ST-JEPA: Joint-Embedding Predictive Architecture for Spatial Transcriptomics

Sebastian Birk [1 2 3] **Amirhossein Vahidi** [3] **Mohammad Vali Sanian** [3 4 5] **Arpit Merchant** [3] **Mohammad Lotfollahi** [3 6]

Abstract

Biological function emerges from the interplay of molecular programs within individual cells and the spatial organization of cells within tissues. Spatial transcriptomics captures both layers simultaneously, yet existing computational methods model them at a single scale: task-specific approaches focus on neighborhood-level clustering, while recent foundation models such as Nicheformer omit spatial context entirely during pretraining or, like Novae, lack hierarchical representations spanning multiple biological resolutions. We introduce ST-JEPA, the first joint-embedding predictive architecture for spatial transcriptomics. ST-JEPA operates across three biological scales through a multi-scale graph tokenization that converts spatial cell graphs into structured transformer sequences at gene, cell, and cellular neighborhood resolution, producing hierarchical embeddings suitable for diverse downstream tasks. Trained with a JEPA objective on mouse brain data spanning two technologies with non-overlapping gene panels, ST-JEPA achieves the best niche identification (weighted NMI=0.67) and the best batch integration (iLISI) among methods that perform well on niche identification, without explicit integration objectives. A technology metatoken enables cross-platform generalization even when gene panels do not overlap. Systematic ablations across six design axes provide practical guidance for self-supervised learning on spatial transcriptomics data.

[1]Institute of Computational Biology, Helmholtz Munich, Germany [2]School of Computation, Information and Technology, Technical University of Munich, Germany [3]Wellcome Sanger Institute, Wellcome Genome Campus, UK [4]Institute for Molecular Medicine Finland (FIMM), University of Helsinki, Helsinki, Finland [5]Department of Computer Science, University of Helsinki, Helsinki, Finland [6]Cambridge Stem Cell Institute, University of Cambridge, UK. Correspondence to: Sebastian Birk <sb75@sanger.ac.uk>, Mohammad Lotfollahi <ml19@sanger.ac.uk>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

1. Introduction

Biological function is inherently multi-scale: molecular programs within individual cells give rise to cell-level identity, cells organize into local neighborhoods that form functional niches, and these niches compose the tissue-level architecture underlying organ function. Single-cell spatial transcriptomics technologies such as MERFISH (Chen et al., 2015), STARmap PLUS (hereafter STARmap) (Shi et al., 2023), and Xenium (Janesick et al., 2023) now capture gene expression at subcellular resolution while preserving the spatial organization of cells within intact tissues. A central goal in spatial biology is to identify recurrent spatial patterns, or *niches* (also called *spatial domains*), from cellular neighborhoods. Foundation models that learn representations spanning these multiple biological scales could enable a broad range of downstream analyses from a single pretrained model.

A growing number of *task-specific* methods address individual spatial analysis problems. CellCharter (Varrone et al., 2024) combines a VAE with Gaussian mixture clustering; Banksy (Singhal et al., 2024) augments cell features with neighborhood statistics; GraphST (Long et al., 2023) uses contrastive graph learning. While effective for niche identification, these methods learn representations at a single scale and are tailored to one objective. Recent *foundation models* aim for more general representations but fall short of multi-scale modeling: Nicheformer (Schaar et al., 2025) (49.3M parameters, 110M cells) uses masked language modeling with a gene-rank tokenizer inspired by Geneformer (Theodoris et al., 2023), but does not incorporate spatial coordinates during pretraining, capturing spatial context only indirectly through cell type composition. Novae (Music et al., 2025) (30M cells) employs SwAV-style contrastive learning with optimal transport on a graph attention network; while it captures spatial context at the neighborhood level, it shows limited niche identification performance and lacks the hierarchical representations spanning gene, cell, and neighborhood resolution needed for diverse tasks in spatial biology.

We propose that the path toward multi-scale spatial foundation models requires three ingredients: (i) a tokenization that explicitly encodes biological structure at multiple resolutions, (ii) a self-supervised objective that learns to relate information across these scales, and (iii) a mechanism for cross-platform generalization. ST-JEPA realizes all three. Its *graph tokenization* converts spatial cell graphs into structured transformer sequences at three biological resolutions (gene, cell, and cellular neighborhood) by combining rank-based encoding (inspired by Geneformer (Theodoris et al., 2023)) with learned value embeddings for expression counts (inspired by scFoundation (Hao et al., 2024)). Spatial structure is captured through fixed sinusoidal segment embeddings on a k-nearest neighbor graph, producing a multi-segment sequence whose hierarchy mirrors that of the tissue.

Joint-Embedding Predictive Architectures (JEPA) (LeCun, 2022; Assran et al., 2023) predict *representations* of masked targets from visible context entirely in latent space, avoiding the pitfalls of reconstruction in data space and augmentation-dependent contrastive learning. I-JEPA (Assran et al., 2023) demonstrated this for images; JEPA-DNA (Charton et al., 2026) and GeneJEPA (Roohani et al., 2025) extend it to genomics and single-cell transcriptomics, but neither incorporates spatial structure. We introduce ST-JEPA (Figure 1), the first JEPA for spatial transcriptomics. Our contributions are: (1) a *multi-scale graph tokenization* that converts spatial cell graphs into hierarchical transformer sequences at gene, cell, and neighborhood resolution, producing embeddings at each biological scale from a single forward pass; (2) *block masking* of gene tokens across the neighborhood sequence, forcing prediction of masked representations from surrounding spatial context and thereby linking molecular and spatial information; (3) a *technology metatoken* enabling cross-platform generalization without explicit batch correction, even when gene panels do not overlap; and (4) *systematic ablations* across six design axes (tokenizer architecture, expression encoding, spatial encoding, sequence length, masking ratio, and normalization strategy) providing practical guidance for self-supervised learning on spatial data.

2. Method

Graph tokenization. Given N cells with expression vectors $\mathbf{x}_i \in \mathbb{R}^G$ and spatial coordinates $\mathbf{s}_i \in \mathbb{R}^2$, we construct a k-nearest neighbor graph ($k = 10$) based on Euclidean distance. For each cell i, we extract a *cellular neighborhood sequence*: the cell itself plus its k neighbors ordered by distance (Figure 1a,b). Genes are ranked by expression level and each assigned a unique token from a gene vocabulary of size V. The multi-segment sequence is:

$$\mathbf{T}_i = [\underbrace{\mathbf{t}_1^{(0)}, \ldots, \mathbf{t}_L^{(0)}}_{\text{index cell}}, \underbrace{\mathbf{t}_1^{(1)}, \ldots, \mathbf{t}_L^{(1)}}_{\text{neighbor 1}}, \ldots, \underbrace{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_L^{(k)}}_{\text{neighbor } k}] \tag{1}$$

where $L = 64$ per cell (704 total tokens). Each token carries four embeddings: gene identity, sinusoidal positional rank within the cell, a fixed sinusoidal segment embedding encoding distance-ranked cell membership (0 for index, 1–k for neighbors ordered by spatial distance), and a count value embedding via MLP projection.

Architecture. ST-JEPA has three components (Figure 1c). The **context encoder** f_θ is a transformer (dimension D, H heads, depth L_{enc}) processing visible tokens. The **target encoder** $f_{\bar{\theta}}$ shares the same architecture with EMA-updated parameters: $\bar{\theta} \leftarrow m\bar{\theta} + (1-m)\theta$, momentum m increasing from 0.995 to 1.0 during training. The **predictor** g_ϕ is a smaller transformer (depth L_{pred}) mapping context representations and learnable mask tokens to target embeddings.

