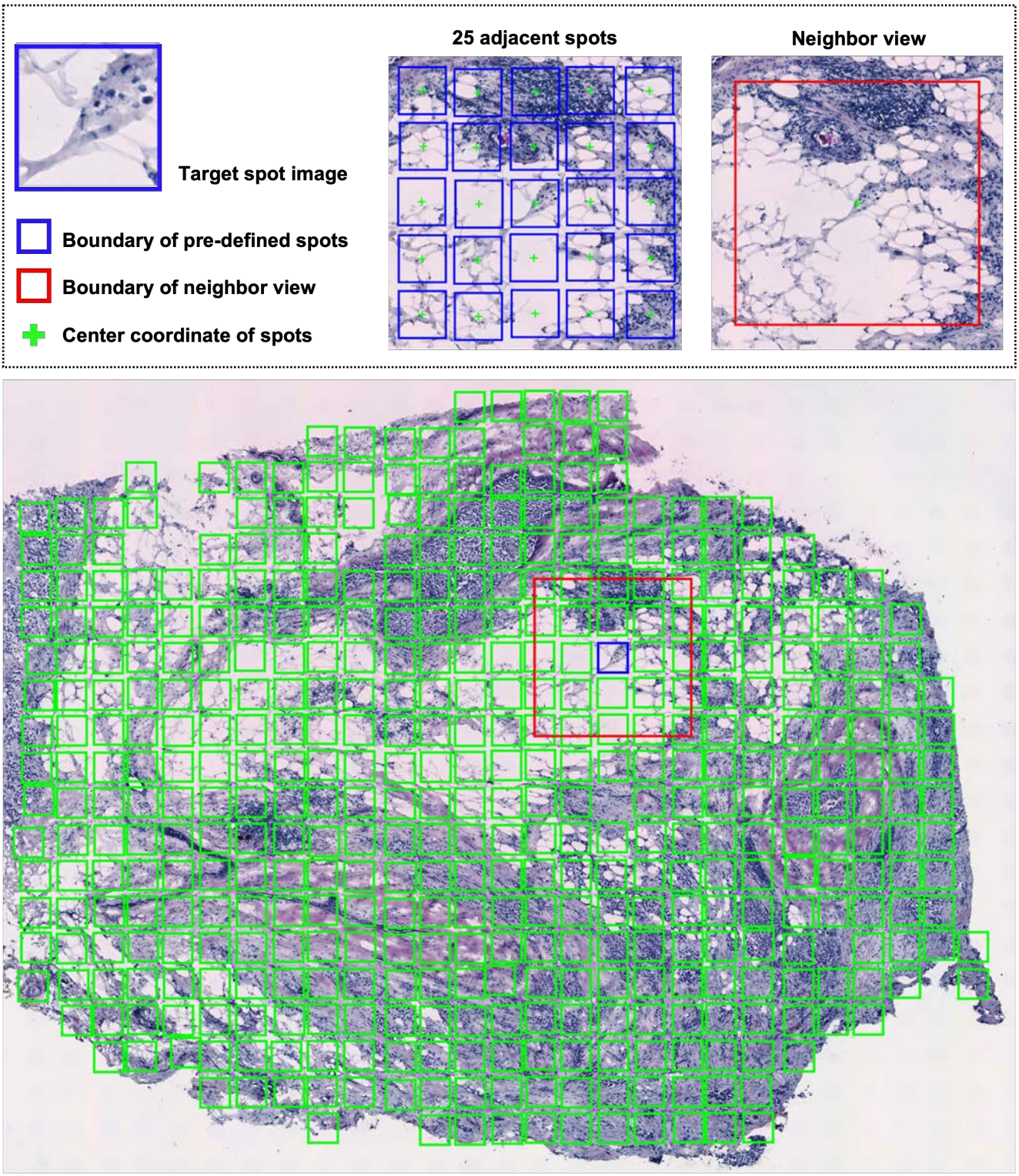

Figure 1. An example of input data for TRIPLEX from BC1 dataset. (Top) Difference between the input data used in NEM and the 25 adjacent spot images around the target spot image. The pre-defined spot image is marked with a blue boundary, while the input data for the NEM model is marked with a red boundary. 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.)

