BUILDING FOUNDATION MODELS TO CHARACTERIZE CELLULAR INTERACTIONS VIA GEOMETRIC SELF-SUPERVISED LEARNING ON SPATIAL GENOMICS

Yuning You[1], **Zitong Jerry Wang**[2], **Kevin Fleisher**[1], **Rex Liu**[1], **Matt Thomson**[1]
[1]California Institute of Technology, [2]Westlake University
{ynyou,kfleishe,rex}@caltech.edu, jerry@westlake.edu.cn

ABSTRACT

Cellular interactions form the fundamental/core circuits that drive development, physiology, and disease within tissues. Advances in spatial genomics (SG) and artificial intelligence (AI) offer unprecedented opportunities to computationally analyze and predict the behavior of cell intricate networks, and to identify interactions that drive disease states. However, challenges arise in both *methodology* and *scalability*: **(i)** how to computationally characterize complicated cellular interactions of multi-scale nature, where chemical genes/circuits in individual cells process information and drive interactions among large numbers of diverse cell types, and **(ii)** how to scale up the pipeline to accommodate the increasing volumes of SG data that map transcriptome-scale gene expression and spatial proximity across millions of cells. In this paper, we introduce the **Cellular Interaction Foundation Model** (**CIFM**), an AI foundation model functioning to analyze and simulate cellular interactions within living tissues. In the CIFM pipeline, we explicitly capture and embed interactions of cells within microenvironments by leveraging the powerful and scalable geometric graph neural network model, and optimize the characterization of cellular interactions with a novel self-supervised learning objective – we train it to infer gene expressions of cells based upon their surrounding microenvironments. As a result, we construct CIFM with 100 million parameters by consuming SG data of 23 million cells. Our benchmarking experiments show CIFM effectively infers gene expressions conditional on the microenvironmental contexts: we achieve a high correlation and a low mismatch error, with 71.4% of cells being annotated as the similar cell type based on their predicted and actual expressions on Visium-HD. We demonstrate the downstream utility of CIFM by: (i) applying CIFM to embed tumor samples to capture cellular interactions within tumor microenvironments (ROC-AUC score of 0.862 on classifying sample conditions via linear probing on embeddings), and identifying shared signatures across samples; and (ii) using CIFM to simulate changes in microenvironmental composition in response to T cell infiltration, which highlights how CIFM can be leveraged to model cellular responses to tissue perturbations – an essential step toward constructing "AI virtual tissues". Our model is open source and publicly accessible at `https://huggingface.co/ynyou/CIFM`.

1 INTRODUCTION

The cell is the fundamental unit of life, and the cellular communication/interaction establishes the cell-level circuits of living functions – it is indisputably critical for all diseases, from cancer (Armingol et al., 2021) and autoimmune diseases (Noack & Miossec, 2021) to aging (Sun et al., 2024) and normal physiology (Sies et al., 2022). The scaling of spatial genomics (SG) data (Eng et al., 2019) and advancements in artificial intelligence (AI) techniques (Bunne et al., 2024) offer unprecedented opportunities to construct computational models to characterize cellular interactions, while current developments do not fully unleash the potential of AI and big SG data: (i) the ineffectiveness of the simplistic model architecture at modeling intricate cellular interaction circuits (e.g. naïve averaging via convolution to represent interactions (Hu et al., 2021)) which are of multi-scale nature and

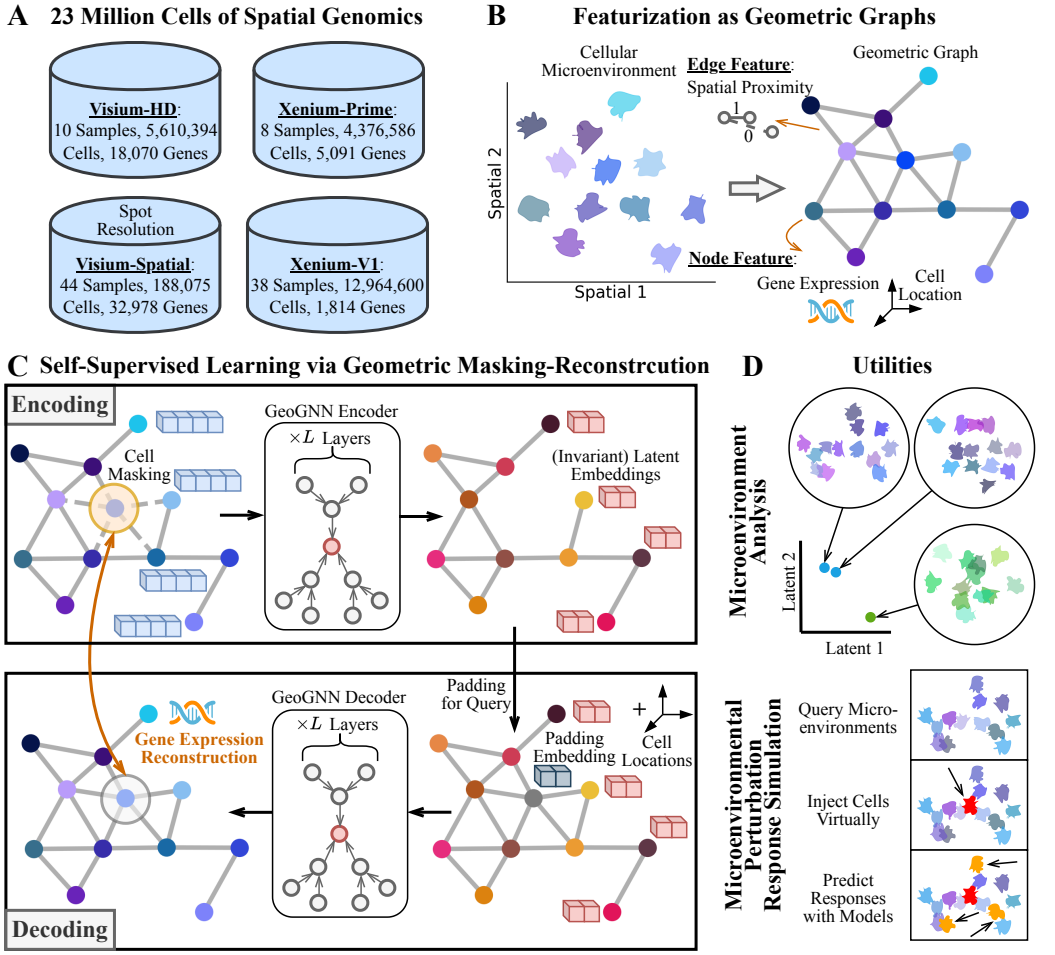

Figure 1: Overview of the CIFM pipeline. (A) SG data with around 23 million cells of four platforms are curated from the 10x Genomics database. (B) A CME is featurized as a geometric graph with node features of gene expressions and cell locations, and edge features of spatial proximity. (C) CIFM is trained to reconstruct the gene expressions of masked cells based on their microenvironmental contexts. (D) The utility of CIFM is demonstrated in the examples of microenvirionment analysis and perturbation response simulation.

inherent complexity: in the real-world systems, each cell acts as a node that processes information and changes state, interacting with other nodes through chemical communication, leading to intricate cell interaction networks, where cellular nodes communicate to compute, process information, make decisions, and execute state transitions; and (ii) the challenge in scaling up to accommodate the increasing volumes of SG data that map transcriptome-scale gene expression and spatial proximity across millions of cells, spanning large sections of tissue, organ systems, and disease states (Wang et al., 2025b). We defer the discussion of more related works to Appdx. A.

In this paper, we aim to build a larger-scale model capable of characterizing cellular interactions to consume massive SG data from varied platforms/sources, in order to function to not only analyze but more importantly, computationally simulate these interactions – an initiative toward constructing "AI virtual tissues". To this end, we leverage the advanced AI model of geometric graph neural networks (GeoGNNs) (Satorras et al., 2021; Joshi et al., 2023) – the powerful and scalable model able to explicitly capture and embed interactions within cellular microenvironments (CMEs) via geometric message passing. To optimize the characterization of cell interactions, we develop and implement a novel self-supervised learning pipeline to train GeoGNNs on the vast SG data – around 100 SG samples with 23 million cells and 32 thousand measured genes of four platforms of Visium and Xenium. We refer our model as the cellular interaction foundation model (CIFM).

Our self-supervised learning pipeline is designed in a masking-reconstruction manner on the SG data of geometric structures. Intuitively, our model is optimized to reconstruct the masked gene expressions of cells based on their microenvironmental contexts. Such a self-supervised learning task is highly intriguing in living systems for the reasons: (i) it requires the model to capture the important cellular interactions within CMEs to deliver the best inference for masked cells, and (ii) the task itself holds significant value in computational simulations of the systems in numerous applications, for instance, it can simulate or "hallucinate" cell state variations in response to tissue-level perturbations in their CMEs.