Block masking and training. We partition the non-zero gene tokens across the entire neighborhood sequence into blocks and randomly mask a fraction ρ of tokens within each block (Figure 1c, steps 1–3). Masked tokens may belong to any cell in the neighborhood—index or neighbors—forcing the model to predict gene representations from surrounding spatial context. The training loss is:

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} \| g_\phi(f_\theta(\mathbf{T}_{\setminus\mathcal{M}}))_j - \text{sg}(f_{\bar{\theta}}(\mathbf{T}_{\mathcal{M}}))_j \|_1 \tag{2}$$

where $\mathcal{M}$ denotes masked positions and $\text{sg}(\cdot)$ is stop-gradient. We use AdamW with cosine LR decay (peak 6×10^{-5}), weight decay 0.04→0.4, and gradient clipping at norm 2.0.

3. Experiments

Setup. We train on mouse brain spatial transcriptomics data from MERFISH (Yao et al., 2023) and STARmap (Shi et al., 2023) (one batch per technology). Critically, ST-JEPA is trained on the *full, technology-specific gene panels* (MERFISH: 1,085 genes; STARmap: 1,015 genes; overlap: 431), rather than restricting to a shared gene subset. At inference time, only overlapping genes are used for cross-technology evaluation. Default configuration: embedding dimension 768, 8 attention heads, 5 encoder layers (3 for ablations), 1 predictor layer, masking ratio $\rho = 0.6$, trained for 800 epochs (batch size 256, bfloat16). We evaluate *niche identification* via Leiden clustering (NMI, ARI against anatomical annotations across resolutions 0.6–1.4; resolution fixed to yield 30 clusters for spatial maps) and *batch integration* via iLISI and MMD. All methods use the same k-NN spatial graph

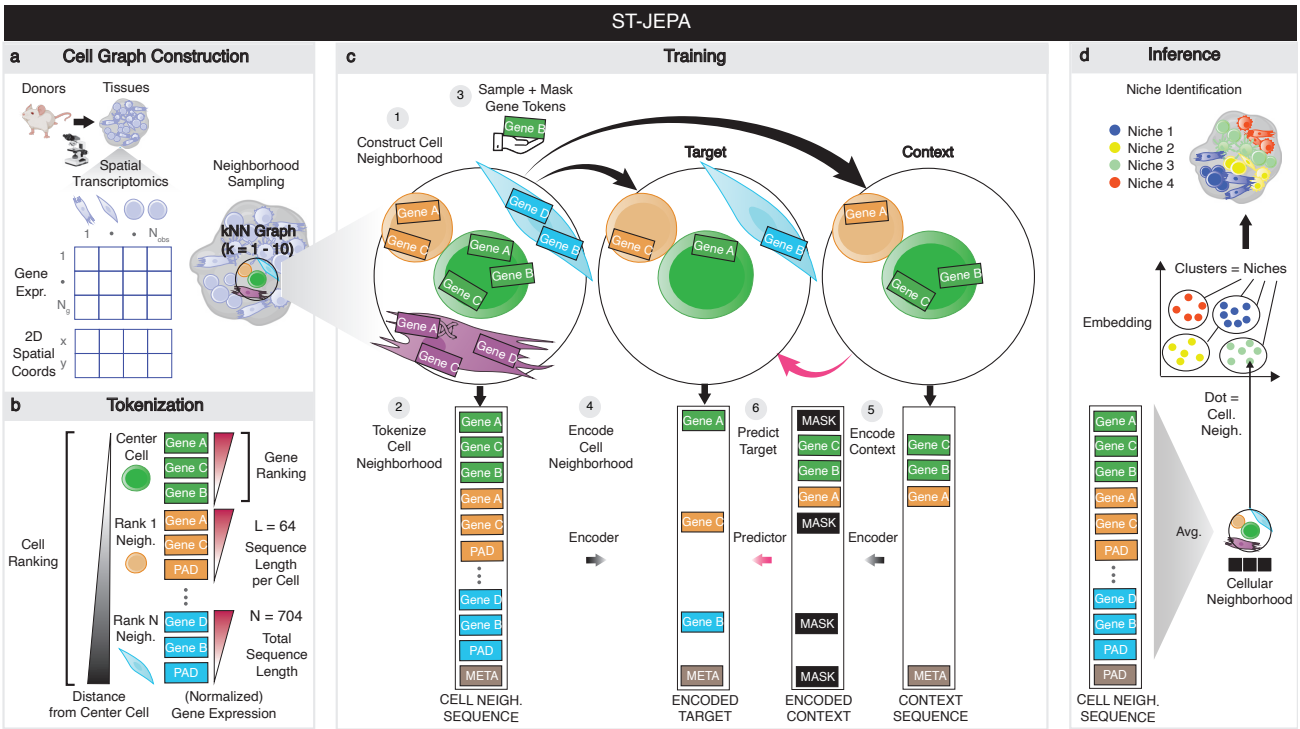

Figure 1. **Overview of ST-JEPA. (a)** Cell graph construction: spatial transcriptomics data is organized into a k-NN graph ($k = 10$) based on spatial coordinates. **(b)** Tokenization: each cell's genes are ranked by expression and encoded into $L = 64$ tokens per cell; $k+1 = 11$ cells form a multi-segment neighborhood sequence of $N = 704$ total tokens, with gene identity, expression value, positional rank, and segment embeddings. **(c)** Training: a target cell is selected and its gene tokens partially masked (steps 1–3); context and target encoders process visible and masked tokens respectively (steps 4–5), and a lightweight predictor maps context representations to target embeddings via ℓ_1 loss in latent space (steps 6–7). The target encoder is updated via EMA. **(d)** Inference: learned cellular neighborhood embeddings are averaged to produce per-cell representations, which are clustered to identify spatial niches.

Table 1. Benchmarking on mouse brain. Weighted NMI/ARI averaged over both technologies and across Leiden resolutions (mean±std; higher=better). iLISI: higher=better batch mixing. MMD: lower=better.

Method	NMI↑	ARI↑	iLISI↑	MMD↓
ST-JEPA v2	**.67±.00**	.34±.00	0.15	0.012
PCA baseline	.64±.01	.29±.03	0.00	0.069
CellCharter	.65±.00	**.35±.02**	0.02	0.027
Banksy	.63±.01	.31±.01	0.18	**0.002**
Novae	.63±.01	.31±.03	0.00	0.117
ST-JEPA v1	.58±.00	.25±.02	0.27	0.014
GraphST	.51±.02	.20±.03	**0.48**	0.041
Nicheformer	.19±.01	.05±.01	0.16	0.037

($k = 10$) as input. Baselines: CellCharter (Varrone et al., 2024), Banksy (Singhal et al., 2024), GraphST (Long et al., 2023), Novae (Music et al., 2025), Nicheformer (Schaar et al., 2025), and neighborhood expression PCA.

Niche identification (Table 1). ST-JEPA v2 achieves the best weighted NMI (0.67±0.00), outperforming all baselines including CellCharter (0.65±0.00), PCA (0.64±0.01), and Banksy (0.63±0.01), and is the most stable method across Leiden resolutions (NMI std=0.00). Importantly, ST-JEPA is a general-purpose self-supervised method *not de-*

Table 2. Ablation studies (mean±std over 3 seeds unless noted). Default in gray. †Single seed.