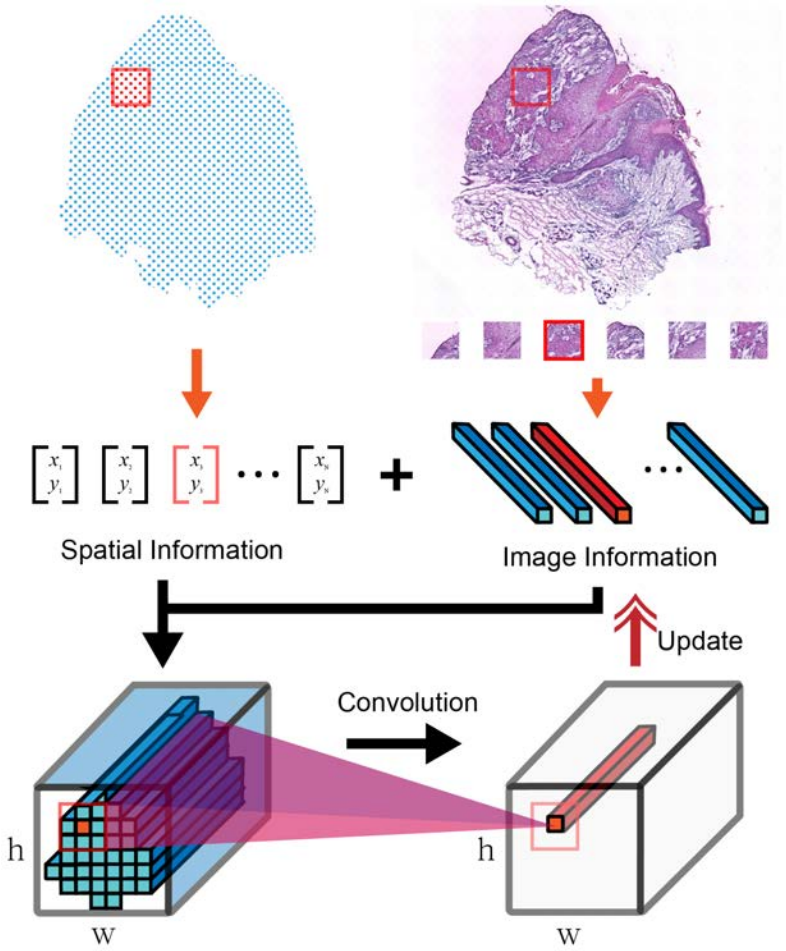

Figure 2. Overview of proposed positional encoding for histology images (APEG). We utilize the coordinates of each spot to reposition the feature token to its original location, apply convolution, and then restore it to its original shape.

