Graph Tokenization Meets JEPA: Self-Supervised Learning on Spatial Cell Graphs

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

Abstract

Applying transformers to graph-structured biological data requires converting irregular topologies into sequential token representations—a challenge that current graph foundation models address through various tokenization and positional encoding schemes. We introduce ST-JEPA, a joint-embedding predictive architecture that operates on spatial cell graphs derived from spatial transcriptomics data. Our key contribution is a *multi-scale graph tokenization* that converts local subgraph neighborhoods in a k-nearest neighbor graph into structured transformer sequences at three biological resolutions—neighborhood, cell, and gene—using fixed sinusoidal segment embeddings to encode graph topology. Unlike graph contrastive methods that require augmentation design or masked autoencoders that reconstruct in data space, ST-JEPA predicts representations of masked graph regions entirely in latent space via block masking across the subgraph sequence. Trained on mouse brain data spanning two technologies with non-overlapping feature sets, ST-JEPA achieves the best niche identification (weighted NMI=0.67) and the strongest cross-technology integration among high-performing methods, without explicit alignment objectives. Systematic ablations across six design axes—graph tokenization strategy, feature encoding, positional encoding, sequence length, masking ratio, and normalization—provide practical guidance for self-supervised learning on attributed spatial graphs.

[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>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

Graph foundation models (GFMs) aim to learn general-purpose representations from graph-structured data that transfer across tasks without task-specific supervision (Liu et al., 2024). A central challenge in building GFMs is *graph tokenization*: converting the irregular topology of graphs into a format suitable for transformer architectures (Vaswani et al., 2017), which expect sequential inputs. Existing approaches span a wide design space, from treating nodes as tokens with structural positional encodings (Ying et al., 2021; Dwivedi et al., 2022; Kreuzer et al., 2021) to various graph-to-sequence linearization strategies (Rampášek et al., 2022). Self-supervised pretraining on graphs has similarly explored diverse objectives, including contrastive learning (You et al., 2020; Veličković et al., 2019; Thakoor et al., 2022; Xia et al., 2022), masked feature reconstruction (Hou et al., 2022; Hu et al., 2020), and hybrid strategies. However, contrastive methods are sensitive to augmentation design on graphs, and masked autoencoders reconstruct in data space, which can encourage learning low-level statistics rather than semantic structure.

Spatial transcriptomics provides a compelling domain for graph foundation models. These technologies—MERFISH (Chen et al., 2015), STARmap (Shi et al., 2023), Xenium (Janesick et al., 2023)—measure gene expression in individual cells while preserving their spatial positions within intact tissues. After cell segmentation, the resulting data naturally forms a *spatial cell graph*: nodes are cells with high-dimensional gene expression features, and edges connect spatially proximate cells. A cell's identity depends not only on its molecular profile but critically on its *cellular neighborhood*—the composition and arrangement of surrounding cells that define tissue architecture. Identifying recurrent spatial patterns, or *niches*, from these graphs is a central goal in spatial biology.

Existing methods for spatial transcriptomics span task-

specific graph approaches and foundation models. CellCharter (Varrone et al., 2024) combines a VAE with Gaussian mixture clustering; Banksy (Singhal et al., 2024) augments node features with neighborhood statistics; GraphST (Long et al., 2023) applies contrastive graph learning. Among foundation models, Nicheformer (Schaar et al., 2025) uses masked language modeling but omits spatial structure during pretraining. Novae (Music et al., 2025) employs SwAV-style contrastive learning on a graph attention network, but remains sensitive to augmentation choices and shows limited niche identification. None of these methods leverage the joint-embedding predictive architecture (JEPA) paradigm for graphs.

We introduce ST-JEPA (Figure 1), a JEPA that operates on spatial cell graphs. Our approach makes four contributions to graph representation learning:

(1) Multi-scale graph tokenization. We convert local subgraph neighborhoods from a k-NN spatial graph into structured transformer sequences that preserve graph topology at three hierarchical resolutions: neighborhood (subgraph), cell (node), and gene (feature) (Figure 1a–e). Fixed sinusoidal segment embeddings encode distance-ranked graph structure, providing a learnable-free positional encoding of local topology.

(2) Graph-level JEPA with block masking. Rather than reconstructing node features (as in GraphMAE (Hou et al., 2022)) or contrasting augmented graph views (as in GraphCL (You et al., 2020)), we predict *representations* of masked graph regions from visible context entirely in latent space (Figure 1f,g). Block masking across the subgraph sequence forces the model to predict node representations from topologically distant context within the local neighborhood.

(3) Cross-graph integration via metatokens. A learnable metatoken per data source, appended as an additional segment to the subgraph sequence, enables integration across graphs with non-overlapping feature sets without explicit alignment—analogous to domain tokens in multi-domain graph learning (Figure 1g).

(4) Comprehensive design-space exploration. We systematically ablate six axes of the graph tokenization and training pipeline, providing practical guidance for self-supervised learning on attributed spatial graphs.

2. Related Work

Graph transformers and positional encodings. Adapting transformers to graphs requires encoding graph structure into the attention mechanism. Graphormer (Ying et al., 2021) uses centrality and spatial encodings as attention biases. SAN (Kreuzer et al., 2021) employs learned spectral positional encodings from the graph Laplacian. GPS (Rampášek et al., 2022) combines message-passing GNNs with transformer layers in a modular framework. GraphiT (Mialon et al., 2021) encodes structure via diffusion kernels. LSPE (Dwivedi et al., 2022) learns structural and positional representations jointly. These approaches typically treat each node as a single token, requiring $O(N^2)$ attention over the full graph or windowed approximations. Our approach instead tokenizes local k-hop subgraph neighborhoods into multi-token sequences, naturally limiting attention to topologically relevant context while preserving fine-grained node features.