Axis	Variant	NMI	ARI	iLISI	MMD
Tok.	Gr.+comb.	.65±.02	.33±.05	.31	.020
	Gr.+rank	.62±.04	.32±.04	.26	.021
	Gr.+counts	.42±.10	.16±.01	.07	.027
	Nh.+comb.	.52±.04	.25±.03	.00	.038
	Nh.+counts	.60±.06	.31±.05	.01	.016
	Nh.+rank	.49±.02	.20±.02	.15	.017
Expr.	MLP	.65±.02	.33±.05	.31	.019
	Emb.(100)	.54±.05	.23±.03	.24	.034
	Emb.(50)	.61±.01	.29±.02	.18	.030
	Emb.(20)	.49±.05	.21±.03	.21	.029
Spat.	Rank	.65±.02	.33±.05	.31	.020
	Coord.	.40±.02	.16±.01	.85	.001
Len.	$L = 16$	.64±.01	.34±.01	.15	.012
	$L = 32$	.66±.03	.35±.01	.11	.013
	$L = 64$	.65±.02	.33±.05	.31	.020
	$L = 128$	.40±.11	.15±.06	.38	.035
	$L = 256$	.55±.04	.24±.03	.28	.029
Mask	$\rho = 0.3$	.43±.28	.21±.17	.24	.034
	$\rho = 0.4$	.47±.31	.23±.18	.30	.049
	$\rho = 0.5$	.28±.34	.13±.20	.39	.039
	$\rho = 0.6$	.65±.02	.33±.05	.31	.020
	$\rho = 0.7$	.57±.11	.28±.08	.18	.039
	$\rho = 0.8$	.59±.11	.31±.09	.12	.029
	$\rho = 0.9$	.63±.10	.32±.07	.02	.067
Norm†	None	.67	.36	.11	.021
	Shift. log	.65	.34	.25	.017
	Pearson	.25	.12	.22	.099
	Seurat	.09	.02	.69	.039

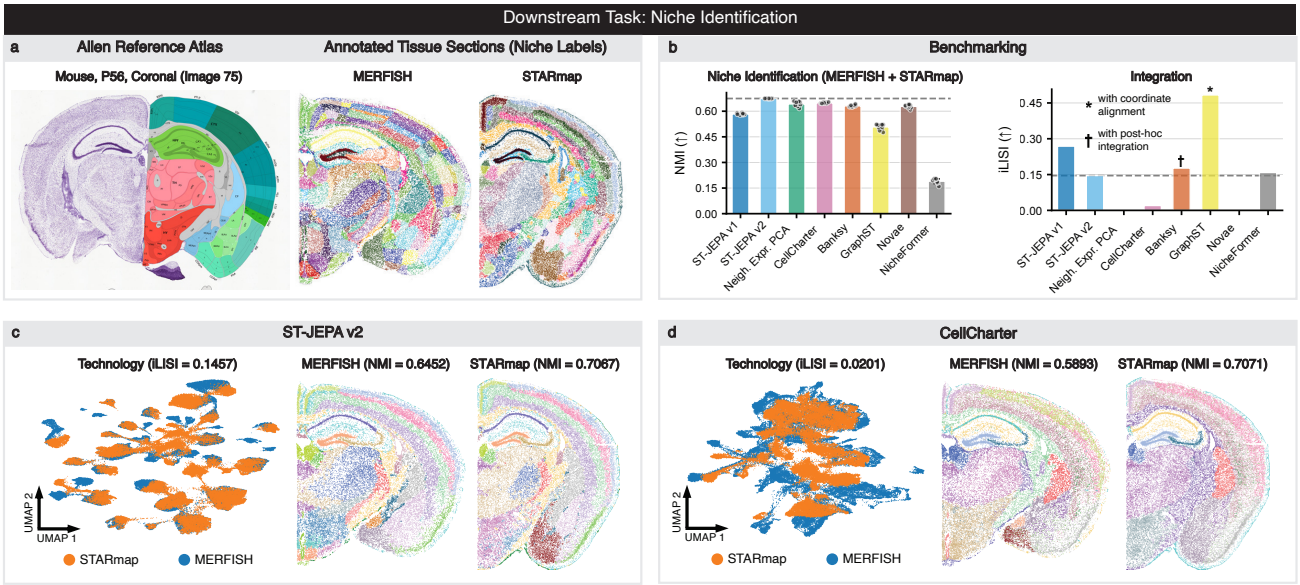

Figure 2. **Benchmarking results. (a)** Allen Reference Atlas and annotated tissue sections with niche labels for MERFISH and STARmap mouse brain. **(b)** Niche identification (NMI) and batch integration (iLISI) across all methods and technologies. **(c)** ST-JEPA v2 spatial embeddings: UMAP colored by technology (left) and niche cluster maps for MERFISH and STARmap (right), recovering anatomically coherent tissue domains. **(d)** CellCharter comparison maps (best competing method). Extended benchmarking visualizations for all baselines in Supplementary Figure S1.

signed for niche identification, yet outperforms task-specific methods that use explicit graph convolutions or GMM priors. Nicheformer—the other foundation model—performs poorly (NMI=0.19), indicating that spatial-aware objectives are essential. NMI and ARI are computed against annotations derived from the Allen Reference Atlas; qualitatively, ST-JEPA's cluster boundaries often align more closely with the reference anatomy than these annotations (Figure 2a,c), suggesting NMI scores underestimate the true quality of the representations.

Batch integration. Despite training on the *full, non-overlapping gene panels* of each technology, ST-JEPA v2 achieves competitive batch integration *without any explicit batch correction*. Among methods that also achieve strong niche identification (NMI>0.60), ST-JEPA v2 attains the best iLISI (0.15). Banksy (iLISI=0.18) and GraphST (iLISI=0.48) achieve higher integration scores, but rely on post-hoc embedding alignment (Harmony (Korsunsky et al., 2019)) and prior spatial coordinate registration (PASTE (Zeira et al., 2022)), respectively—assumptions that hold when integrating matched tissue sections but break down for distinct tissues or organs where spatial correspondence is unavailable. In contrast, ST-JEPA's integration emerges purely from the self-supervised objective: predicting in latent space naturally encourages batch-invariant features, as batch-specific artifacts are uninformative for predicting neighbors' representations. The metatoken—a separate learnable token appended to the cellular neighborhood sequence as an additional segment—provides a shared structural prior across technologies without requiring explicit alignment (Supplementary Figure S4).

ST-JEPA v1 vs. v2. ST-JEPA v1 follows LeJEPA (Garrido et al., 2024): a single encoder with stochastic views, invariance loss, and adversarial batch correction (Ganin et al., 2016). ST-JEPA v2 adopts I-JEPA: context/target encoders with EMA, a lightweight predictor, and block masking. Despite v1's explicit batch correction, v2 achieves substantially better niche identification (NMI=0.67 vs. 0.58) with competitive integration (MMD=0.012 vs. 0.014), suggesting that the JEPA predictive objective with the metatoken implicitly learns batch-invariant representations more effectively than adversarial training.

Ablation studies (Table 2). We ablate six design axes (all averaged over 3 seeds unless noted; Supplementary Figures S3–S7). *Tokenizer*: graph tokenization with combined encoding (rank + counts) performs best. *Expression encoding*: MLP projection outperforms scFoundation-style (Hao et al., 2024) value embeddings. *Spatial encoding*: rank-based dramatically outperforms coordinate-based. *Sequence length*: $L = 32$–64 perform best; $L \geq 128$ degrades as most cells express fewer genes (Supplementary Figure S5). *Masking ratio*: NMI peaks at $\rho = 0.6$; lower ratios show high variance. *Normalization*: raw counts achieve the best niche identification, indicating explicit normalization is unnecessary with combined tokenization (Supplementary Figure S7).