We benchmark CIFM and demonstrate that it effectively infers CME context-dependent gene expressions, achieving high correlations and low mismatch errors across varied samples, platforms, and scenarios, including in-sample, cross-sample, and even cross-platform evaluation (e.g., training on Visium and zero-shot evaluation on Xenium). For instance, in the Visium-HD dataset, 71.4% of cells are annotated with the top-5 cell typing profile containing shared cell types based on their predicted and actual expressions. Furthermore, we illustrate the downstream utility of CIFM with two examples. In the first scenario, we use CIFM embeddings to analyze the CME configurations of different samples. The embeddings well-capture distinguished cellular interactions in healthy and tumor samples, achieving an ROC-AUC score of 0.862 when classifying sample conditions via linear probing. Furthermore, we find that CMEs with highly similar CIFM embeddings – indicative of shared microenvironmental features across tumor samples – are consistently enriched with plasma cells. This suggests that microenvironments with immune responses containing plasma cells commonly appear in different tumor types, orchestrating cell-cell communication in tumor-immune microenvironments. In the second scenario, we apply CIFM to autoregressively infer CME changes during immune cell infiltration. After the virtual injection of T cells into the breast tumor sample, we observe a notable decrease in the population of luminal epithelial cells – a cell type highly associated with breast tumors – indicating the potential elimination of tumor cells.

2 CIFM IS TRAINED TO RECONSTRUCT THE GENE EXPRESSIONS OF MASKED CELLS BASED ON THEIR MICROENVIRONMENTAL CONTEXTS

The CIFM workflow consists of three parts: (i) geometric graph featurization of CMEs (Fig. 1B), (ii) masking and encoding/embedding of geometric graphs (Fig. 1C Top), and (iii) padding and decoding/reconstruction of geometric graph features (Fig. 1C Bot). In the foremost step (i), we featurize CMEs in order to feed them into neural networks (Fig. 1B, Appdx. B.1). Considering the ith CME (out of K, i.e. $i \in \{1, ..., K\}$) containing N_i cells with M_i measured genes, the featurized geometric graph is denoted as:

$$\text{Featurization:} \quad G_i = \Big\{ \overbrace{\boldsymbol{X}_i}^{\text{Genes}}, \overbrace{\boldsymbol{C}_i}^{\text{Locations}}, \overbrace{\boldsymbol{A}(\boldsymbol{C}_i)}^{\text{Connectivity}} \Big\}, \quad i \in \{1, ..., K\}, \tag{1}$$

where $\boldsymbol{X}_i \in \mathbb{R}_{\geq 0}^{N_i \times M_i}$ is the node feature matrix of gene expressions, $\boldsymbol{C}_i \in \mathbb{R}^{N_i \times 2}$ is the coordinate matrix, and $\boldsymbol{A}(\boldsymbol{C}_i) \in \{0, 1\}^{N_i \times N_i}$ is the adjacency matrix constructed based on the spatial proximity of the coordinates. To collect adequate CMEs for model training, we curate data from the 10x Genomics database (Rao et al., 2020; Janesick et al., 2023) (Fig. 1A, Appdx. B.1), a publicly available and extensive resource of spatial genomics data from four different platforms (Visium-Spatial/HD, Xenium-V1/Prime), containing approximately 100 slides of healthy/tumor samples of varied organs, with around 23 million cells and 32 thousand measured genes.

With the featurized geometric graphs, we train CIFM in a self-supervised manner via masking-reconstruction. In the step (ii), we randomly and uniformly remove nodes within the geometric graph and then use a GeoGNN encoder $f_{\text{enc};\theta}(\cdot)$ to embed the remaining structure into the latent space (Fig. 1C Top, Appdx. B.2). In the step (iii), we pad the removed nodes with the learnable padding embeddings $\boldsymbol{e}$, return them to the geometric graph at their original locations, and then use a GeoGNN decoder $f_{\text{dec};\phi}(\cdot)$ to reconstruct their masked gene expressions (Fig. 1C Bot, Appdx. B.2). We optimize the GeoGNN encoder/decoder by minimizing the mismatch between the masked and

reconstructed gene expressions, formulated as:

$$\text{Masking:} \quad \boldsymbol{X}_{\text{unm},i} \oplus_{(2)} \boldsymbol{C}_{\text{unm},i} = \overbrace{\mathrm{Rmv}(\boldsymbol{X}_i, \boldsymbol{C}_i, \mathbb{I}_i)}^{\text{Remove Nodes}}, \quad \boldsymbol{Z}_{\text{enc},i} = \overbrace{f_{\text{enc};\theta}\Big(\boldsymbol{X}_{\text{unm},i}, \boldsymbol{C}_{\text{unm},i}, \boldsymbol{A}(\boldsymbol{C}_{\text{unm},i})\Big)}^{\text{Embed Remaining Structure}}; \tag{2}$$

$$\text{Reconstruction:} \quad \boldsymbol{Z}_{\text{pad},i} = \underbrace{\mathrm{Pad}(\boldsymbol{Z}_{\text{enc},i}, \boldsymbol{e}, \mathbb{I}_i)}_{\text{Pad Masked Nodes}}, \quad \boldsymbol{X}_{\text{dec},i} = \underbrace{f_{\text{dec};\theta}\Big(\boldsymbol{Z}_{\text{pad},i}, \boldsymbol{C}_i, \boldsymbol{A}(\boldsymbol{C}_i)\Big)}_{\text{Reconstruct Masked Features}}; \tag{3}$$

$$\text{Optimization:} \quad \min_{\theta,\phi,\boldsymbol{e}} \quad \frac{1}{K}\sum_{i=1}^{K} \mathrm{Loss}\Big(\underbrace{\boldsymbol{X}_i[\mathbb{I}_i]}_{\text{Masked Node Features}}, \boldsymbol{X}_{\text{dec},i}[\mathbb{I}_i]\Big), \tag{4}$$

where umn is short for unmasked, $\oplus_{(2)}$ is the concatenation alone the 2nd dimension (feature dimension), $\mathrm{Rmv}(\cdot), \mathrm{Pad}(\cdot)$ are the removal and padding functions on geometric graphs, respectively, $\mathbb{I}_i$ is the set of masking indices of the ith CME, $\boldsymbol{X}[\mathbb{I}]$ denotes matrix indexing, and $\mathrm{Loss}(\cdot)$ is the loss function. We illustrate the utility of CIFM through two examples, and its capabilities extend beyond then. In the first scenario, we use the embeddings of CIFM to analyze the CME configurations of tumor samples (Fig. 1D Top). In the second scenario, we apply CIFM to autoregressively infer changes in the cell states of CMEs after virtually injecting immune cells (Fig. 1D Bot).

3 CIFM EFFECTIVELY INFERS CONTEXT-DEPENDENT GENE EXPRESSIONS ACROSS VARIED SPATIAL GENOMICS SAMPLES AND PLATFORMS

We assess CIFM by evaluating its precision in inferring the gene expressions of cells, based on the context of their neighboring cells. For benchmarking, we split each slide of SG samples into training, validation, and test regions for model training, hyperparameter tuning, and evaluation, respectively (Fig. 2A). We refer the evaluation with the regional split as in-sample evaluation. We also evaluate using the sample split beyond the regional split, where samples are randomly held out for evaluation (Appdx. C), referred as cross-sample evaluation. We compared CIFM against baselines that include random expressions following parameterized uniform and Bernoulli distributions (with parameters learned from the data), as well as a naïve neighborhood average approach that computes the mean expressions of the neighboring cells.

We first evaluate CIFM on the Visium-HD dataset (Fig. 2B, Appdx. C). The evaluation is performed using metrics of correlation and mismatch error: correlation assesses whether the model ranks gene expression correctly, while mismatch error measures how accurately it infers the actual values of gene expression. In addition to metrics of correlation and mismatch error, we utilize the neural-network based cell typing tool scTab (Fischer et al., 2024) and SCimilarity (Heimberg et al., 2024) to determine whether reconstructed gene expression is mapped to the same cell type as mapped with the masked gene expression, a measure we term cell typing accuracy. This metric evaluates whether CIFM infers expressions of the most important genes that is critical in determining cell types. We observe that CIFM consistently achieves the best performance across 10 slides in all three metrics. We obtain similar observation when evaluating on their 1,000 most differentially expressed genes per sample (Appdx. C). Since the cell typing accuracy may be affected by the precision of the cell typing tool itself, we relax the stringency of the cell typing evaluation by considering whether the predicted top-k cell types overlap between predictions based on masked and reconstructed gene expressions (Fig. 2D, Appdx. C). We observe CIFM is able to achieve around 71.4% of SCimilarity accuracy on average when $k = 10$ (across a total of around 200 cell type labels).