Self-supervised learning on graphs. Graph self-supervised learning follows three main paradigms. *Contrastive methods*—DGI (Veličković et al., 2019), GraphCL (You et al., 2020), SimGRACE (Xia et al., 2022)—maximize agreement between augmented views, but performance depends heavily on augmentation design, which is domain-specific and hard to tune for attributed graphs. *Non-contrastive methods* like BGRL (Thakoor et al., 2022) avoid negative pairs but still rely on augmentations. *Generative methods*—node-level pretraining (Hu et al., 2020) and GraphMAE (Hou et al., 2022)—reconstruct masked node features or graph structure, but reconstruction in data space can encourage learning low-level statistics. Our approach adopts the JEPA paradigm (LeCun, 2022; Assran et al., 2023): predicting representations in latent space avoids both augmentation dependence (contrastive) and input-level reconstruction (generative), learning abstract semantic features instead.

JEPA beyond vision. I-JEPA (Assran et al., 2023) demonstrated JEPA for images by predicting representations of masked image patches from visible context. Extensions include JEPA-DNA (Charton et al., 2026) for genomic sequences and GeneJEPA (Roohani et al., 2025) for single-cell transcriptomics. GraphJEPA (Geriskenderi et al., 2024) applies the JEPA framework to graph-level classification via masked subgraph prediction. ST-JEPA differs by operating on *attributed* spatial graphs with high-dimensional node features, introducing a multi-scale tokenization that preserves gene-, cell-, and neighborhood-level resolution within each subgraph sequence.

Foundation models for spatial transcriptomics. Spatial transcriptomics data naturally forms cell graphs. Nicheformer (Schaar et al., 2025) (49.3M parameters, 110M cells) uses masked language modeling with rank-based tokenization inspired by Geneformer (Theodoris et al., 2023), but does not incorporate spatial graph structure during pretraining. Novae (Music et al., 2025) applies SwAV-style contrastive learning on a graph attention network (Veličković et al., 2018), capturing graph structure but remaining sensi-

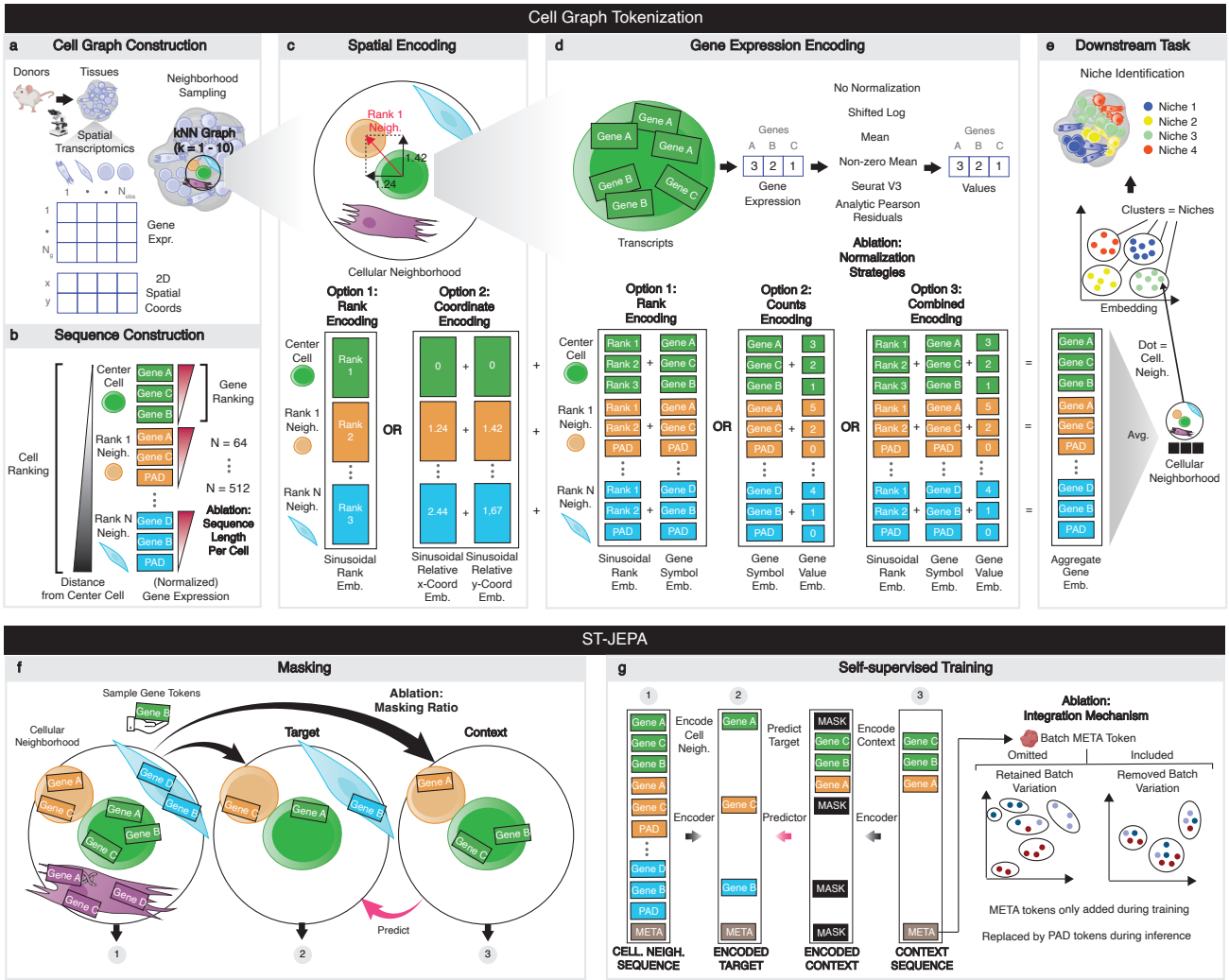

Figure 1. **Graph tokenization and ST-JEPA architecture. (a)** Spatial cell graph construction: data is organized into a k-NN graph ($k = 10$) based on spatial coordinates. **(b)** Multi-segment sequence construction: each node's genes are ranked by expression and truncated to the top L per node; the index node and its k neighbors form a subgraph sequence, with an ablation over sequence length per node. **(c)** Graph positional encoding options: rank-based segment encoding (default) vs. coordinate-based encoding. **(d)** Node feature encoding: normalization strategies and three encoding options (rank only, counts only, combined). **(e)** Downstream task: subgraph embeddings are averaged to produce per-node representations and clustered for niche identification. **(f)** Block masking: feature tokens are sampled from the subgraph and partitioned into target and context sets, with masking ratio ablation. **(g)** Self-supervised training: context encoder, target encoder (EMA-updated), and predictor operate in latent space; a technology metatoken is appended during training and replaced by a padding token at inference, with an integration mechanism ablation (included vs. omitted).

tive to augmentation choices. GraphST (Long et al., 2023) uses contrastive learning between graph-augmented and original features. Unlike these approaches, ST-JEPA combines explicit graph tokenization that preserves multi-scale topology with a JEPA objective that avoids both augmentation design and data-space reconstruction.