4. Conclusion

We presented ST-JEPA, the first JEPA for spatial transcriptomics. Our graph tokenization produces hierarchical embeddings at gene, cell, and neighborhood resolution, while a technology metatoken enables cross-platform generalization without explicit batch correction. ST-JEPA achieves the best niche identification and competitive integration. Ablations across six axes provide practical guidance for self-supervised learning on spatial data.

References

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., and Ballas, N. Self-supervised learning from images with a joint-embedding predictive architecture. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Charton, F. et al. JEPA-DNA: a foundation model for genomic sequences via joint-embedding predictive architecture. *arXiv preprint arXiv:2602.17162*, 2026.

Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S., and Zhuang, X. Spatially resolved, highly multiplexed RNA profiling in single cells. *Science*, 348(6233):aaa6090, 2015.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Garrido, Q., Assran, M., Ballas, N., Bardes, A., Najman, L., and LeCun, Y. Learning and leveraging world models in visual representation learning. *Advances in Neural Information Processing Systems*, 2024.

Hao, M., Gong, J., Zeng, X., Liu, C., Guo, Y., Cheng, X., Wang, T., Ma, J., Zhang, X., and Song, L. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, 21:1481–1491, 2024.

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

Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.-r., and Raychaudhuri, S. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods*, 16 (12):1289–1296, 2019.

LeCun, Y. A path towards autonomous machine intelligence. *Open Review*, 2022.

Long, Y., Ang, K. S., Li, M., Chong, K. L. K., Sethi, R., Zhong, C., Xu, H., Ong, Z., Sachaphibulkij, K., Chen, A., et al. Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with GraphST. *Nature Communications*, 14(1):1155, 2023.

Music, Q. B. et al. Novae: a graph-based foundation model for spatial transcriptomics data. *Nature Methods*, 2025.

Roohani, Y. et al. GeneJEPA: self-supervised learning of cellular representations from single-cell transcriptomics. *bioRxiv*, 2025.

Schaar, A. C. et al. Nicheformer: a foundation model for single-cell and spatial omics. *Nature Methods*, 2025.

Shi, H., He, Y., Zhou, Y., Huang, J., Maher, K., Wang, B., Tang, Z., Luo, S., Tan, P., Wu, M., et al. Spatial atlas of the mouse central nervous system at molecular resolution. *Nature*, 622(7983):552–561, 2023.

Singhal, V., Chou, N., Lee, J., Yue, Y., Liu, J., Chock, W. K., Lin, L., Chang, Y. C., Teo, E. M. S., Aow, J., et al. BANKSY unifies cell typing and tissue domain segmentation for scalable spatial omics data analysis. *Nature Genetics*, 56:431–441, 2024.

Theodoris, C. V., Xiao, L., Chopra, A., Chaffin, M. D., Al Sayed, Z. R., Hill, M. C., Manber, H., Lazzeroni, C., et al. Transfer learning enables predictions in network biology. *Nature*, 618:616–624, 2023.

Varrone, M., Tavernari, D., Santamaria-Martínez, A., Walsh, L. A., and Ciriello, G. CellCharter reveals spatial cell niches associated with tissue remodeling and cell plasticity. *Nature Genetics*, 56:74–84, 2024.

Yao, Z., van Velthoven, C. T., Kunst, M., Zhang, M., McMillen, D., Lee, C., Jung, W., Goldy, J., Abdelhak, A., et al. A molecularly defined and spatially resolved cell atlas of the whole mouse brain. *Nature*, 624(7991): 343–354, 2023.

Zeira, R., Land, M., Strzalkowski, A., and Raphael, B. J. Alignment and integration of spatial transcriptomics data. *Nature Methods*, 19(5):567–575, 2022.

Supplementary Material

A. Architecture Details

The context and target encoders share the same architecture: a standard transformer with pre-layer normalization, multi-head self-attention (8 heads), and feed-forward networks with GELU activation and expansion ratio 4. The predictor is a smaller transformer (1 layer, same embedding dimension) with learnable mask tokens. All embeddings are 768-dimensional. Gene tokens are embedded via a learned table of size V, positional encodings are fixed sinusoidal, and segment embeddings are also fixed sinusoidal (encoding distance-ranked neighbor ordering). Expression counts are projected through a 2-layer MLP ($1 \rightarrow 384 \rightarrow 768$) with GELU activation. Total: 23.7M parameters. Supplementary Figure S2 provides detailed diagrams of the tokenization pipeline and all encoding alternatives explored in ablations.

B. Extended Benchmarking

Supplementary Figure S1 shows UMAP embeddings and spatial tissue maps for all baseline methods not shown in the main text, including PCA, BANKSY+Harmony, GraphST+PASTE, Novae (zero-shot), and Nicheformer (zero-shot).