We next assess the transferability of CIFM by evaluating it on the Xenium-V1 and Xenium-Prime datasets while training on Visium-HD (Fig. 2E), referred as cross-platform evaluation. In spite of the substantial technical differences between Visium (sequencing-based) and Xenium (imaging-based), CIFM demonstrates effective gene expression inference. In the Xenium-Prime samples, it consistently achieves the highest correlations and lowest mismatch errors, and outperforms in cell typing accuracy for 6 out of 8 slides, and the performance advantage further enhanced after finetuning. However, it underperforms in the Xenium-V1 samples (Appdx. C), possibly due to the

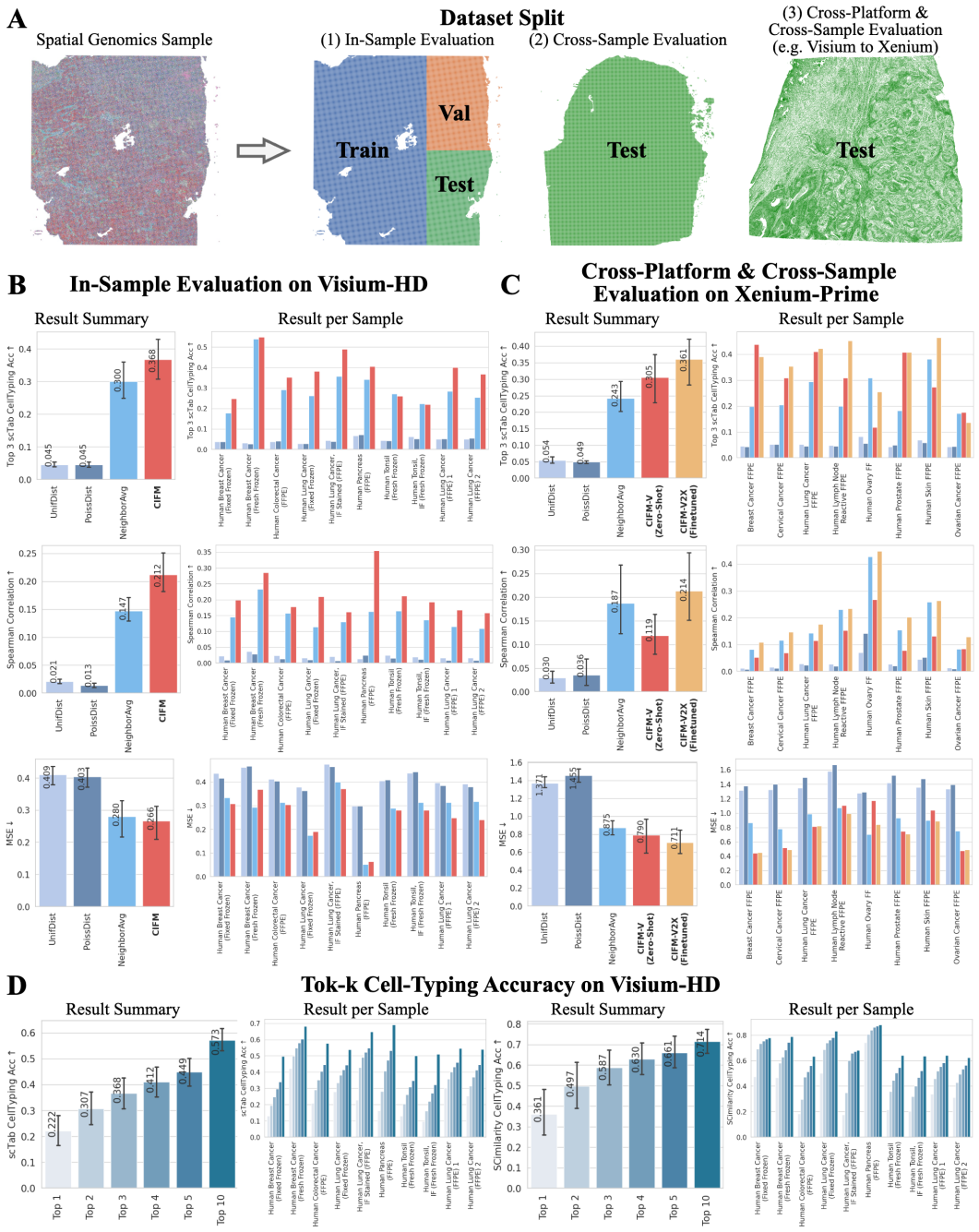

Figure 2: Benchmarking of the CIFM performance. (A) Dataset split for in-sample evaluation, cross-sample evaluation and cross-platform evaluation. (B) In-sample evaluation on the Visium-HD dataset. (C) Cross-platform & cross-sample evaluation on the Xenium-Prime dataset. (D) Evaluation on top-k cell typing accuracy of CIFM on the Visium-HD dataset.

huge discrepancies in the gene measurement scale: around 17K genes on average in Visium-HD, 300 genes in Xenium-V1, and 5K genes in Xenium-Prime (Fig. 1A). This can be remedied with further finetuning: we further finetune CIFM on Xenium-V1, which results in the best correlation across all 38 slides (Appdx. C).

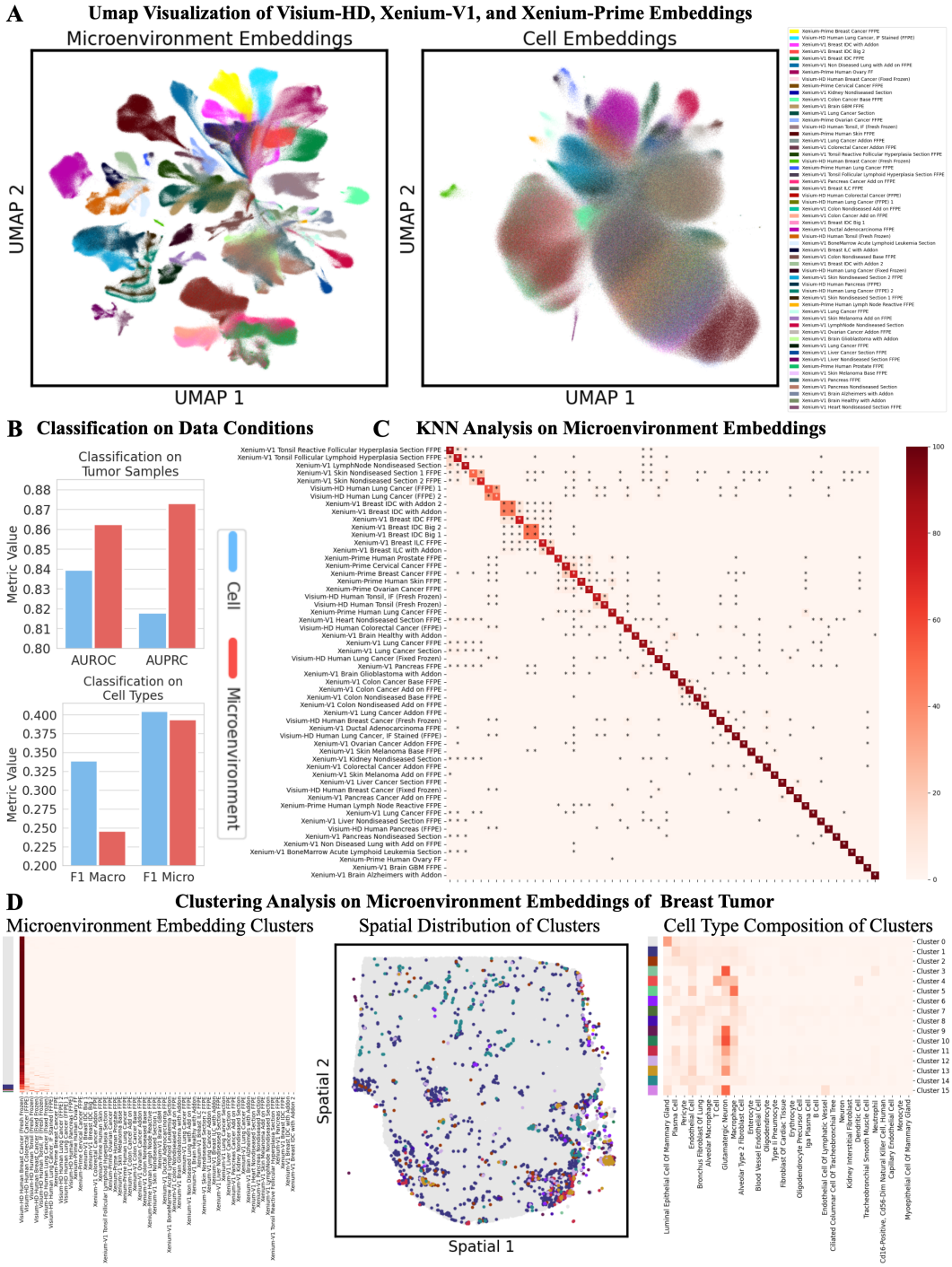

Figure 3: CME embedding with CIFM. (A) Lower-dimensional visualization of CIFM embeddings of microenvironments and cells on Visium-HD, Xenium-V1, and Xenium-Prime datasets. (B) Linear probing on CIFM embeddings to classify tumor vs non-tumor samples, and scTab annotated cell types. (C) The fraction (%) of the 100 nearest neighbors queried with CIFM microenvironmental embeddings across different samples. The value is averaged across all CME embeddings of the entire sample. $*$ denotes a value greater than 0.1. (D) The (clustered) KNN profile for each CME in the breast tumor sample without averaging, spatial distribution of each CME cluster, and the cell type composition of each cluster. The colorbar is shared with panel (C).

