Spatial Transcriptomics as Images for Large-Scale Pretraining

Supplementary Material

This supplementary document provides comprehensive technical details, extended experimental validations, and in-depth analyses supporting the main paper. We elaborate on the mathematical foundations of our patch-based pretraining framework, provide exhaustive dataset statistics and preprocessing protocols, disclose additional implementation specifics for reproducibility, and present extended ablation studies that substantiate our design choices.

A. Experiment Details

A.1. Spatial Domain Detection.

As described in the main text, we use a 3-layer multilayer perceptron (MLP) with hidden sizes $[512, 256, 128]$ to classify each spot into one of C spatial domains. For an input feature vector $x \in \mathbb{R}^{d_{\text{in}}}$, the MLP computes a sequence of hidden representations

$$h^{(0)} = x, \tag{1a}$$

$$h^{(\ell)} = \mathrm{ReLU}\big(W_\ell h^{(\ell-1)} + b_\ell\big), \qquad \ell = 1, 2, 3, \tag{1b}$$

$$z = W_{\text{out}} h^{(3)} + b_{\text{out}}, \tag{1c}$$

where $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{d_\ell}$ are the learnable weights and biases of the ℓ-th hidden layer with $d_1 = 512$, $d_2 = 256$, and $d_3 = 128$, and $W_{\text{out}} \in \mathbb{R}^{C \times 128}$ and $b_{\text{out}} \in \mathbb{R}^C$ are the parameters of the output layer. The vector $z \in \mathbb{R}^C$ contains the logits for the C classes.

We obtain class probabilities by applying the softmax function to the logits:

$$p = \mathrm{softmax}(z), \tag{2a}$$

$$\mathrm{softmax}(z)_i = \frac{\exp(z_i)}{\sum_{j=0}^{C-1} \exp(z_j)}, \qquad i = 0, \ldots, C-1. \tag{2b}$$

Given a ground-truth class index $y \in \{0, \ldots, C-1\}$, the model is trained by minimizing the cross-entropy loss

$$\mathcal{L}(x, y) = -\log p_y = -\log\big(\mathrm{softmax}(z)_y\big). \tag{3}$$

During inference, the class label is predicted by

$$\hat{y} = \arg\max_i \; p_i = \arg\max_i \; z_i, \tag{4}$$

which simply selects the class with the largest logit (equivalently, the highest softmax probability).

As discussed in the main paper, Spatial Domain Detection can be implemented using k-nearest neighbors based on features extracted by pre-trained models or raw gene expressions, thereby eliminating the need for training in Spatial Domain Detection. The prediction for the i-th sample, y_i, can be written as:

$$\hat{y}_i = \mathrm{mode}(y_{nn_1}, y_{nn_2}, \ldots, y_{nn_k}), \tag{5}$$

$$y_{nn_k} = \mathrm{rank}(d_j)_k, \; d_j = \mathrm{distance}(\bar{e}_i, \bar{e}_j), \; x_j \in \mathcal{D}_{\sqcup\triangledown} \tag{6}$$

Where $\hat{y}_i$ is the predict class label for the i-th sample, determined using KNN, the mode function returns the most frequent class label among the k nearest neighbors $(y_{nn_1}, y_{nn_2}, ..., y_{nn_3})$, which are identified base on d_j, d_j measures the distances between the feature of the i-th sample, $\bar{e}_i$, and that of each sample x_j in the training dataset $\mathcal{D}_{tr}$, using a suitable distance metric like Euclidean distance, the function rank sorts these distances to determine the k nearest neighbors.

Both of these classification protocols can be applied to Spatial Domain Detection task to better explore the capabilities of pre-trained models and to clearly distinguish them from traditional method, which pre-trained by spot-based samples. Additionally, since the labels in spatial transcriptomics (ST) data are highly spatially correlated—e.g., in the DLPFC (Dorsolateral Prefrontal Cortex) dataset the "Layer 3" spots almost always form a continuous band that lies between Layer 2 and Layer 4—different methods' performance on the spatial-domain detection task can clearly reveal whether they have effectively captured such spatial context. Consequently, this experiment is the main benchmark we use to evaluate the quality of different approaches.

A.2. Masked Region Reconstruction.