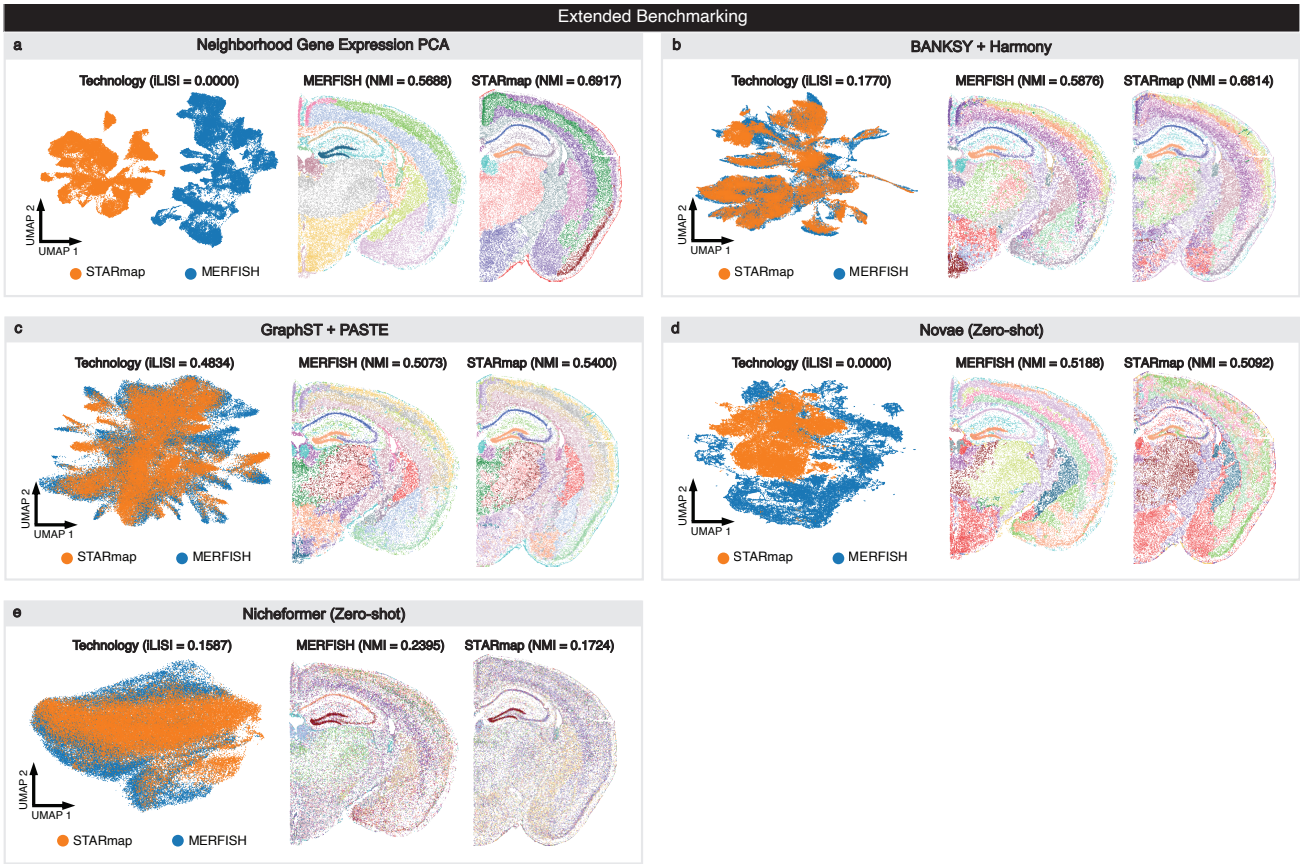

Figure S1. **Extended benchmarking.** UMAP embeddings colored by technology and niche cluster spatial maps for each baseline method. **(a)** Neighborhood gene expression PCA (iLISI=0.00). **(b)** BANKSY + Harmony (iLISI=0.18). **(c)** GraphST + PASTE (iLISI=0.48). **(d)** Novae zero-shot (iLISI=0.00). **(e)** Nicheformer zero-shot (iLISI=0.16). NMI values per technology shown above each spatial map.

C. Tokenization Details

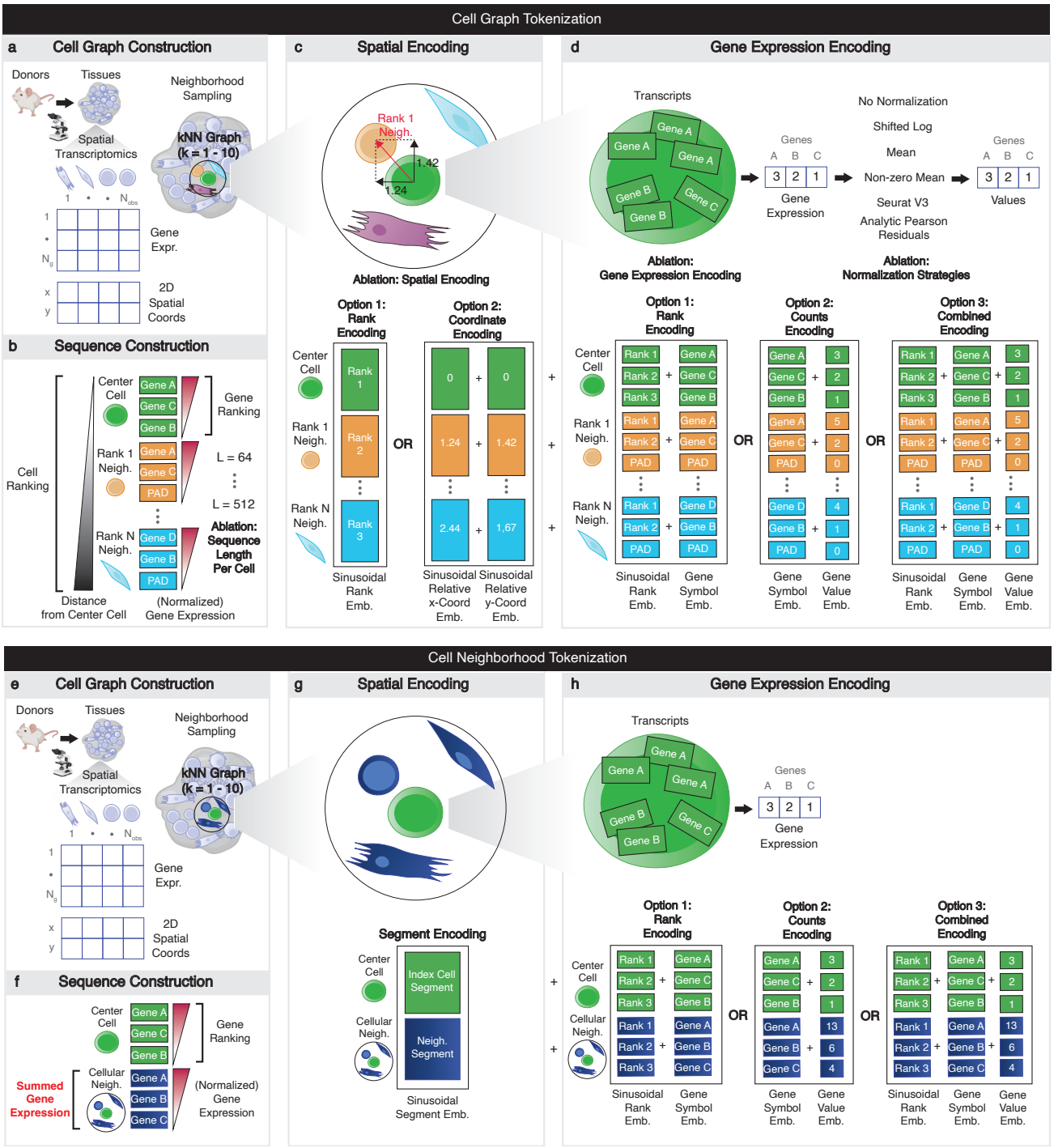

Figure S2. **Tokenization pipeline and design alternatives. Top:** Cell graph tokenization. **(a)** Cell graph construction from spatial transcriptomics data via k-NN. **(b)** Sequence construction with ranked gene tokens per cell, including sequence length ablation ($L = 64$ vs. $L = 512$). **(c)** Spatial encoding options: rank-based segment encoding vs. coordinate-based encoding. **(d)** Gene expression encoding options: rank encoding, counts encoding, and combined encoding, with normalization strategy ablation (analytic Pearson residuals, Seurat V3, shifted log, non-zero mean, no normalization). **Bottom:** Cell neighborhood tokenization variant. **(e)** Cell graph construction. **(f)** Sequence construction with summed neighborhood expression. **(g)** Segment encoding. **(h)** Gene expression encoding options for the neighborhood tokenizer.

We explore two tokenization strategies that differ in how spatial neighborhoods are represented (Supplementary Figure S2). The *cell graph tokenizer* (default) constructs a k-NN graph and represents each cellular neighborhood as a multi-segment sequence: individual gene tokens are preserved per cell, with segment embeddings distinguishing the index cell from each neighbor. This retains cell-level resolution within the neighborhood. The *cell neighborhood tokenizer* instead aggregates gene expression across the neighborhood before tokenization, producing a single-segment sequence from the summed expression profile. While more compact, this variant loses the ability to distinguish individual cell contributions. Within each strategy, we ablate three encoding schemes for gene expression: rank-only (ordinal position), counts-only (raw expression value via MLP), and combined (rank + counts). Two spatial encoding options are compared: rank-based segment embeddings that encode relative spatial distances to the index cell via sinusoidal functions, and coordinate-based encodings that encode absolute relative x/y coordinates with sinusoidal functions. The choice of normalization applied to expression counts before encoding is also explored (see Section H).