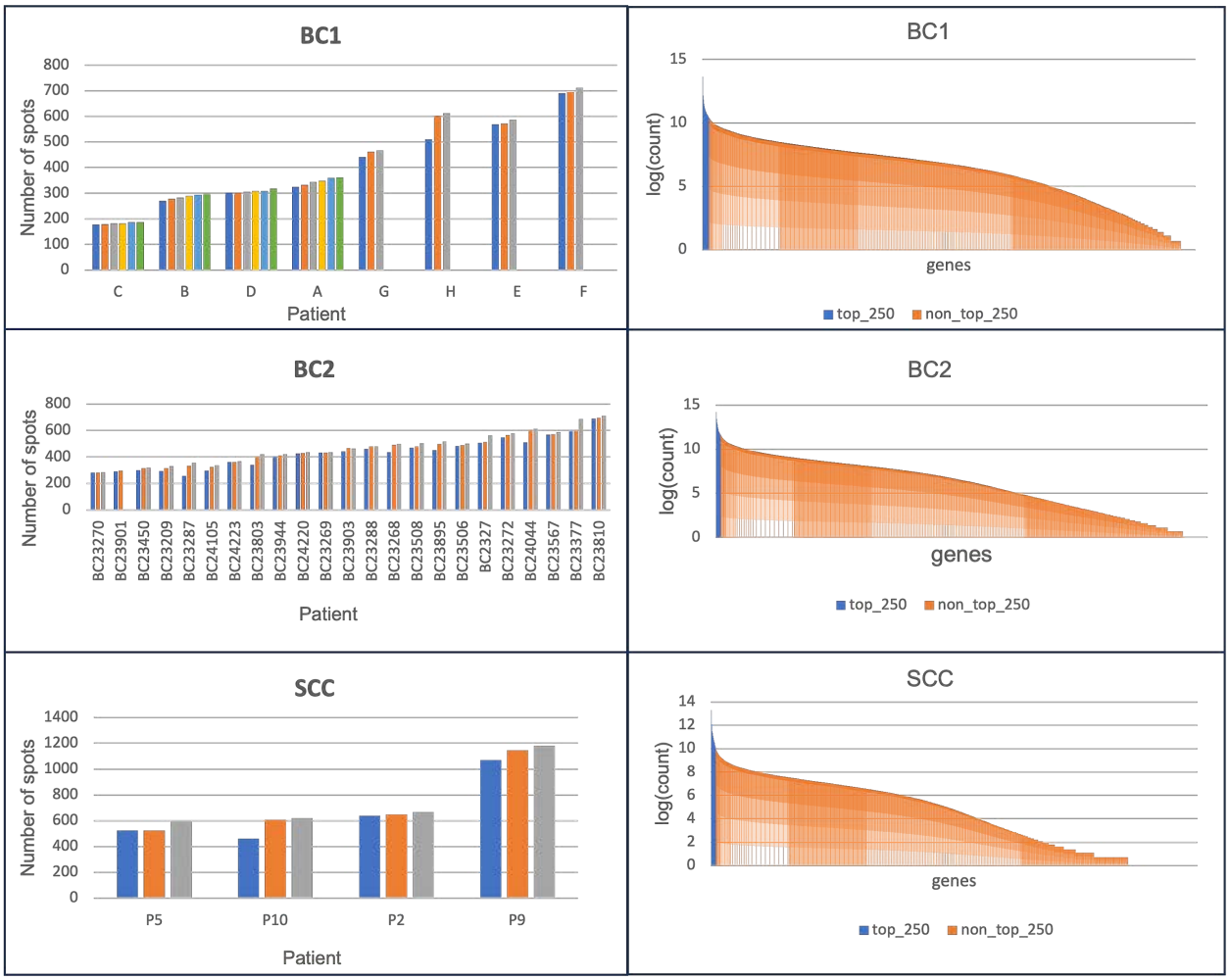

Figure 3. Dataset summary of ST data used for cross-validation. (Left) Number of spots per sample in each dataset. The x-axis label represents each patient, with multiple samples existing for every patient. (Right) Log-transformed count values for each gene in the datasets. The 250 genes utilized in 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, SPRR1B, 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, SH3BGRL3, 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, HNRNPK, RPL35, TMBIM6, HSP90AB1
BC1	IGKC, TMSB10, ERBB2, IGHG3, IGLC2, IGHA1, GAPDH, ACTB, IGLC3, IGHM, SERF2, PSMB3, PFN1, ACTG1, KRT19, RACK1, MUCL1, 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, SH3BGRL3, 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, ARHGDIA, VMP1, UBC, IGFBP2, COPE, NUPR1, PERP, KRT81, PPP1R1B, LGALS3BP, SSR2, KIAA0100, MYL9, CIB1, IDH2, STARD10, LGALS1, COX6C, GRN, MAPKAPK2, GNAI2, KDELR1, COL18A1, UQCRQ, COX5B, ELOVL1, CHPF, CLDN4, C12orf57, LGALS3, HSP90AA1, JUP, A2M, NDUFB7, PGAP3, HSPA8, TCEB2, PEBP1, COPS9, ATP5G2, ATP6AP1, MYH9, LSM4, COX8A, UQCR11, 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, UQCR11, RPS7, A2M, RPS15A, VIM, S100A6, NDUFA11, PSMD3, EVL, APOC1, H33B', ATP5F1E, PLXNB2, MYL9, TUBA1B, CTSB, ISG15, FLNA, RPS13, NDUFB9, EIF4A1, POLR2L, CYBA, CRIP2, EEF1D, ATP1A1, ELF3, TUFM, SH3BGRL3, STARD10, C3, GUK1, ZNF90, C12orf57, TLE5, SEC61A1, SDC1, PLD3, SPDEF, ARHGDIA, IFI6, LAPTM5, RPL41, CLU, GNAI2, PFDN5, RPL39, SSR2, COX4I1, RHOC, JUP, EIF4G1, FXYD3, TSPO, UQCRQ, 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 4

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', 'H3-3B', '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 5