4 CIFM IS ABLE TO EMBED CMEs AND CHARACTERIZE MICROENVIRONMENTAL PATTERNS ACROSS TUMOR SAMPLES

Since the encoding process of CIFM functions to map the information of CME contexts into latent embeddings, we ask the question that how are the CMEs of different samples distributed within such latent space? We visualize the CME embeddings of Visium and Xenium from CIFM in a lower-dimensional UMAP space (Fig. 3A). The microenvironment embeddings are directly acquired from the GeoGNN encoder, and the cell embeddings are acquired by removing all the edges before fed into the encoder, so that they contain no neighborhood information. We observe that in the geometries of these two embedding landscapes are very different – there existing high heterogeneity across tumor samples indicating the sample-specific heterogeneity within CMEs; while there also existing interesting clusters of closely located points indicating shared CMEs across samples.

To quantify how well the microenvironment-level embeddings of CIFM capture tumor conditions, compared between microenvironment and cell embeddings, we perform linear probing to classify tumor and non-tumor samples, and also scTab annotated cell types (Fig. 3B). Specifically, we train a linear classifier on Visium-HD embeddings and test it on Xenium-Prime sample embeddings on the specific tasks. We observe that microenvironment embeddings surpass in distinguishing tumor conditions and cell embeddings in cell types. Furthermore, even this simple linear model on microenvironment embeddings achieves decent classification performance in distinguishing tumor from non-tumor samples (ROC-AUC score of 0.862) evaluated on cross-platform data.

To quantify the similarity of CME profiles across samples, we performed a KNN analysis on the CME embeddings, where we ask the question that within the k nearest neighbors of each CME embedding, what fraction of these neighbors comes from different samples? As a result, each cell is assigned a KNN fraction vector, where the vector's length equals the number of samples, its values range from 0 to 100, and its entries sum to 100. We average all the KNN fraction vectors and visualize the sample-level profile (Fig. 3C). Notably, the diagonal blocks capture the shared CMEs within the same tumor types across different samples, and more interestingly the off-diagonal entries capture shared CMEs across different tumor types.

Building upon the above observation, we further investigate a more specific question that what is the cell type composition of these shared CMEs across tumor samples? To answer this question, we retrieve the CME-level KNN profiles (without averaging on samples) and perform clustering (Fig. 3D Left), and then we annotate the cell type composition within each cluster using scTab (Fischer et al., 2024) (Fig. 3D Right). When we cluster the KNN profile of CMEs from a breast tumor sample, we observe that while a large proportion of the CMEs are unique to this particular sample (in gray), there exist a set of CMEs that are appear to be shared across other tumor samples (shown in other colors), e.g. colorectal cancer samples. Furthermore, these shared CMEs across tumor samples are enriched with plasma cells, with their spatial visualization provided (Fig. 3D Mid).

5 CIFM IS ABLE TO SIMULATE CME CHANGES IN RESPONSE TO MICROENVIRONMENTAL PERTURBATIONS AUTOREGRESSIVELY

Since the decoding process of CIFM maps CME latent embeddings to the potential gene expression profiles associated with CMEs, we ask the question whether we can simulate the evolution of CMEs by iteratively updating their cells? We initialize this process by first virtually injecting cells of interest into CMEs as microenvironmental perturbations, and collect the model outputs that represent the simulated cellular responses to these perturbations. We inject a fixed number of T cells at random locations as perturbations. We visualize the simulation on a breast tumor sample (Fig. 4A). It is important to note that this simulation pipeline is highly general and not limited to specific samples or perturbations. We observe a increase in various types of immune cells across the three examples, such as memory B cells. By summarizing changes in cell type composition across a large number CMEs, we observe a notable decrease in the population of luminal epithelial cells – a cell type highly associated with breast tumors – indicating the potential elimination of tumor cells (Fig. 4B). We also observe other variations (both increases and decreases) in the populations of endothelial, neuron and fibroblast cells. We further visualize the cell type transition matrix in Fig. 4C.

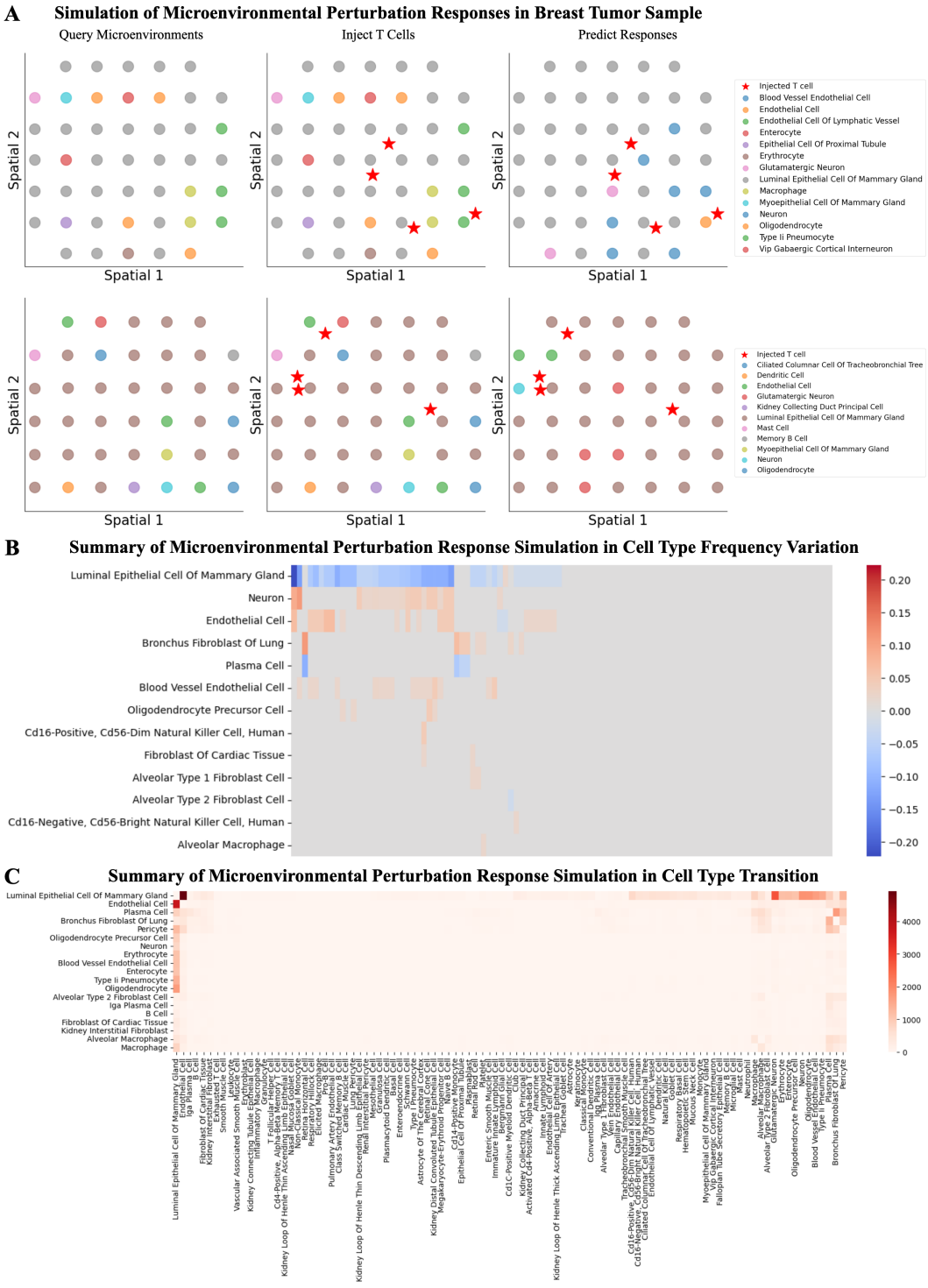

Figure 4: CME response simulation to perturbation with CIFM. (A) Three examples (rows) of the perturbation response simulation in the breast tumor sample. column 1: querying CMEs from samples; column 2: virtually injecting T cells at random locations within CMEs; column 3: masking and predicting gene expressions autoregressively for all cells in the CMEs. (B) Summary of the cell state change in simulation in cell type frequency variation. (C) Summary of the cell state change in simulation in cell type transition.