3. Method

3.1. Spatial Cell Graph Construction

Given N cells with gene 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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ based on Euclidean distance, with $k = 10$. Each node $i \in \mathcal{V}$ corresponds to a cell with feature vector $\mathbf{x}_i$, and directed edges $(i, j) \in \mathcal{E}$ connect each cell to its k nearest spatial neighbors. This graph captures the local tissue architecture: cells that are spatially proximate are more likely to interact through signaling, mechanical

forces, and shared microenvironment.

3.2. Multi-Scale Graph Tokenization

The core challenge is converting the irregular graph topology into structured sequences for a transformer. We extract a local subgraph for each node and tokenize it at three hierarchical resolutions (Figure 1a,b):

Subgraph extraction. For each node i, we extract its 1-hop ego subgraph: the node itself (the *index node*) plus its k neighbors, ordered by ascending edge weight (Euclidean distance). This produces a subgraph of $k+1=11$ nodes.

Node-level tokenization. Within each node, genes are ranked by expression level and the top L genes are retained. Each gene is assigned a token from a vocabulary of size V, producing L tokens per node. The complete subgraph sequence is:

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

with $L=64$ tokens per node, yielding $N=(k+1)\times L = 704$ total tokens per subgraph.

Token embeddings. Each token $\mathbf{t}_j^{(s)}$ carries four additive embeddings:

- **Gene identity embedding**: a learned lookup from a vocabulary table of size $V \times D$, where $D=768$ is the embedding dimension.
- **Positional rank embedding**: a fixed sinusoidal encoding of the gene's rank position $j \in \{1, \ldots, L\}$ within its node, analogous to positional encodings in standard transformers.
- **Segment embedding** (graph topology; Figure 1c): a fixed sinusoidal encoding of the segment index $s \in \{0, 1, \ldots, k\}$, where segment 0 is the index node and segments 1–k are neighbors ordered by spatial distance. This encodes the local graph topology—specifically, the distance-ranked edge structure around each node—without requiring learnable graph positional encodings.
- **Value embedding** (node features; Figure 1d): continuous encoding of gene expression counts via a 2-layer MLP ($1 \to 384 \to D$) with GELU activation. This preserves the continuous nature of expression values rather than discretizing them into bins.

The final token representation is: $\mathbf{e}_j^{(s)} = \mathbf{E}_{\text{gene}}[g_j] + \mathbf{E}_{\text{pos}}[j] + \mathbf{E}_{\text{seg}}[s] + \text{MLP}(x_j)$, where g_j is the gene index and x_j the expression value.

Alternative tokenization: neighborhood aggregation. We also explore a *neighborhood tokenizer* variant (Supplementary Figure S1) that first aggregates gene expression across all nodes in the subgraph (sum pooling), then tokenizes the aggregated profile as a single segment. This is more compact but loses the ability to distinguish individual node contributions.

3.3. Architecture

ST-JEPA consists of three components (Figure 1g):

Context encoder f_θ: a standard transformer (Vaswani et al., 2017) with pre-layer normalization, multi-head self-attention ($H=8$ heads), and feed-forward networks with GELU activation and expansion ratio 4. Depth $L_{\text{enc}}=5$ for benchmarking, $L_{\text{enc}}=3$ for ablation studies. Embedding dimension $D=768$. Flash attention and bfloat16 precision are used throughout.

Target encoder $f_{\bar{\theta}}$: shares the same architecture as the context encoder. Parameters are updated via exponential moving average (EMA): $\bar{\theta} \leftarrow m\bar{\theta} + (1-m)\theta$, with momentum m increasing from 0.995 to 1.0 during training following a cosine schedule. The EMA target encoder provides stable prediction targets, avoiding representation collapse without requiring negative pairs or explicit regularization.

Predictor g_ϕ: a lightweight transformer (1 layer, same embedding dimension) with learnable mask tokens. The predictor maps context encoder representations, concatenated with learnable mask tokens at target positions, to predicted target embeddings.

Total: 23.7M parameters (context encoder: 10.5M, target encoder: 10.5M shared architecture, predictor: 2.7M).

3.4. Block Masking on Subgraph Sequences

We design a masking strategy that operates across the graph-structured subgraph sequence (Figure 1f). Non-zero gene tokens across the entire sequence are partitioned into $V=4$ contiguous blocks, and a fraction ρ of tokens within each block is randomly masked. Importantly, masked tokens may belong to *any node* in the subgraph—index or neighbors—forcing the model to predict gene representations from topologically and featurally distant context. This differs from standard graph masking that operates at the node level: our block masking operates at the sub-node feature level across the subgraph, encouraging richer cross-node reasoning.

3.5. Training Objective

The training loss is the ℓ_1 distance between predicted and target representations:

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

where $\mathcal{M}$ denotes masked positions and $\text{sg}(\cdot)$ is stop-gradient on the target encoder outputs. Unlike masked autoencoders (He et al., 2022; Hou et al., 2022) that reconstruct input features, our objective predicts in the learned representation space, encouraging the model to capture abstract semantic structure rather than low-level feature statistics.