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

Figure 6

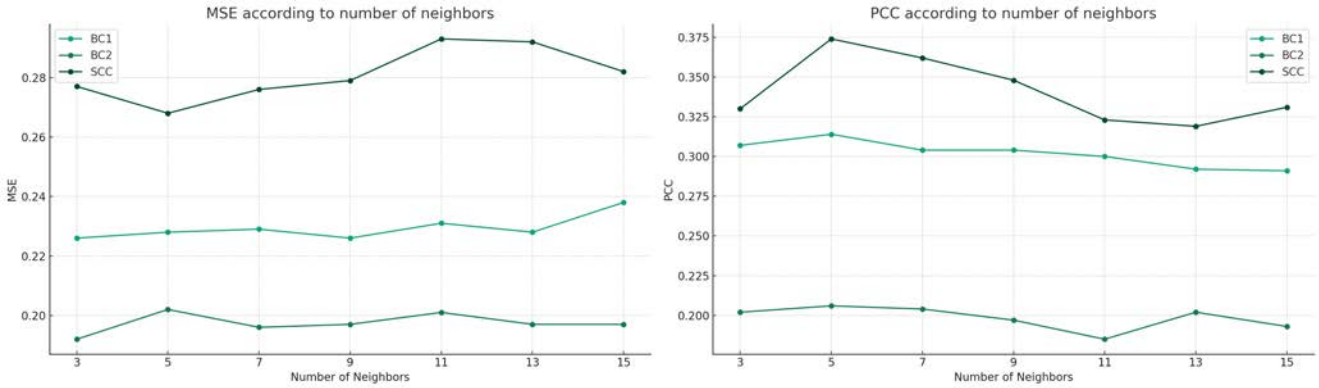

Figure 7. The performance varies with the size of the neighbor view. Variations in MSE (**Left**) and PCC (M) (**Right**) relative to the size. "Number of neighbors" represents the count of 224x224 patches along an axis