6 DISCUSSION

With the emerging interest in demystifying cellular interactions related to various phenotypes, ranging from aging to cancer, there is a growing demand for computational models designed for this purpose. Advances in artificial intelligence (AI), a powerful data-driven approach, alongside the increasing quality and quantity of single-cell genomic (SG) data, present an exciting opportunity to develop such models. We develop CIFM here to fully unleash the potential of both modeling and data. Our large-scale model captures interactions across cells and is trained on massive SG datasets. The training process is carefully designed and novel, leveraging cellular interactions to infer missing gene expressions – a feature that enables in silico simulation of living systems. We believe CIFM represents an important step toward constructing the ultimate biological digital twin, or what could be termed "AI virtual tissue" in the future.

ACKNOWLEDGMENTS

This project was partially supported by the Chan Zuckerberg Initiative, the NIH, and the Orr Family Foundation.

REFERENCES

Erick Armingol, Adam Officer, Olivier Harismendy, and Nathan E Lewis. Deciphering cell–cell interactions and communication from gene expression. *Nature Reviews Genetics*, 22(2):71–88, 2021.

Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B Burkhardt, et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024.

Ao Chen, Sha Liao, Mengnan Cheng, Kailong Ma, Liang Wu, Yiwei Lai, Xiaojie Qiu, Jin Yang, Jiangshan Xu, Shijie Hao, et al. Spatiotemporal transcriptomic atlas of mouse organogenesis using dna nanoball-patterned arrays. *Cell*, 185(10):1777–1792, 2022.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexandre Duval, Simon V Mathis, Chaitanya K Joshi, Victor Schmidt, Santiago Miret, Fragkiskos D Malliaros, Taco Cohen, Pietro Lio, Yoshua Bengio, and Michael Bronstein. A hitchhiker's guide to geometric gnns for 3d atomic systems. *arXiv preprint arXiv:2312.07511*, 2023.

Chee-Huat Linus Eng, Michael Lawson, Qian Zhu, Ruben Dries, Noushin Koulena, Yodai Takei, Jina Yun, Christopher Cronin, Christoph Karp, Guo-Cheng Yuan, et al. Transcriptome-scale super-resolved imaging in tissues by rna seqfish+. *Nature*, 568(7751):235–239, 2019.

Felix Fischer, David S Fischer, Roman Mukhin, Andrey Isaev, Evan Biederstedt, Alexandra-Chloé Villani, and Fabian J Theis. sctab: scaling cross-tissue single-cell annotation models. *Nature Communications*, 15(1):6611, 2024.

Graham Heimberg, Tony Kuo, Daryle J DePianto, Omar Salem, Tobias Heigl, Nathaniel Diamant, Gabriele Scalia, Tommaso Biancalani, Shannon J Turley, Jason R Rock, et al. A cell atlas foundation model for scalable search of similar human cells. *Nature*, pp. 1–3, 2024.

Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J Irwin, Edward B Lee, Russell T Shinohara, and Mingyao Li. Spagcn: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature methods*, 18(11):1342–1351, 2021.

Amanda Janesick, Robert Shelansky, Andrew D Gottscho, Florian Wagner, Stephen R Williams, Morgane Rouault, Ghezal Beliakoff, Carolyn A Morrison, Michelli F Oliveira, Jordan T Sicherman, et al. High resolution mapping of the tumor microenvironment using integrated single-cell, spatial and in situ analysis. *Nature Communications*, 14(1):8353, 2023.

Chaitanya K Joshi, Cristian Bodnar, Simon V Mathis, Taco Cohen, and Pietro Lio. On the expressive power of geometric graph neural networks. In *International conference on machine learning*, pp. 15330–15355. PMLR, 2023.

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

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021.

Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and S Yu Philip. Graph self-supervised learning: A survey. *IEEE transactions on knowledge and data engineering*, 35(6): 5879–5900, 2022.

Mélissa Noack and Pierre Miossec. Importance of lymphocyte–stromal cell interactions in autoimmune and inflammatory rheumatic diseases. *Nature Reviews Rheumatology*, 17(9):550–564, 2021.

Noetik. Simulating spatial biology with virtual cells and cellular systems. *Technical Report*, 2024.

Nikhil Rao, Sheila Clark, and Olivia Habern. Bridging genomics and tissue pathology: 10x genomics explores new frontiers with the visium spatial gene expression solution. *Genetic Engineering & Biotechnology News*, 40(2):50–51, 2020.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Helmut Sies, Vsevolod V Belousov, Navdeep S Chandel, Michael J Davies, Dean P Jones, Giovanni E Mann, Michael P Murphy, Masayuki Yamamoto, and Christine Winterbourn. Defining roles of specific reactive oxygen species (ros) in cell biology and physiology. *Nature reviews Molecular cell biology*, 23(7):499–515, 2022.

Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò. 3d infomax improves gnns for molecular property prediction. In *International Conference on Machine Learning*, pp. 20479–20502. PMLR, 2022.

Eric D Sun, Olivia Y Zhou, Max Hauptschein, Nimrod Rappoport, Lucy Xu, Paloma Navarro Negredo, Ling Liu, Thomas A Rando, James Zou, and Anne Brunet. Spatial transcriptomic clocks reveal cell proximity effects in brain ageing. *Nature*, pp. 1–12, 2024.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Chloe Xueqi Wang, Haotian Cui, Andrew Hanzhuo Zhang, Ronald Xie, Hani Goodarzi, and Bo Wang. scgpt-spatial: Continual pretraining of single-cell foundation model for spatial transcriptomics. *bioRxiv*, pp. 2025–02, 2025a.

Zitong Jerry Wang, Abdullah S Farooq, Yu-Jen Chen, Aman Bhargava, Alexander M Xu, and Matt W Thomson. Identifying perturbations that boost t-cell infiltration into tumours via counterfactual learning of their spatial proteomic profiles. *Nature Biomedical Engineering*, pp. 1–15, 2025b.

Johann Wenckstern, Eeshaan Jain, Kiril Vasilev, Matteo Pariset, Andreas Wicki, Gabriele Gut, and Charlotte Bunne. Ai-powered virtual tissues from spatial proteomics for clinical diagnostics and biomedical discovery. *arXiv preprint arXiv:2501.06039*, 2025.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. Self-supervised learning of graph neural networks: A unified review. *IEEE transactions on pattern analysis and machine intelligence*, 45(2):2412–2429, 2022.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823, 2020a.

Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. When does self-supervision help graph convolutional networks? In *international conference on machine learning*, pp. 10871–10880. PMLR, 2020b.

Meng Zhang, Stephen W Eichhorn, Brian Zingg, Zizhen Yao, Kaelan Cotter, Hongkui Zeng, Hongwei Dong, and Xiaowei Zhuang. Spatially resolved cell atlas of the mouse primary motor cortex by merfish. *Nature*, 598(7879):137–143, 2021.

APPENDIX

A RELATED WORKS

Spatial genomics. Spatial genomics is a rapidly evolving biotechnology that bridges the gap between molecular profiling and spatial context, enabling the study of gene expression within the native tissue architecture. Unlike traditional transcriptomic techniques that analyze dissociated cells and lose spatial information, spatial genomics preserves the spatial relationships between cells, offering unprecedented insights into cellular function, tissue organization, and microenvironmental interactions. Recent technological advances have significantly enhanced our ability to generate multiplexed tissue imaging data that captures cellular neighborhoods within intact tissues. A wide range of platforms, including sequencing-based methods (e.g., Visium (Rao et al., 2020), Stereo-seq (Chen et al., 2022)) and imaging-based approaches (e.g., Xenium, seqFISH (Eng et al., 2019), MERFISH (Zhang et al., 2021)), now allow for the simultaneous measurement of thousands of genes with high spatial precision. Moreover, the rapid development of new platforms and methods continues to increase the diversity and complexity of spatial genomics data, enabling deeper insights into different biological conditions.

AI virtual tissue. AI-powered virtual tissue models, a brand-new concept, represent an exciting intersection of artificial intelligence and biological research, enabling the analysis of tissue structure and function with unprecedented precision (Bunne et al., 2024). These models leverage advanced machine learning algorithms to analyze tissue-level data such as spatial genomics, transcriptomics, proteomics, and imaging. Compared to recent concurrent works with a similar perspective (Wenckstern et al., 2025; Noetik, 2024; Wang et al., 2025a) that are based on transformer architectures, our CIFM demonstrates superior scalability due to the unique advantages of geometric graph neural networks.