As described in the main paper, this downstream task aims to impute the gene expression values of spots in masked regions by leveraging information from the surrounding spots. In our experimental setup, we set all gene expression values of spots within the masked regions to zero. Consequently, both scGPT [6] and the naive approach of directly using the original gene expression of these spots lack usable information and therefore cannot effectively accomplish this task. In contrast, our model and scGPT-spatial [25] are both capable of generating gene expression values from input features, which allows for a fair comparison between them on this downstream task. By evaluating their performance, we further demonstrate that, even though scGPT-spatial [25] attempts to use information from neighboring spots to compensate for the lack of spatial information caused by its

spot-based sampling strategy, it still falls short of our proposed patch-based approach.

For scGPT-spatial [25], we predict the gene expression of a masked central spot using the gene expression profiles of its k nearest neighboring spots. Specifically, we first identify the k nearest neighbors around the central spot and use their gene expression values as input features. The neighborhood embedding of spot i $h_n'^{(i)}$ is defined as the average spot embedding from the surrounding spots:

$$h_n'^{(i)} = \frac{1}{|S_{knn}^{(i)}|} \sum_{q \in S_{knn}^{(i)}} h_s'^{(q)}, \tag{7}$$

where $S_{knn}^{(i))}$ denotes the set of k nearest neighbors of spot i based on spatial coordinates. And $h_s'^{(q)}$ represent the embedding of spot q. scGPT-spatial [25] learns to predict the query gene expressions of spot i through the neighborhood embedding $h_n'^{(i)}$:

$$\tilde{x}_s^{(i)} = W h_n'^{(i)}, \tag{8}$$

where $\tilde{x}_s^{(i)}$ refer the predicted gene expressions of spot i and W is a a pre-trained linear projection matrix that maps the neighborhood embedding to the gene expression space. For our model, we reconstruct the gene expression values of the masked spots in the mask region by using a SS spot surrounding the masked region containing the mask region. The specific formula is as follows:

$$H' = \text{encoder}(X', G), \tag{9}$$

$$\tilde{X} = \text{decoder}(H'), \tag{10}$$

where $\tilde{X}, X' \in \mathbb{R}^{SS|G|}$ denote the reconstructed gene expression values and the partially masked gene expression values, respectively. G represents the gene ids we selected to reconstruct. H' denotes the embeddings of spots in the mask region. While the encoder and decoder represent the processes through which our model generates embeddings from the gene-expression matrix and gene-id vector, and then uses these embeddings to reconstruct the gene-expression profile.

For the gene-expression values reconstructed by scGPT-spatial [25] or by our model, we compute the Mean Squared Error (MSE) and the Mean Absolute Error (MAE) against the original gene-expression values as performance metrics.

$$\text{MSE} = \frac{1}{|\mathcal{D}_{\text{mask}}||G|} \sum_{i \in \mathcal{D}_{\text{mask}}} \left(x_s^{(i)} - \tilde{x}_s^{(i)}\right)^2, \tag{11}$$

$$\text{MAE} = \frac{1}{|\mathcal{D}_{\text{mask}}||G|} \sum_{i \in \mathcal{D}_{\text{mask}}} \left|x_s^{(i)} - \tilde{x}_s^{(i)}\right|, \tag{12}$$

where $x_s^{(i)}$ and $\tilde{x}_s^{(i)}$ denote the true gene expressions and the predicted gene expressions of spot i ,respectively. And $\mathcal{D}_{\text{mask}}$ represents the masked spot set.

A.3. Downstream Evaluation Datasets.

For downstream assessment, we curated six human ST datasets annotated at either the region or cell-type level. Collectively, these benchmarks encompass diverse tissue architectures and biological complexities, providing a rigorous testbed for evaluating model generalization. The following are the key datasets included:

- The **human dorsolateral prefrontal cortex (DLPFC) [19]** transcriptomics dataset is a widely used human brain benchmark generated with the 10x Genomics Visium platform on postmortem dorsolateral prefrontal cortex (DLPFC) tissue. It contains 12 tissue sections from several donors (commonly three individuals, each contributing four adjacent or nearby sections), and each section covers the full cortical column spanning layers I–VI and the underlying white matter. For each slice, gene expression is measured at roughly 3,600–4,900 spatial spots, with transcript counts available for over 30,000 genes, together with high-resolution H&E histology images. A key feature of this dataset is the expert-curated anatomical annotation: cortical layers and white matter regions are manually labeled based on both histological morphology and known marker genes.
- The **Lymph Node A1 (LNA)** and **Lymph Node D1 (LND)[16]** datasets are are two human lymph node sections profiled with 10x Genomics Visium spatial multi-omics, providing spot-level gene (and protein) expression together with high-resolution histology images. Both slices cover the full lymph node architecture from capsule and cortex down to medullary regions and surrounding adipose tissue, with detailed region labels that make them well suited for benchmarking spatial domain detection and cell-type deconvolution in an immune organ context. In the A1 slice, the annotated structural categories include: medulla sinuses, medulla cords, cortex, pericapsular adipose tissue, follicle, capsule, hilum, medulla vessels, subcapsular sinus, and trabeculae. In the D1 slice, the labeled categories include: Medullary Chords, Medullary Sinus, Paracortex, Adipose Tlssue, B cell Follicle, Cortex, Capsule, Exclude, Marginal Sinus, Endothelial, and Connective Tissue.
- The **Breast Cancer (BrC) [12]** dataset was generated using 10x Visium technology, with manual annotations performed by pathologists based on H&E images and gene expression patterns. This annotation establishes a comprehensive hierarchical classification comprising four main morphological categories—ductal/lobular carcinoma in situ (DCIS/LCIS), healthy tissue, invasive ductal carcinoma (IDC), and tumor edge—which are fur-

Table S1. Masked region reconstruction performance, measured by mean squared error (MSE) and mean absolute error (MAE) for our method and the scGPT-spatial baseline across different masking ratios. Note: scGPT and raw methods cannot perform this task due to lack of spatial information. Size refers to the spatial dimensions (width and height) of pretraining samples in our method.

method	size	DLPFC		LNA		LND		Col		BrC		Ton	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ours	30	**0.243**	**0.187**	0.246	0.196	0.226	0.152	0.335	0.258	0.315	0.257	0.351	0.292
	26	0.256	0.196	0.254	0.200	0.179	0.165	0.334	0.258	0.326	0.265	**0.259**	0.262
	24	0.256	0.191	0.301	0.205	0.183	0.141	**0.312**	**0.248**	**0.285**	**0.244**	0.273	**0.254**
	22	0.252	0.192	0.262	**0.197**	**0.177**	**0.136**	0.321	0.249	0.285	0.247	0.321	0.282
	20	0.246	0.213	**0.212**	0.220	0.179	0.170	0.327	0.279	0.326	0.281	0.265	0.268
scGPT-sp	-	0.320	0.328	0.349	0.356	0.270	0.150	0.384	0.385	0.313	0.281	0.357	0.368

Table S2. Accuracy comparison with SToFM for spatial domain detection using MLP classification on downstream task datasets.

Method	DLPFC	LNA	LND	Col	BrC	Ton	Avg
ours(16)	0.753	0.670	0.635	0.467	0.773	0.674	0.662
sToFM	0.645	0.602	0.555	0.461	0.443	0.658	0.542

ther subdivided into 20 sub-regions. For instance, IDC is partitioned into subclasses (e.g., IDC_1, IDC_2), while the tumor edge encompasses multiple delineated domains (e.g., edge 1–6). This spatially-resolved hierarchical architecture precisely captures intra-tumoral heterogeneity and microenvironmental characteristics. It serves as a robust benchmark for evaluating spatial domain detection methods such as EnSDD across multiple clustering resolutions, and enables biological insights—such as immune cell enrichment—through differential gene expression and cell-type analyses of the sub-regions.

- The **Tonsil (Ton) [31]** dataset captures spatial transcriptomic profiles from three slices of human tonsil tissue, encompassing 13,366 spots classified into four key structural domains—tonsillar parenchyma, lymphoid follicle, germinal center, and connective & epithelial tissue. This secondary lymphoid organ dataset, with its complex architectural organization, serves as an important benchmark for assessing spatial domain detection performance in immune-related tissues.
- The **Colorectal (Col) [23]** dataset consists of 13 colorectal cancer tissue slices profiled using the 10x Genomics Visium spatial transcriptomics platform, each accompanied by detailed pathological annotations provided by expert pathologists. These samples cover multiple anatomical locations along the colorectum—including cecum, right colon, sigmoid colon, and rectum—and exhibit rich clinical diversity in terms of metastatic status (e.g., lymph node or liver metastasis), growth patterns (such as tubular or mucinous), and varying levels of immune cell infiltration, all systematically documented in a clinical information table. At the spatial level, each spot is assigned to one of 17 histopathology-based categories, namely: IC_aggregate_stroma_or_muscularis, IC_aggregate_submucosa, epithelium&submucosa, exclude, muscularis_IC_med_to_high, nan, non_neo_epithelium, stroma_desmoplastic_IC_low, stroma_desmoplastic_IC_med_to_high, stroma_fibroblastic_IC_high, stroma_fibroblastic_IC_low, stroma_fibroblastic_IC_med, submucosa, tumor, tumor&stroma, tumor&stroma_IC_low, and tumor&stroma_IC_med_to_high. Together, these multi-level annotations—spanning anatomy, clinical context, and finely resolved spatial labels—make the Col dataset a valuable resource for studying colorectal tumor heterogeneity, tumor–stroma–immune interactions, and for benchmarking spatial domain detection and cell-type deconvolution methods.