D. Ablation Studies

Supplementary Figure S3 presents the core ablation results across three design axes: tokenizer architecture, expression encoding, and spatial encoding. All experiments use 3 random seeds with the default configuration varied along one axis at a time. The cell graph tokenizer with combined encoding (rank + counts) achieves the best niche identification (NMI) and the strongest batch integration (iLISI) across both technologies. MLP-based continuous projection of gene counts substantially outperforms scFoundation-style value embeddings at all tested dimensions (20, 50, 100). The spatial encoding comparison reveals a striking result: rank-based segment embeddings dramatically outperform coordinate-based encodings on niche identification, while coordinate-based encoding achieves near-perfect batch integration (iLISI=0.85)—likely because absolute coordinates vary systematically across technologies, and the model learns to "mix" batches by projecting them into a shared coordinate space rather than learning biologically meaningful representations. Extended results including sequence length and masking ratio sweeps are in Supplementary Figures S6–S7.

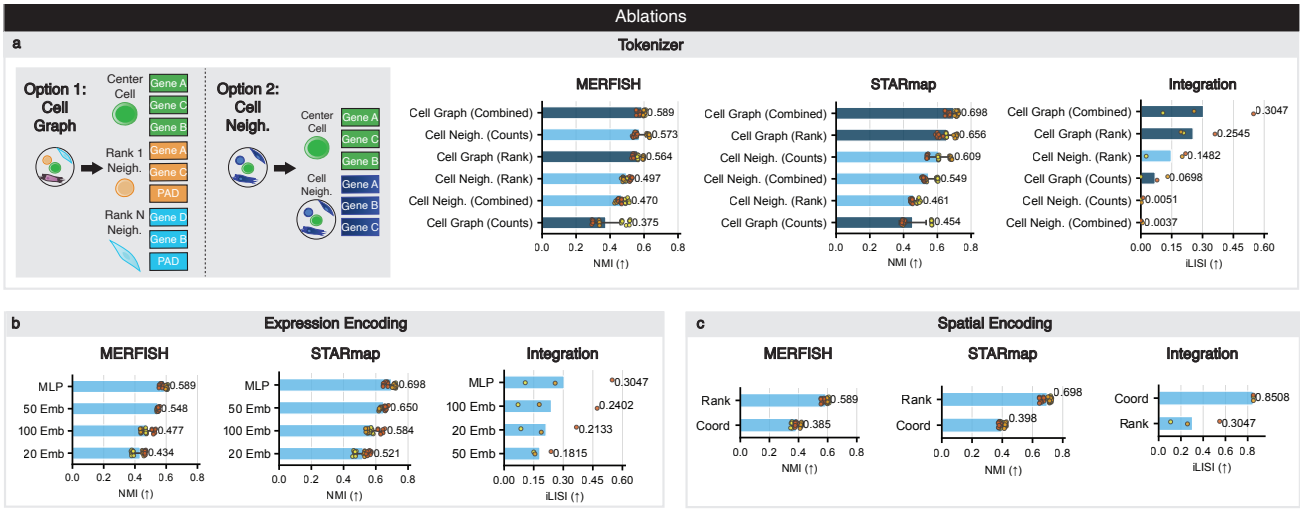

Figure S3. **Ablation studies. (a)** Tokenizer: cell graph tokenization (Option 1) outperforms cell neighborhood tokenization (Option 2) on NMI and iLISI across both MERFISH and STARmap. **(b)** Expression encoding: MLP-based continuous projection of gene counts outperforms scFoundation-style value embeddings (Emb.; 50, 100, 20 dimensions). **(c)** Spatial encoding: rank-based encoding dramatically outperforms coordinate-based encoding on both niche identification and batch integration. Full ablation results including sequence length and masking ratio sweeps in Supplementary Figures S6–S7.

E. Integration Ablation

The technology metatoken is a key architectural component for cross-technology integration. It is a separate learnable token that is appended to the cellular neighborhood sequence as an additional segment, distinct from the gene tokens of any individual cell. During training, a different metatoken is used for each technology (MERFISH vs. STARmap), providing the model with an explicit signal of data origin. Supplementary Figure S4 compares ST-JEPA with and without this mechanism. With the metatoken, the UMAP embedding space shows partial mixing of MERFISH and STARmap cells (iLISI=0.11)

while maintaining strong niche identification in both technologies (MERFISH NMI=0.60, STARmap NMI=0.71). Without it, batch integration collapses entirely (iLISI=0.00) and niche identification degrades substantially (MERFISH NMI=0.52, STARmap NMI=0.57). At inference, the metatoken is padded (zeroed out), so the model produces technology-agnostic embeddings. This demonstrates that the metatoken provides a shared structural prior that enables the self-supervised objective to learn batch-invariant features, without requiring any explicit alignment loss or post-hoc correction.

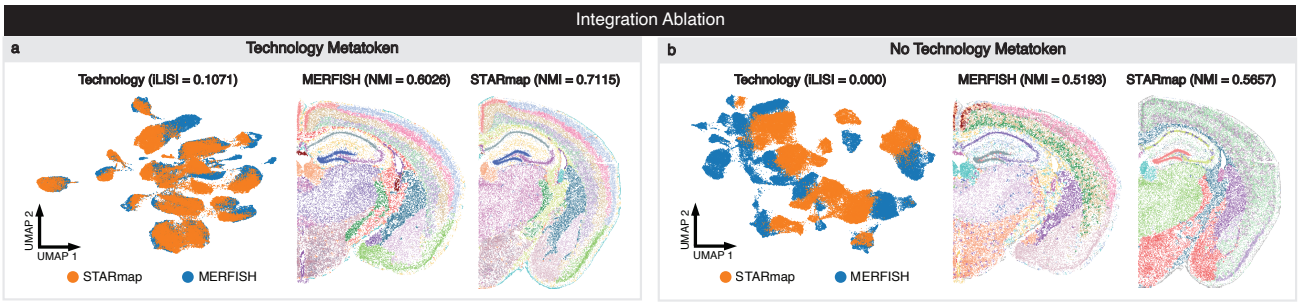

Figure S4. **Integration ablation: effect of the technology metatoken.** UMAP embeddings colored by technology and niche cluster spatial maps for MERFISH and STARmap. **(a)** With the technology metatoken (default): the embedding space shows partial mixing of technologies (iLISI=0.11) while maintaining strong niche identification (MERFISH NMI=0.60, STARmap NMI=0.71). **(b)** Without the technology metatoken: batch integration collapses (iLISI=0.00) and niche identification degrades (MERFISH NMI=0.52, STARmap NMI=0.57). The metatoken provides a shared structural prior that enables cross-technology integration without explicit alignment.