Cross-graph integration. When training on data from multiple graphs (technologies) with non-overlapping feature sets, we introduce a *metatoken*: a learnable token per data source appended to the subgraph sequence as an additional segment (Figure 1g). During training, each technology has a distinct metatoken, providing the model with a data-source signal. At inference, the metatoken is *padded (zeroed out)*, so the model produces source-agnostic embeddings. This mechanism is analogous to domain tokens in multi-domain pretraining and enables integration without explicit alignment losses. Each training batch contains data from a single technology.

Optimization. We use AdamW (Loshchilov & Hutter, 2019) with cosine learning rate decay (warmup: 1 epoch, peak: 6×10^{-5}, final: 2×10^{-5}), weight decay increasing from 0.04 to 0.4, gradient clipping at norm 2.0, batch size 256, bfloat16 precision. We train for 300 epochs for benchmarking (5 encoder layers) and evaluate ablations (3 encoder layers) at 300, 600, and 900 epochs.

4. Experiments

4.1. Setup

Data. We train on two mouse brain spatial transcriptomics datasets: MERFISH (Yao et al., 2023) (1,085 genes) and STARmap (Shi et al., 2023) (1,015 genes), with only 431 overlapping genes. Critically, ST-JEPA trains on the *full, technology-specific gene panels* rather than restricting to the shared subset. At inference, only overlapping genes are used for cross-technology evaluation. The k-NN spatial graphs ($k=10$) are constructed independently per tissue section.

Evaluation. We evaluate on two tasks: (1) *niche identification*—clustering the learned node embeddings via Leiden clustering and measuring agreement with anatomical annotations from the Allen Reference Atlas using weighted NMI and ARI (averaged over Leiden resolutions 0.6–1.4; resolution fixed to yield 30 clusters for spatial maps); and (2) *cross-graph integration*—measuring batch mixing via iLISI (higher = better mixing) and MMD (lower = better). All methods use the same k-NN spatial graph ($k=10$) as input.

Baselines. We compare against both task-specific graph methods and foundation models: CellCharter (Varrone et al., 2024) (VAE + GMM), Banksy (Singhal et al., 2024) (neighborhood feature augmentation + Harmony (Korsunsky et al., 2019)), GraphST (Long et al., 2023) (contrastive graph learning + PASTE (Zeira et al., 2022) alignment), Novae (Music et al., 2025) (graph attention + SwAV contrastive), Nicheformer (Schaar et al., 2025) (masked language model, no graph structure), and neighborhood expression PCA.

4.2. Benchmarking Results

Niche identification (Figure 2). ST-JEPA v2 achieves the best weighted NMI (0.67 ± 0.00), outperforming CellCharter (0.65), PCA (0.64), and Banksy (0.63), with the lowest variance across Leiden resolutions. Notably, ST-JEPA is a general-purpose self-supervised method *not designed for clustering*, yet outperforms task-specific methods using explicit graph convolutions or Gaussian mixture priors. Nicheformer—the other foundation model, applied zero-shot—performs poorly (NMI=0.19), suggesting that pretraining without graph structure is insufficient for spatial niche identification. Qualitatively, ST-JEPA's cluster boundaries often align more closely with the anatomical reference than the ground-truth annotations (Figure 2a,d), suggesting NMI may underestimate true quality.

Cross-graph integration. Despite training on *non-overlapping feature sets*, ST-JEPA v2 achieves competitive integration *without explicit alignment*. Among methods with strong niche identification (NMI>0.60), ST-JEPA v2 attains the best iLISI (0.15; Figure 2b). Banksy (iLISI=0.18) and GraphST (iLISI=0.48) achieve higher integration but rely on post-hoc alignment (Harmony) or spatial coordinate registration (PASTE). ST-JEPA's integration emerges purely from the self-supervised objective: predicting in latent space encourages source-invariant features.

ST-JEPA v1 vs. v2. Both variants use the JEPA framework but differ in training design. ST-JEPA v1 (adapted from LeJEPA; Garrido et al. 2024) uses distributional regularization and adversarial domain correction (Ganin et al., 2016); v2 replaces these with an EMA target encoder and a learnable metatoken. Despite v1's explicit domain 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 simpler EMA-based design with metatokens is more effective. The quali-