Geometric graph neural networks. Geometric graph neural networks (GeoGNNs) are an extension of traditional graph neural networks (GNNs) (Kipf & Welling, 2016; Veličković et al., 2017), designed to process data that resides on geometric structures like manifolds, point clouds, or other spatially embedded graphs, enabling superior performance in various domains such as computational biology, physics simulations, robotics, and computer vision (Duval et al., 2023; Satorras et al., 2021). Compared to GNNs, GeoGNNs have the advantage of better expressiveness. They leverage the geometric and topological properties of data (Joshi et al., 2023). Compared to transformers (Vaswani, 2017; Devlin, 2018), GeoGNNs are more scalable, as they employ local message passing rather than global (fully connected) mechanisms.

(Geometric) graph self-supervised learning. Self-supervised learning on graphs is shown to learn more generalizable, transferable and robust graph representations, through exploiting vast unlabelled data (Xie et al., 2022; Liu et al., 2022). The main advantage of self-supervised learning is that it does not require any human annotation which is usually resource-intensive in biology, enabling the utilization of rich data resources to their fullest potential (You et al., 2020a;b). Recent efforts have also extended this technique to geometric graphs, demonstrating its effectiveness in various applications, such as molecular representation learning (Liu et al., 2021; Stärk et al., 2022).

B EXTENDED METHODOLOGICAL DETAILS

B.1 SPATIAL GENOMICS DATA PREPROCESSING AND GEOMETRIC GRAPH FEATURIZATION

We download the raw count SG data of Visium and Xenium from 10x Genomics (`https://www.10xgenomics.com/datasets`), using filters "Visium Spatial" and "Xenium In Situ" under Platform, and "Human" under Species. We read the raw counts with coordinates measured in micrometers, and their metadata into the AnnData format (Wolf et al., 2018). We remove mitochondrial genes, void cells (those without detected gene expression), and retain only cells annotated as "in tissue" based on their histological images, and then normalize gene counts and conduct log1p-transformation. For Visium-HD data, we select a bin size of $8\times8\mu$m as recommended. We eventually obtain 100 slides, with 23,139,655 cells and 32,986 measured genes (Fig. 5A).

For benchmarking our approach, we split each slide into training, validation, and test regions by segmenting it. We define the segmentation rule as follows: using a slide-specific threshold along

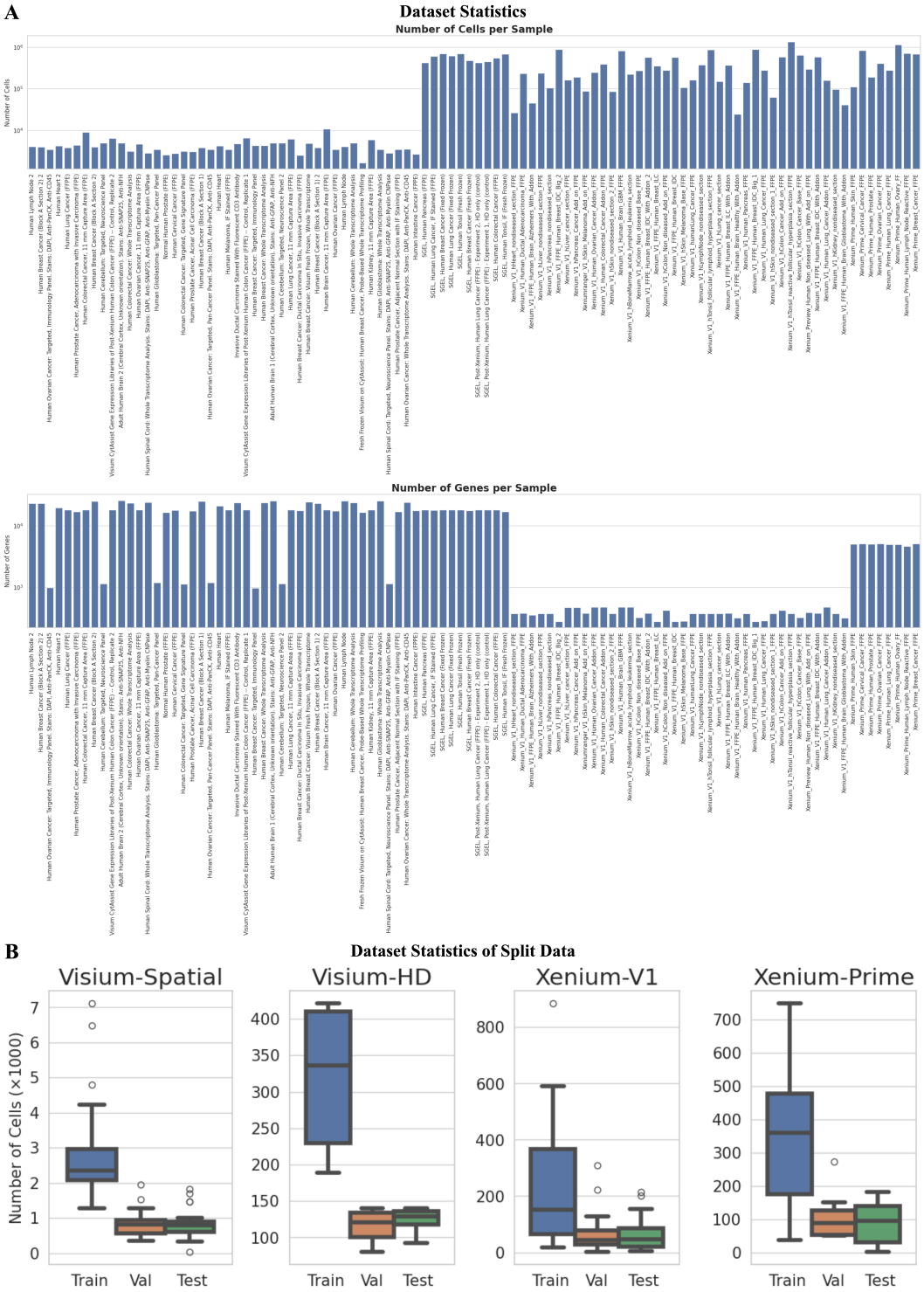

Figure 5: Dataset statistics. (A) Number of cells and measured genes per sample in the Visium and Xenium datasets. (B) Number of cells in the training, validation and test in the regional split.

the x- and y-axes as $x_{\text{thres}}, y_{\text{thres}}$, we allocate a cell with coordinates (x, y) to the training data if $x \leq x_{\text{thres}}$; to the validation data if $x > x_{\text{thres}}$ and $y \geq y_{\text{thres}}$; and to the test data if $x > x_{\text{thres}}$ and $y < y_{\text{thres}}$. This unbiased split allows us to preserve the CMEs of each cell as much as possible.

We compute the thresholds as follows: given the minimum and maximum coordinates of each slide along the x- and y-axes as $x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}}$, we calculate $x_{\text{thres}} = x_{\text{min}} + (x_{\text{max}} - x_{\text{min}}) \times 0.6$, and $y_{\text{thres}} = y_{\text{min}} + (y_{\text{max}} - y_{\text{min}}) \times 0.5$. The number of cells in training, validation, and testing regions follows an approximate 3:1:1 ratio in each slide (Fig. 5B).

With the above preprocessing, we obtain multiple slides of SG data containing information about the expressions and locations of cells. For the ith slide containing N_i cells and M_i measured genes, we featurize it into a geometric graph $G_i = \{\boldsymbol{X}_i, \boldsymbol{C}_i, \boldsymbol{A}_i(\boldsymbol{C}_i)\}$: $\boldsymbol{X}_i \in \mathbb{R}_{\geq 0}^{N_i \times M_i}$, the node feature matrix, is derived from gene expressions; $\boldsymbol{C}_i \in \mathbb{R}^{N_i \times 2}$, the geometric feature matrix, is derived from spatial locations; and $\boldsymbol{A}_i(\boldsymbol{C}_i) \in \{0,1\}^{N_i \times M_i}$, the adjacency matrix, is constructed from $\boldsymbol{C}_i$ to capture spatial proximity information. We build a radius graph for the adjacency matrix that given the radius threshold r_{thres}, the adjacency matrix is computed as: $\boldsymbol{A}_i(\boldsymbol{C}_i)[j,k] = \begin{cases} 1, \text{ if } \|\boldsymbol{C}_i[j] - \boldsymbol{C}_i[k]\| \leq r_{\text{thres}} \\ 0, \text{ otherwise} \end{cases}$, with the radius threshold r_{thres} set to 20μm for Visium-HD, Xenium-V1 and Xenium-Prime and 150μm for Visium-Spatial of spot resolution.