F. Data Statistics

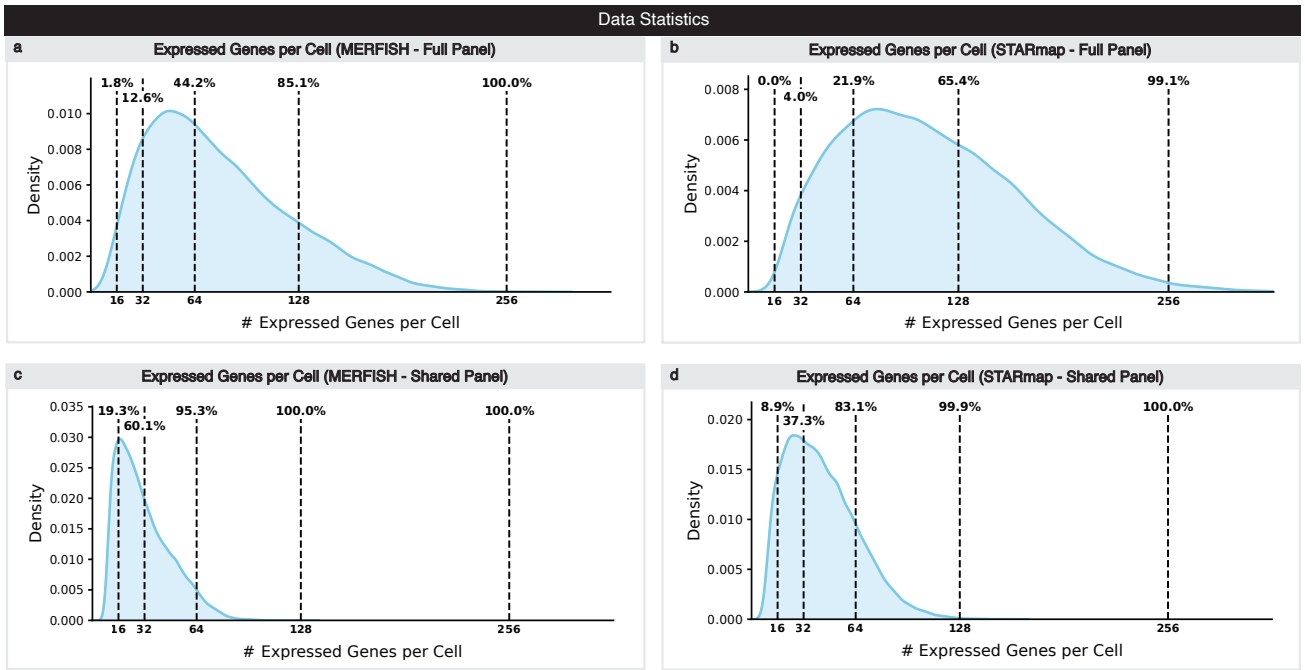

Figure S5. **Data statistics.** Distribution of expressed genes per cell for MERFISH and STARmap datasets using full and shared gene panels. Vertical dashed lines indicate cumulative percentages of cells with at least that many expressed genes at sequence length thresholds ($L = 16, 32, 64, 128, 256$). **(a)** MERFISH full panel: 44.2% of cells express $\geq$64 genes. **(b)** STARmap full panel: 21.9% at $L \geq 64$. **(c)** MERFISH shared panel: 95.3% at $L \geq 64$. **(d)** STARmap shared panel: 83.1% at $L \geq 64$. These distributions motivate the default sequence length $L = 64$.

The choice of sequence length L (number of gene tokens per cell) is constrained by the empirical distribution of expressed genes across technologies and gene panels. Supplementary Figure S5 shows these distributions for MERFISH and STARmap under both full (technology-specific) and shared (overlapping) gene panels. When using full panels, MERFISH cells express a median of approximately 60 genes and STARmap cells approximately 40, meaning that large sequence lengths ($L \geq 128$)

introduce substantial zero-padding. Specifically, only 44.2% of MERFISH cells and 21.9% of STARmap cells express ≥64 genes under full panels. However, evaluation uses only the 431 shared genes, where the distributions are denser: 95.3% of MERFISH cells and 83.1% of STARmap cells express ≥64 shared genes. The default $L = 64$ balances information content against padding overhead. Ablation results (Table 2, Supplementary Figure S6) confirm that $L = 32$–64 achieves the best niche identification; $L \geq 128$ degrades performance despite masked attention ignoring zero-padded positions, likely because cells with fewer expressed genes than L provide sparser training signal per sequence, while $L = 16$ slightly limits the transcriptomic information available per cell.

G. Extended Ablation Results

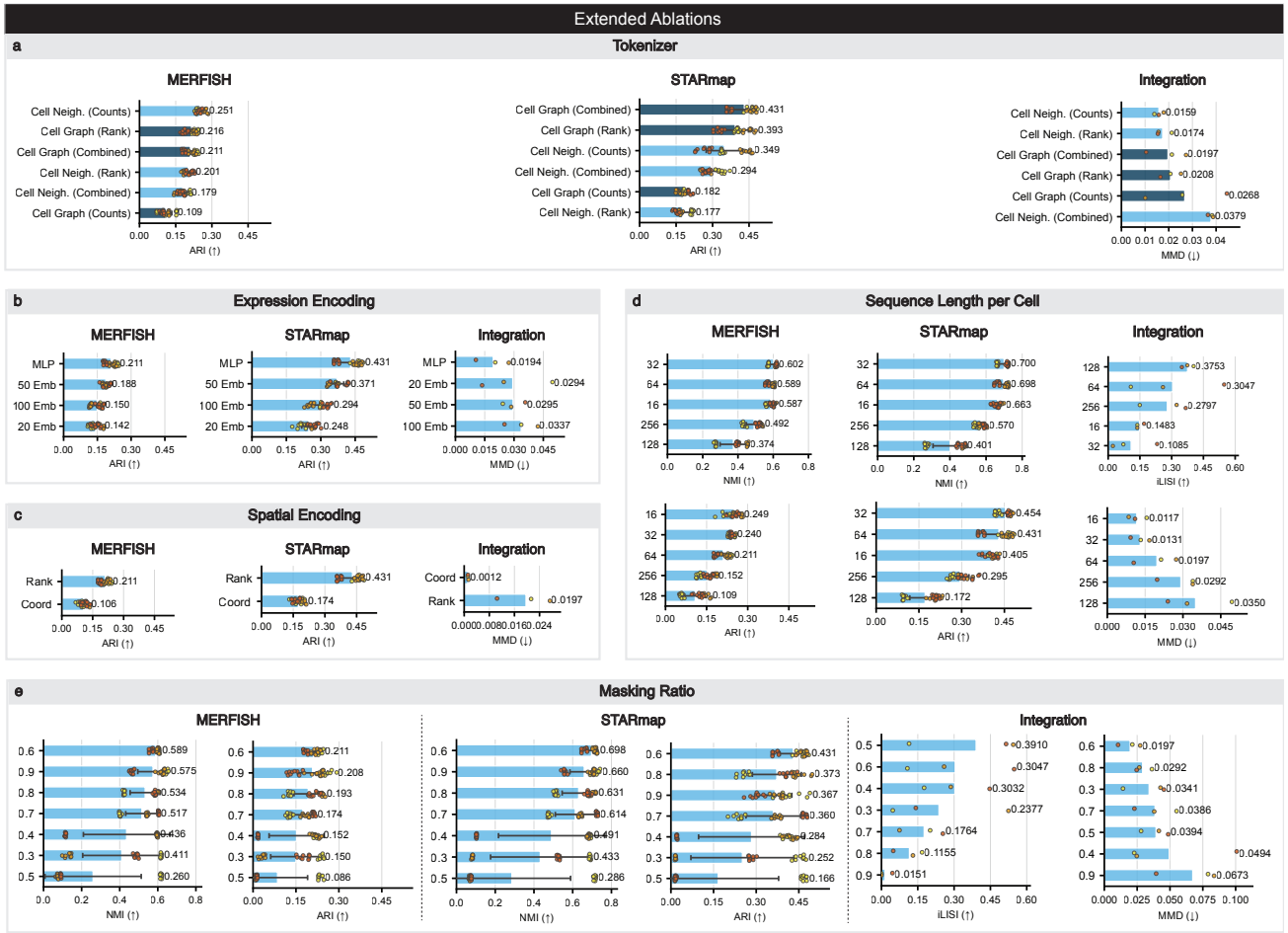

Figure S6. **Extended ablation results.** Full ablation sweeps with NMI, ARI, iLISI, and MMD metrics across MERFISH, STARmap, and cross-technology integration. **(a)** Tokenizer variants: cell graph vs. cell neighborhood with rank, counts, and combined encoding. **(b)** Expression encoding: MLP vs. value embeddings (20, 50, 100 dimensions). **(c)** Spatial encoding: rank-based vs. coordinate-based. **(d)** Sequence length per cell ($L = 16$–256): $L = 32$ achieves the best balance of niche identification and batch integration. **(e)** Masking ratio ($\rho = 0.3$–0.9): $\rho = 0.6$ is optimal for niche identification; batch integration improves monotonically with lower masking.