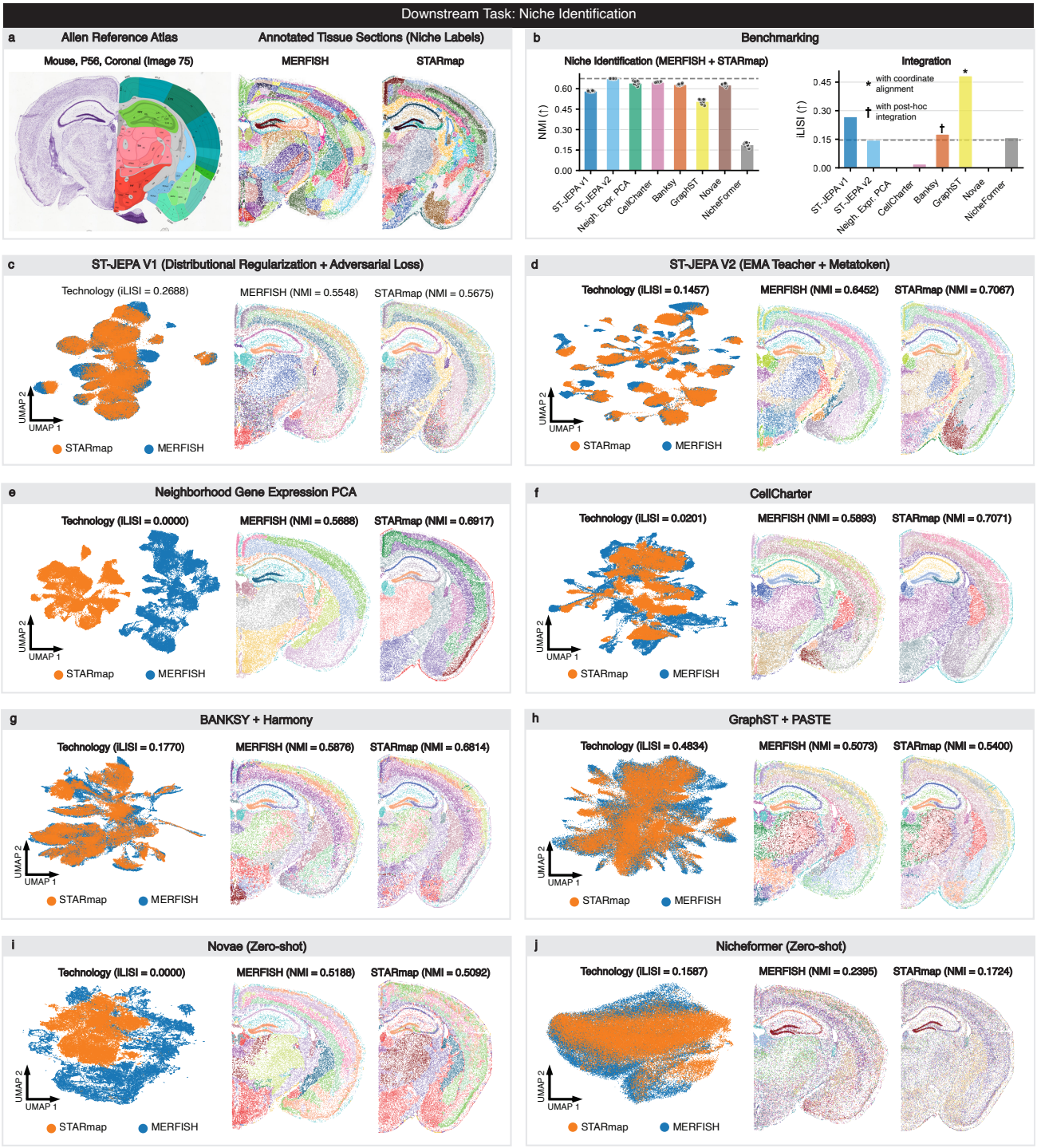

Figure 2. **Downstream task: niche identification.** **(a)** Allen Reference Atlas (Mouse P56, coronal Image 75) and annotated tissue sections with niche labels for MERFISH and STARmap. **(b)** Benchmarking: weighted NMI (niche identification) and iLISI (batch integration) across all methods; $\star$ indicates methods using coordinate alignment, † indicates post-hoc integration. **(c–j)** Per-method UMAP embeddings colored by data source (left) and niche cluster spatial maps (right), with per-technology iLISI and NMI: **(c)** ST-JEPA v1 (distributional regularization + adversarial loss), **(d)** ST-JEPA v2 (EMA teacher + metatoken), **(e)** neighborhood gene expression PCA, **(f)** CellCharter, **(g)** BANKSY + Harmony, **(h)** GraphST + PASTE, **(i)** Novae (zero-shot), **(j)** Nicheformer (zero-shot).

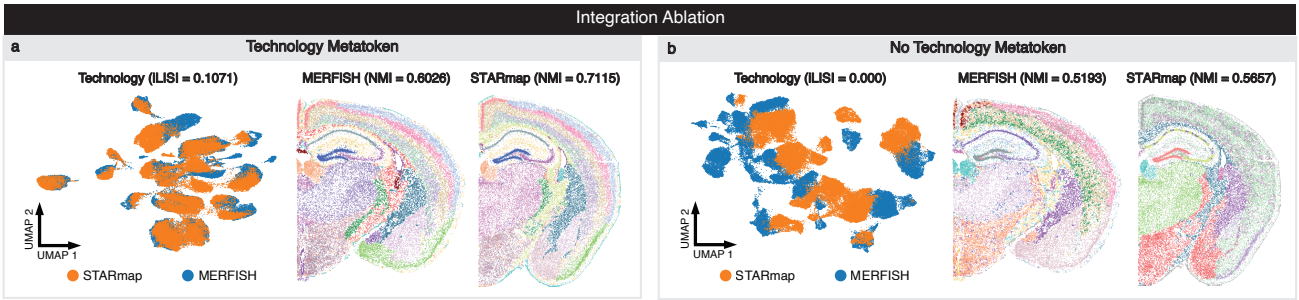

Figure 3. **Metatoken ablation.** UMAP embeddings colored by data source and niche cluster spatial maps. **(a)** With metatoken: partial cross-graph mixing (iLISI=0.107) with strong niche identification (MERFISH NMI=0.60, STARmap NMI=0.71). **(b)** Without metatoken: integration collapses (iLISI=0.000), niche identification degrades (MERFISH NMI=0.52, STARmap NMI=0.57).

tative difference is visible in Figure 2c,d.

4.3. Integration Mechanism: Metatoken Ablation

The metatoken is critical for cross-graph integration (Figure 3). Without it, integration collapses entirely (iLISI: $0.11 \rightarrow 0.00$) and niche identification degrades substantially (overall NMI: $0.66 \rightarrow 0.55$). The metatoken provides a shared structural prior that enables the JEPA objective to learn source-invariant features without explicit alignment. At inference, zeroing out the metatoken produces source-agnostic embeddings. This mechanism is analogous to domain tokens in multi-domain graph pretraining and demonstrates that simple architectural modifications can enable cross-graph transfer.

4.4. Ablation Studies

We systematically ablate six design axes of the graph tokenization and training pipeline (Figure 4). All experiments use 3 encoder layers with the default configuration varied along one axis, evaluated at 300, 600, and 900 training epochs.

(1) Graph tokenization strategy (Figure 4a). The *subgraph tokenizer* (default) preserves per-node structure, while the *aggregated tokenizer* sums expression across nodes before tokenizing. Preserving per-node structure is consistently better: NMI=0.65 vs. 0.52 with combined encoding (25% relative improvement). The aggregated tokenizer also shows zero cross-graph integration (iLISI=0.00), likely because losing node-level structure prevents learning generalizable graph features.