B. Implementations

In this section, we provide more details for the implementations. To ensure consistent and fair comparisons, we standardized the experimental setup for all methods during pretraining and downstream tasks.

B.1. Data preprocessing.

To eliminate library-depth variation and stabilize subsequent training, we additionally preprocess the gene-expression values in the spatial transcriptomics data. Beyond the coordinate normalization shown in Equation 5 of the main paper, we perform per-spot total-count normalization followed by a log-transform. Specifically, for each spot vector $\mathbf{x}_i \in \mathbb{R}^G$ containing the raw UMI counts of $|G|$ genes, the normalization step scales the total counts to the target sum:

$$\tilde{\mathbf{x}}_i = 10000 \cdot \frac{\mathbf{x}_i}{\sum_{g=1}^{|G|} x_{ig}},$$

and the log-transform is then applied element-wise:

$$\mathbf{x}'_{ig} = \log(1 + \tilde{x}_{ig}) \quad \text{for } g = 1, \ldots, |G|.$$

After this preprocessing, all spots share a common scale and an approximately Gaussian-like distribution, providing a stable input for the self-supervised reconstruction task.

B.2. Model Architecture and Pre-training

scGPT. scGPT [6] receives a paired input $(\mathbf{G}, \mathbf{V}) \in \mathbb{R}^{c\times 2}$, where c is the number of detected genes in a spot, $\mathbf{G}$ denotes the gene-ID sequence, and $\mathbf{V}$ represents the corresponding expression values. Each ID is first embedded into a 256-dimensional vector through a learnable lookup table, while each expression value is independently projected to the same 256-dimensional space by a two-layer MLP $(1 \to 256, \mathrm{ReLU}, 256 \to 256)$. The two embeddings are added element-wise to form a gene-specific token. A learnable positional encoding is further added to incorporate gene-order information. The resulting sequence of n tokens is fed into a standard Transformer encoder with 6 layers, 8 attention heads and a hidden dimension of 256. The encoder output is then decoded in a 2-layer MLP head $(256 \to 256 \to 1)$ that reconstructs the original expression value for each gene.

scGPT-spatial. As described in the original paper, scGPT-spatial [25] shares the core architecture of scGPT [6] but introduces key adaptations during training to incorporate spatial information. Specifically, the model employs a spatial sampling strategy and leverages gene expression values from surrounding spots to reconstruct the expression profile of the central spot. To ensure a fair comparison, we adopted the recommended configuration from the original paper: each reconstruction task uses information from the 16 spots immediately surrounding the central spot

Our model. As shown in the main paper, we adopt a masked autoencoder-style architecture to process our patch-based samples. The input is a 3D gene expression tensor $X \in \mathbb{R}^{w\times h\times c}$, where w and h are the spatial dimensions of the patch and c is the number of selected genes. Gene expression values are filled into X according to the spots' relative spatial coordinates.

For each spot, the c-dimensional gene expression vector is first linearly projected to a 256-dimensional representation, which is then added to a learnable gene ID embedding to form the token for that spot. All tokens are further augmented with a learnable 2D positional encoding to encode spatial relationships, yielding a sequence of 256-dimensional tokens.

This sequence is fed into a Transformer encoder with 6 layers, 8 attention heads, and hidden size 256. A Transformer decoder with the same hidden size and number of heads (256, 8) but only 2 layers then reconstructs a 512-dimensional representation for each spot. Finally, a 2-layer MLP head with dimensions $256 \to 512 \to c$ maps the hidden representations back to c-dimensional gene expression vectors. The resulting c-dimensional outputs are used for self-supervised training and for the Masked Region Reconstruction task.