B.2 Geometric Graph Neural Network Encoder and Decoder

GeoGNNs are capable of processing geometric data while respecting their inherent symmetry of permutation and spatial transformations (Satorras et al., 2021; Joshi et al., 2023), which is important for cellular systems as biological functions remain unaffected by permutations in cell features or by global transformations (of rotation and translation) in location. Specifically, a GeoGNN $f(\cdot)$ takes a geometric graph as input and outputs a $(D+2)$-dimensional invariant and equivariant embedding for each node (cell) as $\boldsymbol{Z}_{\text{inv}} \oplus_{(2)} \boldsymbol{Z}_{\text{eqv}} = f(\boldsymbol{X}, \boldsymbol{C}, \boldsymbol{A}(\boldsymbol{C}))$, where $\oplus_{(2)}$ is the concatenation alone the 2nd dimension (feature dimension), $\boldsymbol{Z}_{\text{inv}} \in \mathbb{R}^{N \times D}$ is the invariant embedding matrix, and $\boldsymbol{Z}_{\text{eqv}} \in \mathbb{R}^{N \times 2}$ the equivariant embedding matrix. It respects the symmetry via strictly satisfying:

$$\text{Permutation Symmetry:} \quad \boldsymbol{P}(\boldsymbol{Z}_{\text{inv}} \oplus_{(2)} \boldsymbol{Z}_{\text{eqv}}) = f\Big(\boldsymbol{P}\boldsymbol{X}, \boldsymbol{P}\boldsymbol{C}, \boldsymbol{P}\boldsymbol{A}(\boldsymbol{C})\boldsymbol{P}^\top\Big), \quad \forall \boldsymbol{P} \in S_n; \tag{5}$$

$$\text{Transformation Symmetry:} \quad \boldsymbol{Z}_{\text{inv}} \oplus_{(2)} (\boldsymbol{Z}_{\text{eqv}}\boldsymbol{R} + \boldsymbol{t}) = f\Big(\boldsymbol{X}, \boldsymbol{C}\boldsymbol{R} + \boldsymbol{t}, \boldsymbol{A}(\boldsymbol{C}\boldsymbol{R} + \boldsymbol{t})\Big), \quad \forall (\boldsymbol{R}, \boldsymbol{t}) \in SE(2), \tag{6}$$

where S_n is the group of permutation of n elements, and $SE(2)$ is the Special Euclidean group in 2D.

We leverage the E(n) equivariant graph neural network (EGNN) (Satorras et al., 2021) as the base GeoGNN encoder and decoder architecture for its effectiveness and efficiency. Considering the input and output of the lth EGNN layer (out of L, $l \in \{1, ..., L\}$) is denoted as $\boldsymbol{Z}_{\text{inv}}^{(l)} \oplus_{(2)} \boldsymbol{Z}_{\text{eqv}}^{(l)} = f^{(l)}(\boldsymbol{Z}_{\text{inv}}^{(l-1)}, \boldsymbol{Z}_{\text{eqv}}^{(l-1)}, \boldsymbol{A}(\boldsymbol{C}))$ where $\boldsymbol{Z}_{\text{inv}}^{(0)} = \boldsymbol{X}, \boldsymbol{Z}_{\text{eqv}}^{(0)} = \boldsymbol{C}, \boldsymbol{Z}_{\text{inv}} = \boldsymbol{Z}_{\text{inv}}^{(L)}, \boldsymbol{Z}_{\text{eqv}} = \boldsymbol{Z}_{\text{eqv}}^{(L)}$, the layer-wise message passing of the ith node is formulated as:

$$\text{Message:} \quad \text{Msg}_{i,j}^{(l)} = \text{MLP}_1\Big(\boldsymbol{Z}_{\text{inv}}^{(l-1)}[i] \oplus_{(2)} \boldsymbol{Z}_{\text{inv}}^{(l-1)}[j] \oplus_{(2)} \Big\|\boldsymbol{Z}_{\text{eqv}}^{(l-1)}[i] - \boldsymbol{Z}_{\text{eqv}}^{(l-1)}[j]\Big\|\Big); \tag{7}$$

$$\text{EqvEmb:} \quad \boldsymbol{Z}_{\text{eqv}}^{(l)}[i] = \boldsymbol{Z}_{\text{eqv}}^{(l-1)}[i] + \frac{1}{\text{Sum}(\boldsymbol{A}(\boldsymbol{C})[i])} \sum_{\substack{j \in \{1,\ldots,N\}, \\ \boldsymbol{A}(\boldsymbol{C})[i,j]=1}} (\boldsymbol{Z}_{\text{eqv}}^{(l-1)}[i] - \boldsymbol{Z}_{\text{eqv}}^{(l-1)}[j]) \times \text{MLP}_2(\text{Msg}_{i,j}^{(l)}); \tag{8}$$

$$\text{InvEmb:} \quad \boldsymbol{Z}_{\text{inv}}^{(l)}[i] = \text{MLP}_3\Big(\frac{1}{\text{Sum}(\boldsymbol{A}(\boldsymbol{C})[i])} \sum_{\substack{j \in \{1,\ldots,N\}, \\ \boldsymbol{A}(\boldsymbol{C})[i,j]=1}} \text{Msg}_{i,j}^{(l)}\Big), \tag{9}$$

where $\text{MLP}(\cdot)$ denotes the multilayer perceptron, $\text{Sum}(\cdot)$ represents the summation function, and the computation can be parallelized across all nodes simultaneously. We rely on the final invariant representation $\boldsymbol{Z}_{\text{inv}}$ for encoder embeddings and decoder reconstructions, so the final layer I/O is formulated as $\boldsymbol{Z}_{\text{inv}}^{(L)} = f^{(L)}(\boldsymbol{Z}_{\text{inv}}^{(L-1)}, \boldsymbol{Z}_{\text{eqv}}^{(L-1)}, \boldsymbol{A}(\boldsymbol{C}))$.

One issue with the current GeoGNN architectures is the presence of void node biases, where the addition of dummy void nodes unnecessarily affects the representations of real nodes. In cellular systems, the addition of non-realistic void cells (e.g. without any gene being expressed) introduces

bias into GeoGNN models, affecting cell representation in arbitrary directions. Specifically, we aim to further enforce the GeoGNN model to respect the void symmetry by strictly satisfying:

$$\text{Void Symmetry:} \quad \boldsymbol{Z}_{\text{inv}} \oplus_{(2)} \boldsymbol{Z}_{\text{eqv}} = f\Big(\boldsymbol{X} \oplus_{(1)} \tilde{\boldsymbol{X}}, \boldsymbol{C} \oplus_{(1)} \tilde{\boldsymbol{C}}, \boldsymbol{A}(\boldsymbol{C} \oplus_{(1)} \tilde{\boldsymbol{C}})\Big)[: N], \tag{10}$$

where $\tilde{\boldsymbol{X}} \in \mathbb{R}^{\tilde{N} \times M}$ represents the problem-specific void node features, $\tilde{N}$ is the number of void nodes, and $\tilde{\boldsymbol{C}} \in \mathbb{R}^{\tilde{N} \times 2}$ denotes the arbitrary locations of the void nodes. We focus on eliminating the bias for void cells, specifically when $\tilde{\boldsymbol{X}} = \mathbf{0}$. We achieve this by proposing a modified void-invariant architecture based upon EGNNs (Eqs. equation 7 - equation 9), with message-passing formulated as:

$$\text{Intensity:} \quad \text{Int}_{i,j}^{(l)} = \text{BFMLP}_1(\boldsymbol{Z}_{\text{inv}}^{(l-1)}[i]^\top \boldsymbol{Z}_{\text{inv}}^{(l-1)}[j]); \tag{11}$$

$$\text{Message:} \quad \text{Msg}_{i,j}^{(l)} = \text{BFMLP}_2\Big((\boldsymbol{Z}_{\text{inv}}^{(l-1)}[i] \oplus_{(2)} \boldsymbol{Z}_{\text{inv}}^{(l-1)}[j] \oplus_{(2)} \Big\|\boldsymbol{Z}_{\text{eqv}}^{(l-1)}[i] - \boldsymbol{Z}_{\text{eqv}}^{(l-1)}[j]\Big\|) \times \text{Int}_{i,j}^{(l)}\Big); \tag{12}$$

$$\text{EqvEmb:} \quad \boldsymbol{Z}_{\text{eqv}}^{(l)}[i] = \boldsymbol{Z}_{\text{eqv}}^{(l-1)}[i] + \sum_{\substack{j \in \{1,\ldots,N\}, \\ \boldsymbol{A}(\boldsymbol{C})[i,j]=1}} (\boldsymbol{Z}_{\text{eqv}}^{(l-1)}[i] - \boldsymbol{Z}_{\text{eqv}}^{(l-1)}[j]) \times \text{BFMLP}_3(\text{Msg}_{i,j}^{(l)}); \tag{13}$$

$$\text{InvEmb:} \quad \boldsymbol{Z}_{\text{inv}}^{(l)}[i] = \text{BFMLP}_4(\sum_{\substack{j \in \{1,\ldots,N\}, \\ \boldsymbol{A}(\boldsymbol{C})[i,j]=1}} \text{Msg}_{i,j}^{(l)}), \tag{14}$$

where $\text{BFMLP}(\cdot)$ denotes the bias-free multilayer perceptron containing no bias weights.