(2) Node feature encoding (Figure 4b). MLP projection (continuous, default) consistently outperforms discretized value embeddings, with the gap widening at lower resolutions (20-bin NMI=0.49 vs. MLP NMI=0.65), indicating that preserving continuous feature information is important.

(3) Graph positional encoding (Figure 4c). *Rank-based* segment encoding (default) dramatically outperforms *coordinate-based* encoding (NMI=0.65 vs. 0.40). Coordinate-based encoding achieves high iLISI (0.85), but this is artifactual: the model learns to align coordinate spaces rather than learning meaningful graph representations. Rank-based encoding provides a permutation-equivariant summary of local topology invariant to global coordinate transformations.

(4) Sequence length (Figure 4e). $L = 32$–64 achieves the best niche identification; $L \geq 128$ degrades because most nodes have fewer non-zero features (only 44.2% of MERFISH and 21.9% of STARmap nodes have $\geq$64 under full panels; Supplementary Figure S2).

(5) Masking ratio (Figure 4d). NMI peaks at $\rho = 0.6$; lower ratios ($\rho \leq 0.5$) exhibit high variance across training epochs (std up to 0.34), suggesting training instability when the prediction task is too easy. The optimal 60% masking ratio (applied to non-zero tokens) is lower than MAE's 75% (He et al., 2022).

(6) Feature normalization. Among 22 normalization strategies (Supplementary Figure S4), no normalization achieves the best niche identification (NMI=0.67), because rank-based tokenization already provides implicit normalization. Explicit normalization distorts rank ordering and degrades performance, though Seurat V3 achieves the best integration (iLISI=0.69).

5. Discussion

Our results provide several insights for graph foundation models. First, *graph tokenization preserving multi-scale structure is important*: Nicheformer (49.3M parameters, 110M cells) performs poorly on spatial niche identification, potentially because it does not incorporate graph structure during pretraining. Our hierarchical tokenization—subgraph, node, and feature levels—naturally produces em-

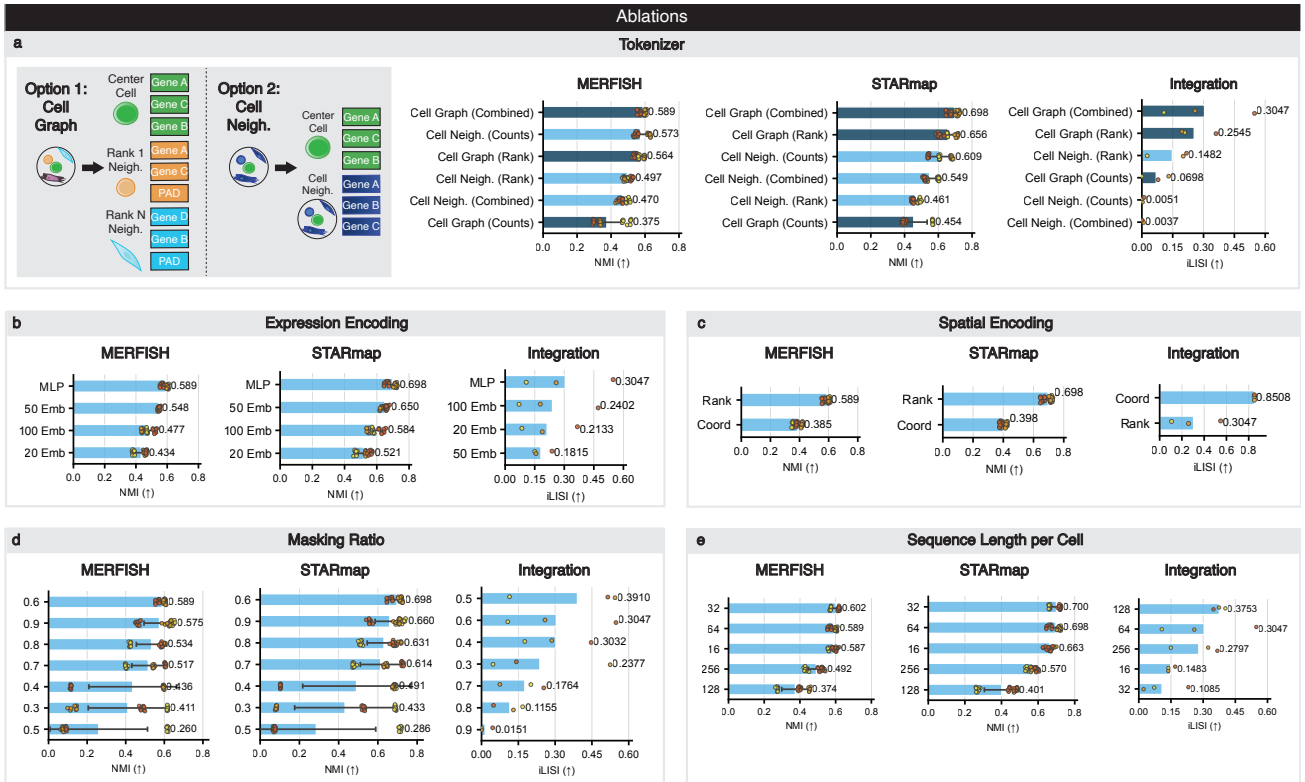

Figure 4. **Ablation studies.** Per-technology NMI (MERFISH, STARmap) and cross-graph integration (iLISI) across five design axes, each varied independently from the default configuration (evaluated at 300, 600, and 900 epochs). **(a)** Graph tokenization: subgraph tokenization (Option 1, preserving per-node structure) outperforms aggregated neighborhood tokenization (Option 2) across all encoding schemes; combined encoding (rank + counts) achieves the best trade-off. **(b)** Node feature encoding: MLP-based continuous projection outperforms discretized value embeddings at all tested dimensions. **(c)** Graph positional encoding: rank-based segment encoding dramatically outperforms coordinate-based encoding on niche identification; coordinate-based encoding achieves high iLISI by exploiting spatial biases rather than learning graph-structural representations. **(d)** Masking ratio: $\rho = 0.6$ is optimal; lower ratios show high variance across training epochs. **(e)** Sequence length per node: $L = 32$–64 achieve the best niche identification; $L \geq 128$ degrades due to zero-padding.

beddings at multiple granularities that capture spatial context. Second, comparing ST-JEPA v1 and v2 reveals that *training design choices matter*: v1 uses distributional regularization and adversarial domain correction (Ganin et al., 2016) (adapted from LeJEPA; Garrido et al. 2024), while v2 uses an EMA target encoder with a metatoken; v2's simpler design achieves substantially better niche identification (NMI=0.67 vs. 0.58). Third, *graph positional encoding design is critical*: rank-based segment embeddings encoding relative topology outperform coordinate-based encodings that leak global position, suggesting GFMs should encode *relative* graph structure rather than *absolute* node positions.