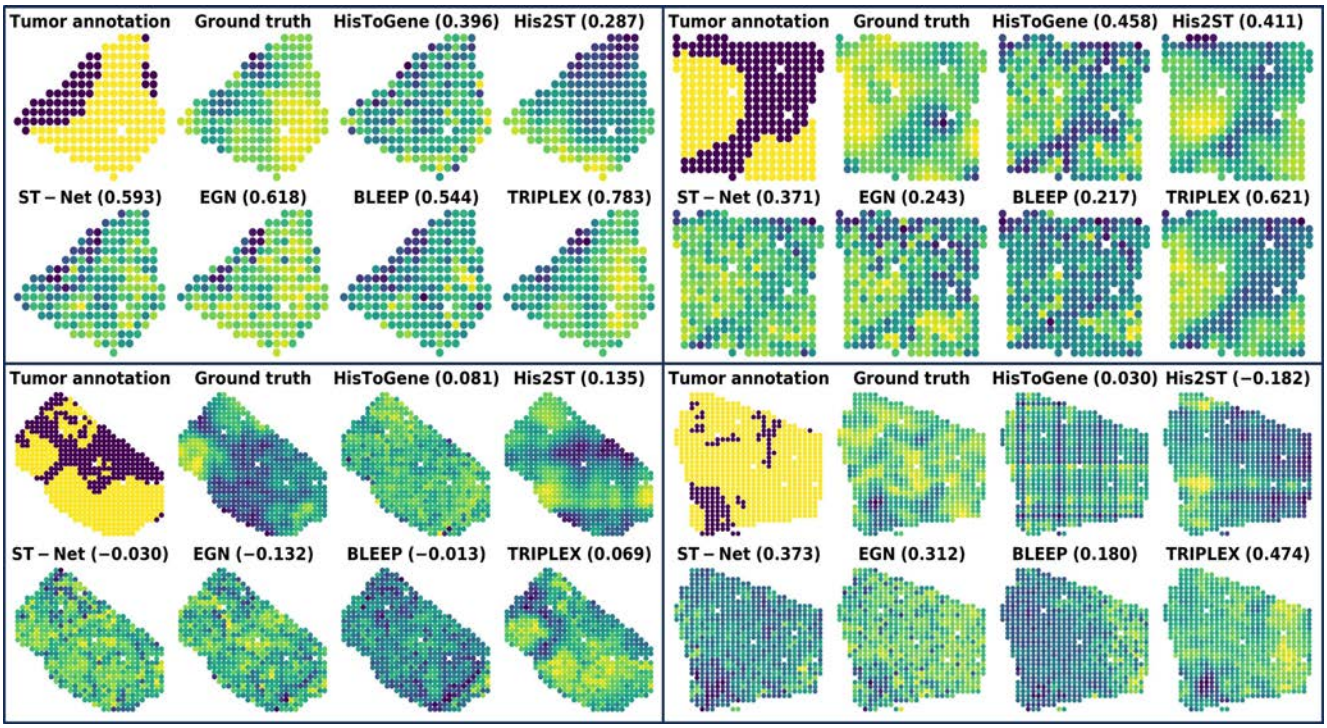

Figure 8. Additional visualization for predicting GNAS gene expression levels in BC1 dataset. We display the Pearson Correlation Coefficient (PCC) values between the ground truth and the prediction of the GNAS expression level estimated by each model.

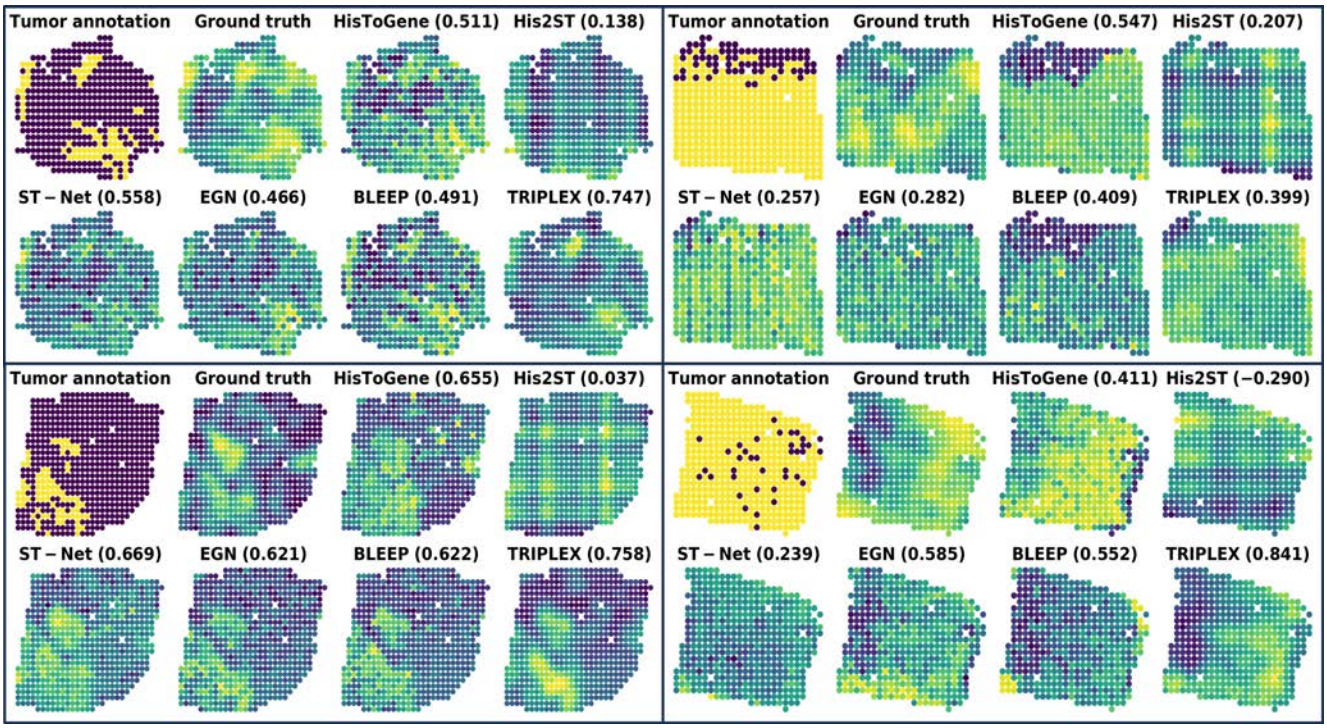

Figure 9. Additional visualization for predicting GNAS gene expression levels in BC2 dataset.