The key components introduced to guarantee void symmetry (Eq. equation 10) are as follows: 1) the introduction of the intensity term (Eq. equation 11) ensuring that the zero intensity if either the ith or jth node is void, thereby nullifying the void message; 2) the employment of bias-free multilayer perceptron to preserve the void message; 3) the utilization of sum pooling (rather than mean pooling) to eliminate the population bias in representation resulting from void nodes. More importantly, the void-invariant architecture allows the computation of derivatives at void features, i.e. $\frac{\partial(\boldsymbol{Z}_{\text{inv}} \oplus_{(2)} \boldsymbol{Z}_{\text{eqv}})}{\partial \tilde{\boldsymbol{X}}}|_{\partial \tilde{\boldsymbol{X}}=\mathbf{0}}$ exists, which is highly valuable for numerous downstream applications requiring gradient information (e.g. counterfactual search (Wang et al., 2025b)).

B.3 Geometric Self-Supervised Learning

We train the GeoGNN in a self-supervised manner through a masking-reconstruction approach. In the encoding stage, we remove 5% of the nodes for masking. Specifically, given an input geometric graph $G = \{\boldsymbol{X}, \boldsymbol{C}, \boldsymbol{A}(\boldsymbol{C})\}$, and denoting the masked and unmasked indices as $\mathbb{I}$ and $\bar{\mathbb{I}}$, respectively, the masking process is performed by removing the nodes as $\boldsymbol{X}_{\text{unm}} \oplus_{(2)} \boldsymbol{C}_{\text{unm}} = \text{Rmv}(\boldsymbol{X}, \boldsymbol{C}, \mathbb{I})$ where $\boldsymbol{X}_{\text{unm}} = \boldsymbol{X}[\bar{\mathbb{I}}], \boldsymbol{C}_{\text{unm}} = \boldsymbol{C}[\bar{\mathbb{I}}]$. A GeoGNN encoder $f_{\text{enc};\theta}(\cdot)$ parametrized with θ is then applied to embed the geometric graph as $\boldsymbol{Z}_{\text{enc}} = f_{\text{enc};\theta}(\boldsymbol{X}_{\text{unm}}, \boldsymbol{C}_{\text{unm}}, \boldsymbol{A}(\boldsymbol{C}_{\text{unm}}))$. In the decoding stage, we pad the masked node in the latent space with a learnable padding embedding $\boldsymbol{e} \in \mathbb{R}^D$. Specifically, the padding process is performed as $\boldsymbol{Z}_{\text{pad}} = \text{Pad}(\boldsymbol{Z}_{\text{enc}}, \boldsymbol{e}, \mathbb{I})$ where $\boldsymbol{Z}_{\text{pad}}[i] = \boldsymbol{e}, \forall i \in \mathbb{I}$. A GeoGNN decoder $f_{\text{dec};\phi}(\cdot)$ parametrized with ϕ is then applied to reconstruct the masked node features as $\boldsymbol{X}_{\text{dec}} = f_{\text{dec};\phi}(\boldsymbol{Z}_{\text{pad}}, \boldsymbol{C}, \boldsymbol{A}(\boldsymbol{C}))$.

We optimize the model by minimizing the discrepancy between the masked and reconstructed gene expressions. We optimize on a balanced MSE loss on all the K samples with the optimization formulated as:

$$\min_{\theta,\phi,\boldsymbol{e}} \frac{1}{K} \sum_{i=1}^{K} \Big(\frac{1}{2\sum_{\substack{j \in \{1,\ldots,N\}, \\ k \in \{1,\ldots,D\}, \\ \boldsymbol{X}_i[j,k]>0}} 1} \sum_{\substack{j \in \{1,\ldots,N\}, \\ k \in \{1,\ldots,D\}, \\ \boldsymbol{X}_i[j,k]>0}} (\boldsymbol{X}_i[j,k] - \boldsymbol{X}_{\text{dec},i}[j,k])^2 + \frac{1}{2\sum_{\substack{j \in \{1,\ldots,N\}, \\ k \in \{1,\ldots,D\}, \\ \boldsymbol{X}_i[j,k]=0}} 1} \sum_{\substack{j \in \{1,\ldots,N\}, \\ k \in \{1,\ldots,D\}, \\ \boldsymbol{X}_i[j,k]=0}} (\boldsymbol{X}_i[j,k] - \boldsymbol{X}_{\text{dec},i}[j,k])^2 \Big), \tag{15}$$

where the subscript represents the sample index.

C Extended Results

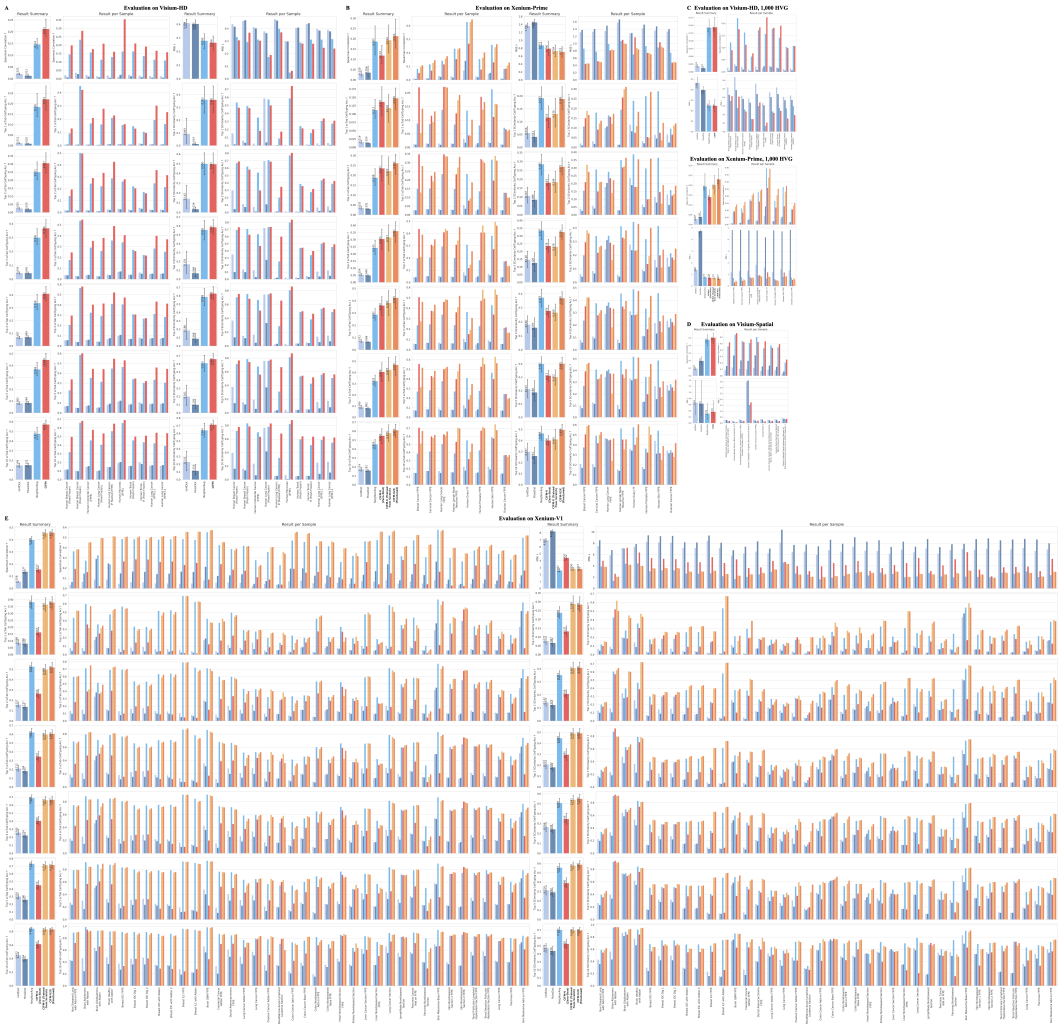

Figure 6: The full benchmarking evaluation results. (A) Evaluation on Visium-HD. (B) Evaluation on Xenium-Prime. (C) Evaluation of the 1,000 most differentially expressed genes in Visium-HD and Xenium-Prime. (D) Evaluation on Xenium-V1. (E) Evaluation on Visium-Spatial.