Future directions and limitations. Our multi-scale tokenization naturally produces gene-, cell-, and neighborhood-level representations; while we evaluate niche identification (neighborhood level), future work can explore gene- and cell-level downstream tasks. Training on a larger corpus spanning diverse tissues and diseases would test generalization potential. Current limitations include evaluation on a single domain (mouse brain spatial graphs), the assumption of a meaningful spatial metric for k-NN construction, and a fixed subgraph size ($k = 10$).

6. Conclusion

We introduced ST-JEPA, a JEPA for spatial cell graphs combining multi-scale graph tokenization with latent-space prediction. ST-JEPA achieves the best niche identification and competitive cross-graph integration without explicit alignment, outperforming both task-specific methods and larger foundation models. Ablations across six design axes provide practical guidance for self-supervised learning on attributed spatial graphs, suggesting that graph tokenization and JEPA objectives are promising directions for graph foundation models.

Impact Statement

This work advances self-supervised learning on graph-structured biological data from mouse brain tissue and is intended to support biological discovery. We do not foresee direct negative societal consequences; downstream applications should be validated experimentally.

LLM Disclosure

Claude Opus 4.6 (Anthropic) was used for refining the manuscript text. All scientific content, experimental design, analysis, and interpretation are the authors' own.

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.

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022.

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.

Geriskenderi, G. et al. Graph-level representation learning with joint-embedding predictive architectures. *Transactions on Machine Learning Research*, 2024.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Hou, Z., Liu, X., Cen, Y., Dong, Y., Yang, H., Wang, C., and Tang, J. GraphMAE: self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 594–604, 2022.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.

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.

Kreuzer, D., Beaini, D., Hamilton, W. L., Létourneau, V., and Tossou, P. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21618–21629, 2021.

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

Liu, J., Yang, C., Lu, Z., Chen, J., Li, Y., Zhang, M., Bai, T., Fang, Y., Sun, L., Yu, P. S., and Shi, C. Towards graph foundation models: a survey and beyond. *arXiv preprint arXiv:2310.11829*, 2024.

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.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019.

Mialon, G., Chen, D., Selosse, M., and Mairal, J. GraphiT: encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.

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

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14501–14515, 2022.

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.

Thakoor, S., Tallec, C., Azar, M. G., Munos, R., Veličković, P., and Valko, M. Large-scale representation learning on graphs via bootstrapping. In *International Conference on Learning Representations*, 2022.

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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

Xia, J., Wu, L., Chen, J., Hu, B., and Li, S. Z. SimGRACE: a simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference*, pp. 1070–1079, 2022.

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.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do transformers really perform badly for graph representation learning? In *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888, 2021.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5812–5823, 2020.

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. Tokenization Pipeline Details

We explore two tokenization strategies that differ in how local subgraph neighborhoods are represented (Figure 1a–d vs. Supplementary Figure S1). The *subgraph tokenizer* (default) preserves per-node structure: each of the $k+1$ nodes contributes its own segment of L gene tokens, with segment embeddings distinguishing the index node from each neighbor. This retains node-level resolution within the subgraph. The *aggregated neighborhood tokenizer* (Supplementary Figure S1) instead sums gene expression across all nodes before tokenization, producing a single-segment sequence. While more compact, this variant loses the ability to distinguish individual node contributions. Within each strategy, we ablate three encoding schemes for node features: rank-only (ordinal position), counts-only (raw expression value via MLP), and combined (rank + counts). Two graph positional encoding options are compared: rank-based segment embeddings that encode distance-ranked neighbor ordering via sinusoidal functions, and coordinate-based encodings that encode absolute relative x/y coordinates.