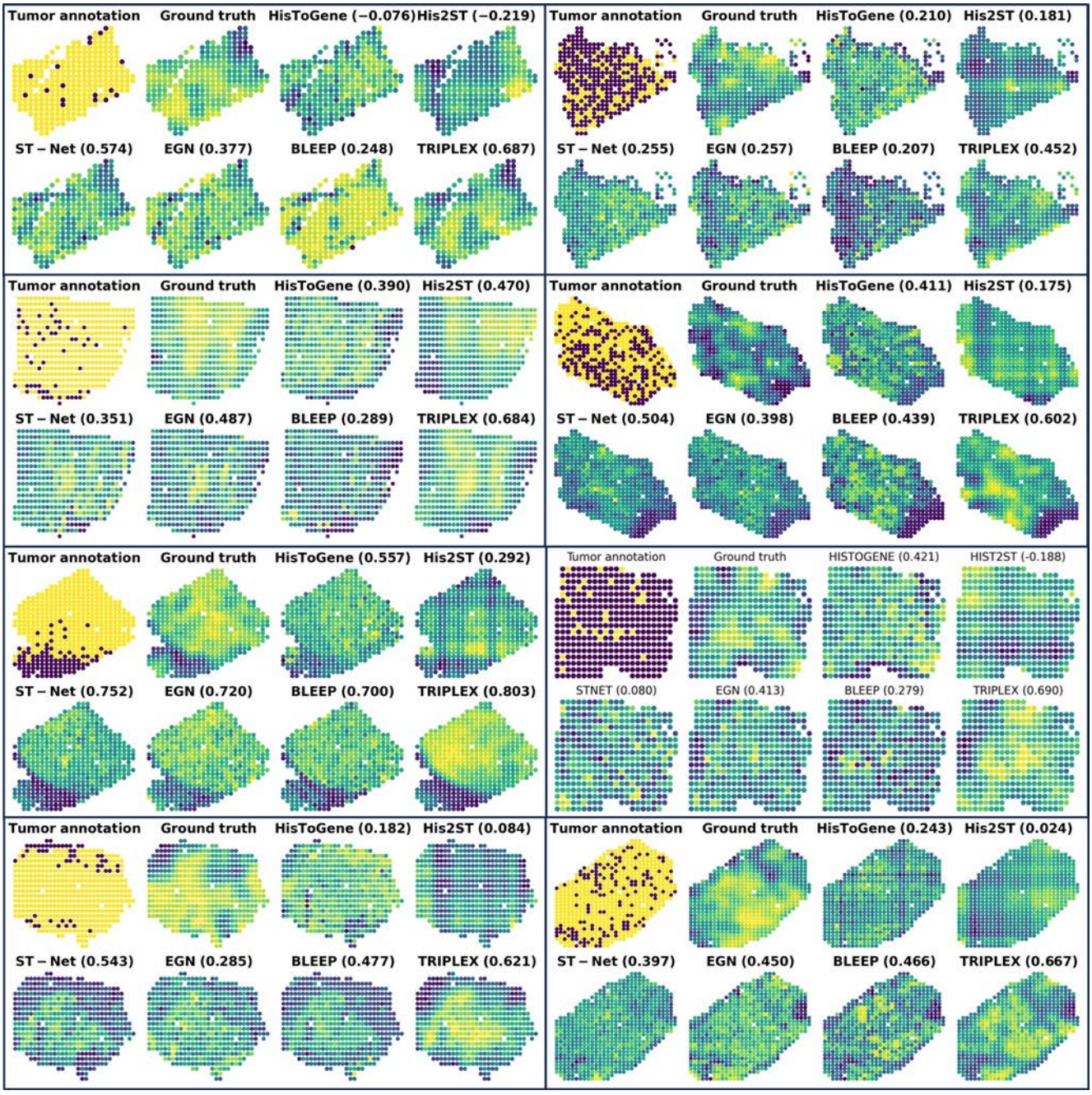

Figure 10. Additional visualization for predicting GNAS gene expression levels in BC2 dataset.