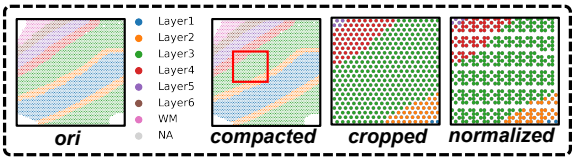

Figure S1. Sample construction — ori (original), compacted (scaled), cropped (fixed-window), normalized (grid-mapped). Visualization: spot size 100 for ori, 1 for others.

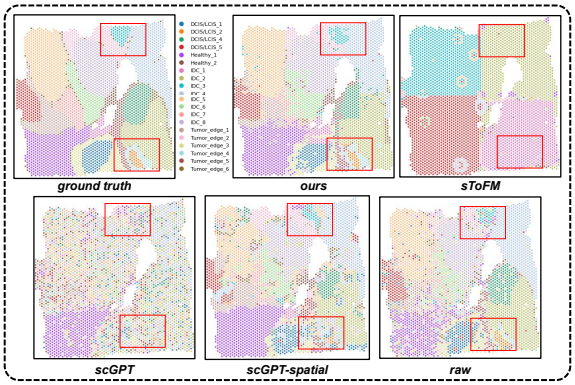

Figure S2. Results visualization of Spatial Domain Detection on BrC — red boxes indicate largest improvements.

Pre-training. To ensure a fair comparison across different sample-construction strategies, we standardize the entire pre-training pipeline for every model. We adopt a *generative self-supervised* objective: randomly masking 30 % of the gene-expression values in *every* spatial spot, and asking the network to reconstruct the original signal from the corrupted input. All experiments share an identical loss—simply the mean-squared-error (MSE) between the predicted and the held-out values—so that progress is measured on the same scale regardless of architecture.

With this fixed protocol, we train every model for 40 epochs. Optimization is performed with AdamW, starting at an initial learning rate of 2×10^{-4} and a weight decay of 0.05. The mini-batch size is set to 32 per GPU; when eight GPUs are used in parallel the effective global batch reaches 256. The first 10 % of total iterations serve as a warm-up, after which the learning rate follows a cosine annealing schedule down to the minimum. By uniformly clipping gradient norms at 1.0 for stability and keeping all hyperparameters constant across settings, we ensure that any performance differences can be directly attributed to variations in sample construction alone.

These settings maximize experimental fairness, allowing the differences arising from various sample-construction strategies to be revealed as clearly as possible.

C. Experiments

Full Tables. In this section, we present Table S1 which reports the detailed results of the Masked Region Reconstruction task. Specifically, we fix the masked region to a 14×14 area and evaluate how well the masked spots' gene expression profiles can be reconstructed when using spot information from different spatial ranges. As in the main experiments, we use mean squared error (MSE) and mean absolute error (MAE) to quantify reconstruction performance. As shown in Table S1, across all datasets—not only the DLPFC dataset—models trained with patch-based samples consistently outperform the spot-based scGPT-spatial baseline, demonstrating the advantage of patch-level training for reconstructing spatial gene expression in masked regions.

We also supplemented the comparative experiments with sToFM (using official weights). Table S2 reports a comparison between our model (trained on 1M spots) and the supplied sToFM model (trained on 88M spots): 0.662 vs. 0.542. These results align with our claim: slice-level sampling rapidly consumes training data and leaves too few samples for comparable performance.

Visualization.Figure S1 shows our coordinate compaction is applied uniformly across all slices to preserve the relative positions of spots. This is a reversible pre-processing step essential for creating dense, learnable image-like representations without altering downstream biological interpretations. **Figure S2** presents the visualization results of the Spatial Domain Detection task on the BrC dataset. The red bounding boxes clearly highlight the improvements achieved by our method.

Further Discussions. Current strategies for sample construction in spatial transcriptomics face a core trade-off: spot-based approaches lose spatial structure, whereas slice-based approaches are computationally expensive and limited in sample size, making it difficult to achieve both spatial fidelity and scalability. Our image-like patch-based framework mitigates this by cropping fixed-size multi-channel patches and applying gene-importance–weighted selection, greatly increasing training sample size while preserving local topology. This balances spatial context and efficiency and provides a standardized, reproducible pretraining paradigm. Looking ahead, promising directions include multi-scale hierarchical modeling, incorporating absolute coordinates and morphological priors, extending to multimodal masking with histology and proteomics, adopting advanced vision architectures such as Swin Transformer, and integrating high-resolution platforms like Stereo-seq together with non-human atlases to move toward general-purpose spatial omics foundation models.