Supplementary Figure S6 presents the full ablation sweeps across all six design axes, with NMI, ARI, iLISI, and MMD metrics reported separately for MERFISH, STARmap, and cross-technology integration. For the *tokenizer* axis, cell graph tokenization consistently outperforms cell neighborhood tokenization across all encoding schemes; within each tokenizer, combined encoding (rank + counts) provides the best trade-off. For *expression encoding*, MLP projection outperforms value embeddings at all tested dimensions, with the gap widening as embedding dimension decreases (20-dim embeddings perform particularly poorly). The *spatial encoding* comparison confirms the main-text finding: coordinate-based encoding catastrophically fails on niche identification despite achieving high integration scores. The *sequence length* sweep reveals that $L = 32$ achieves the best overall balance of niche identification and batch integration, though $L = 64$ (the default) yields competitive NMI with better integration. For *masking ratio*, there is a clear optimum at $\rho = 0.6$; ratios below 0.5 exhibit

high variance across seeds, suggesting training instability when too few tokens are masked. Higher ratios ($\rho \geq 0.7$) are more stable but show degraded NMI, consistent with the model receiving too little visible context for accurate prediction.

H. Normalization Strategy Ablation

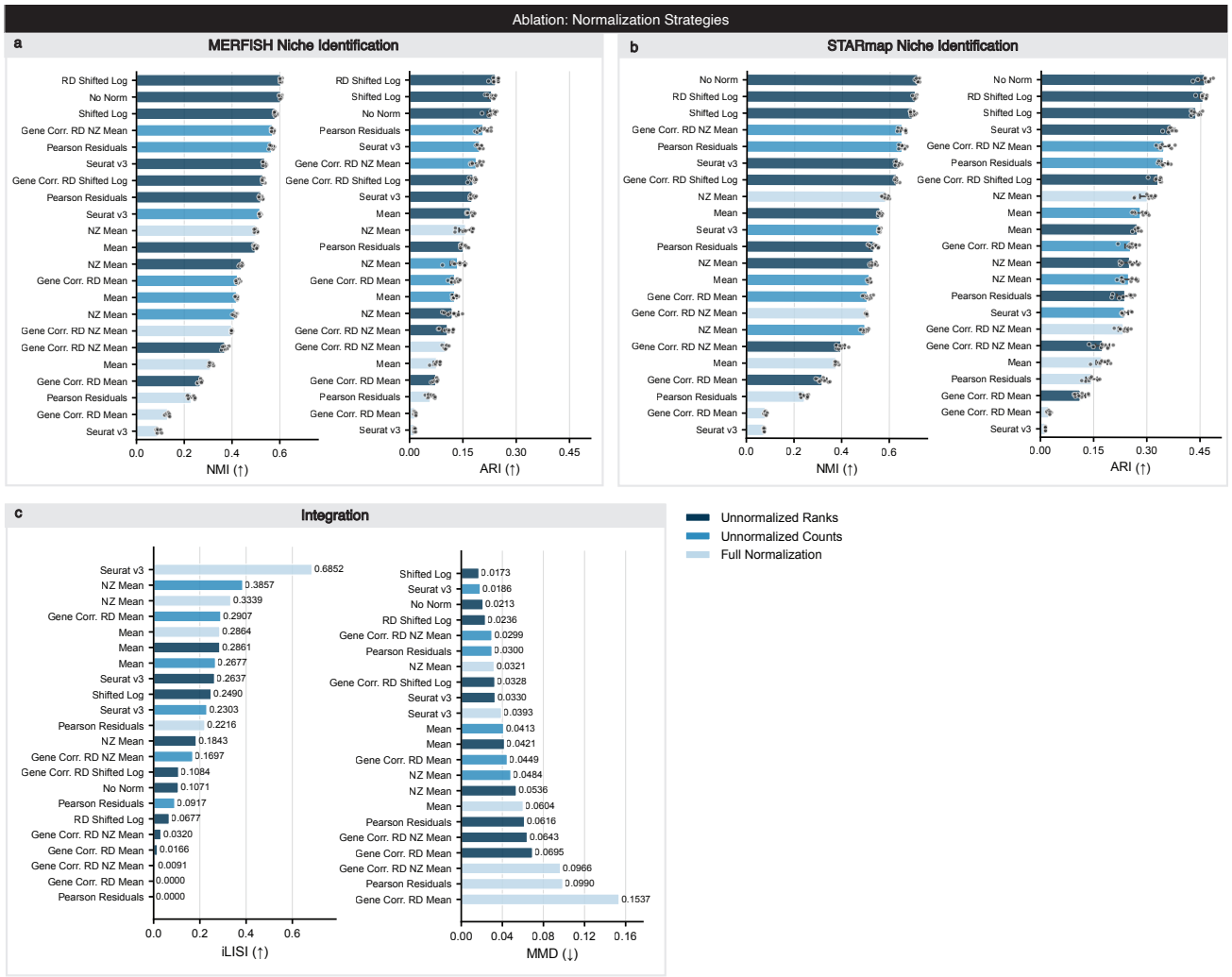

Figure S7. **Normalization strategy ablation.** Comprehensive comparison of normalization strategies for gene expression encoding, evaluated on niche identification (NMI, ARI) and batch integration (iLISI, MMD). All variants use the combined tokenizer (rank + counts encoding); normalizations can be applied independently to either the counts or the rank component. Strategies span three categories: unnormalized ranks (green), unnormalized counts (orange), and full normalization (blue). Results shown separately for **(a)** MERFISH niche identification, **(b)** STARmap niche identification, and **(c)** cross-technology integration. Raw counts with combined tokenization achieve the best niche identification, indicating that explicit normalization is unnecessary. Seurat V3 normalization achieves the best integration (iLISI=0.69).

Gene expression normalization is a critical preprocessing step in single-cell genomics, yet its interaction with learned tokenization schemes is poorly understood. We systematically evaluate 22 normalization strategies applied to the combined tokenizer (rank + counts encoding), where normalizations can be applied independently to either the counts component, the rank component, or both. Supplementary Figure S7 reports results across three categories: unnormalized ranks with various count normalizations (green), unnormalized counts with various rank normalizations (orange), and full normalization of both components (blue). The key finding is that *no normalization* (raw counts with raw ranks) achieves the best niche identification across both MERFISH and STARmap. This is likely because rank-based tokenization already provides implicit normalization—ranking genes by expression level is inherently robust to library size differences and technology-specific scaling factors. Applying additional normalization (particularly Seurat V3 or analytic Pearson residuals) distorts the rank

ordering and degrades niche identification, though Seurat V3 achieves the best batch integration (iLISI=0.69). Shifted log normalization provides a mild compromise, maintaining near-optimal NMI while modestly improving integration. These results suggest that for combined tokenization schemes, explicit normalization is unnecessary and can be counterproductive.