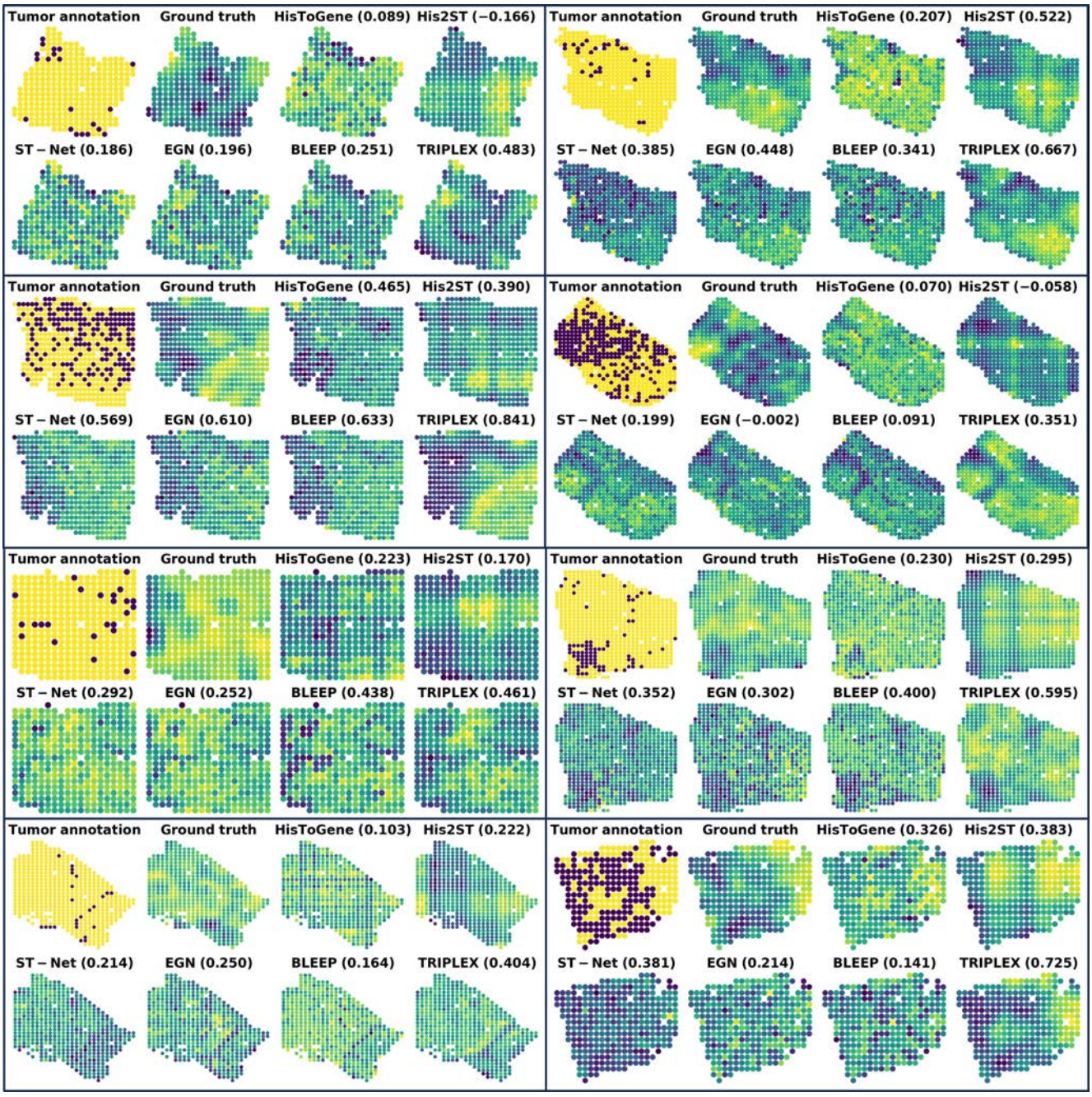

Figure 11. Additional visualization for predicting GNAS gene expression levels in BC2 dataset.

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 9. Selected hyperparameters in each dataset

References

[1] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. 5

[2] 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. 3

[3] Ozan Ciga, Tony Xu, and Anne Louise Martel. Self supervised contrastive learning for digital histopathology. *Machine Learning with Applications*, 7:100198, 2022. 1, 2

[4] 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

[5] Bryan He, Ludvig Bergenstråhle, Linnea Stenbeck, Abubakar Abid, Alma Andersson, Åke Borg, Jonas Maaskola, Joakim Lundeberg, and James Zou. Integrating spatial gene expression and breast tumour morphology via deep learning. *Nature biomedical engineering*, 4(8): 827–834, 2020. 2, 4, 5, 6

[6] 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

[7] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. 3

[8] 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. 3

[9] 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. 3, 4, 5, 6

[10] Ronald Xie, Kuan Pang, Gary D. Bader, and Bo Wang. Spatially resolved gene expression prediction from he histology images via bi-modal contrastive learning, 2023. 3, 4, 5, 6

[11] 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. 3

[12] 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. 3, 4, 5, 6

[13] 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. 3, 4, 5, 6