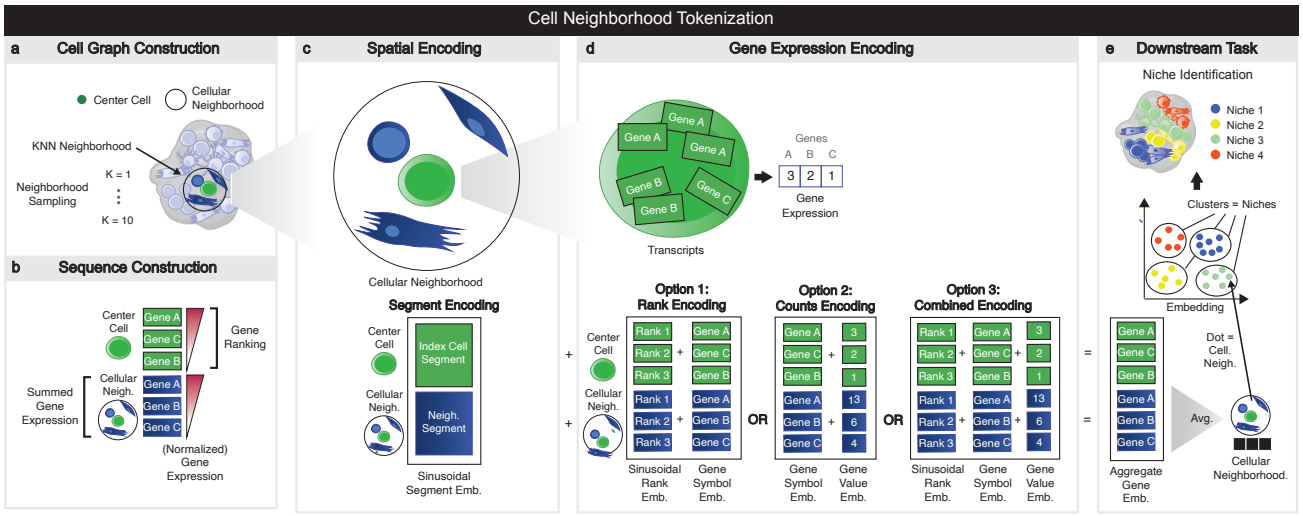

Figure S1. **Aggregated neighborhood tokenization (alternative). (a)** Cell graph construction: center node and neighborhood sampled via k-NN. **(b)** Sequence construction: gene expression is summed across the neighborhood and then ranked, producing a single-segment sequence (contrast with the multi-segment subgraph tokenizer in Figure 1b). **(c)** Graph positional encoding via sinusoidal segment embeddings (index node vs. neighborhood segments). **(d)** Node feature encoding options: rank, counts, and combined. **(e)** Downstream task: niche identification from aggregated neighborhood embeddings.

B. Data Statistics

The choice of sequence length L (number of feature tokens per node) is constrained by the empirical distribution of non-zero features across graphs and feature sets. Supplementary Figure S2 shows these distributions for MERFISH and STARmap under both full (technology-specific) and shared (overlapping) gene panels. When using full panels, MERFISH nodes express a median of approximately 60 genes and STARmap nodes approximately 40, meaning that large sequence lengths ($L \geq 128$) introduce substantial zero-padding. Specifically, only 44.2% of MERFISH nodes and 21.9% of STARmap nodes have $\geq$64 non-zero features under full panels. However, evaluation uses only the 431 shared features, where the distributions are denser: 95.3% of MERFISH nodes and 83.1% of STARmap nodes have $\geq$64 shared features. The default $L = 64$ balances feature coverage against padding overhead.

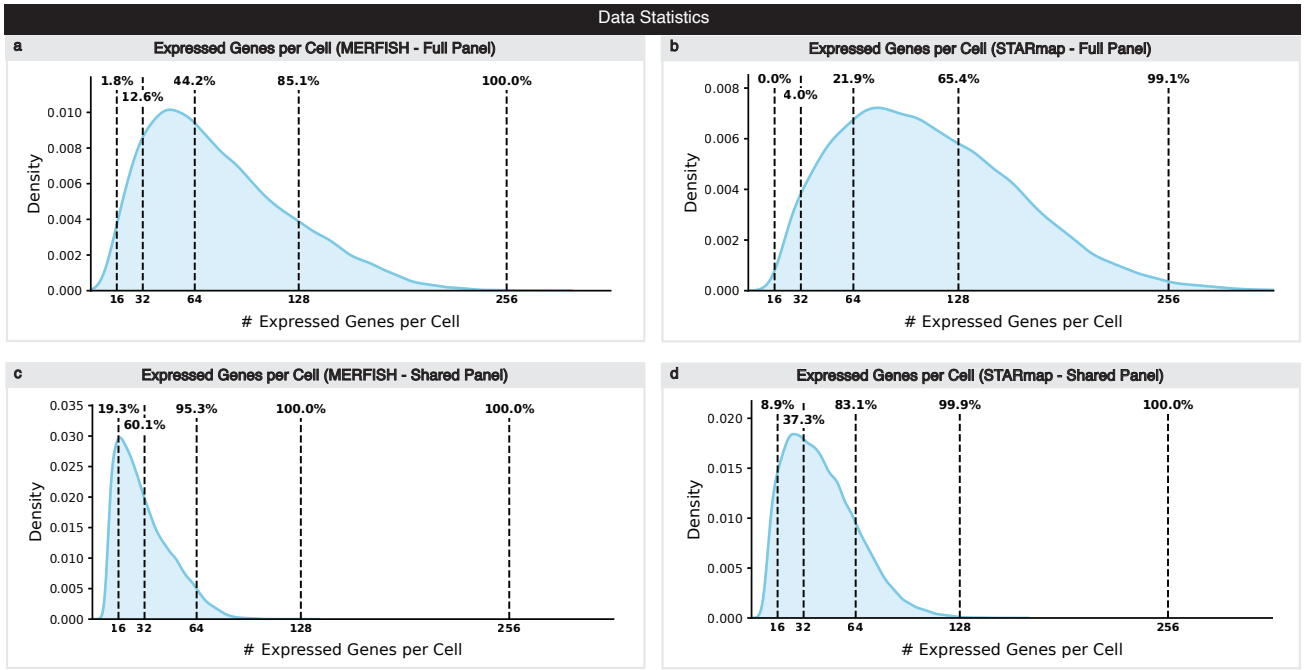

Figure S2. **Feature distribution per node.** Distribution of non-zero features (expressed genes) per node for both graphs using full and shared feature sets. Vertical dashed lines indicate cumulative percentages at sequence length thresholds ($L = 16, 32, 64, 128, 256$). **(a)** MERFISH full panel: 44.2% at $L \geq 64$. **(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 $L = 64$.

C. Extended Ablation Sweeps

Supplementary Figure S3 complements Figure 4 (which reports NMI and iLISI) by showing per-technology ARI and cross-technology MMD across all design axes. For the *graph tokenization* axis, subgraph tokenization consistently outperforms aggregated tokenization across all encoding schemes. For *feature encoding*, MLP projection outperforms value embeddings at all dimensions, with the gap widening at lower resolutions. The *positional encoding* comparison confirms the main-text finding: coordinate-based encoding fails on niche identification despite achieving low MMD. The *sequence length* sweep reveals $L = 32$ achieves the best overall balance, though $L = 64$ (default) yields competitive ARI. For *masking ratio*, $\rho = 0.6$ is optimal; ratios below 0.5 exhibit high variance.

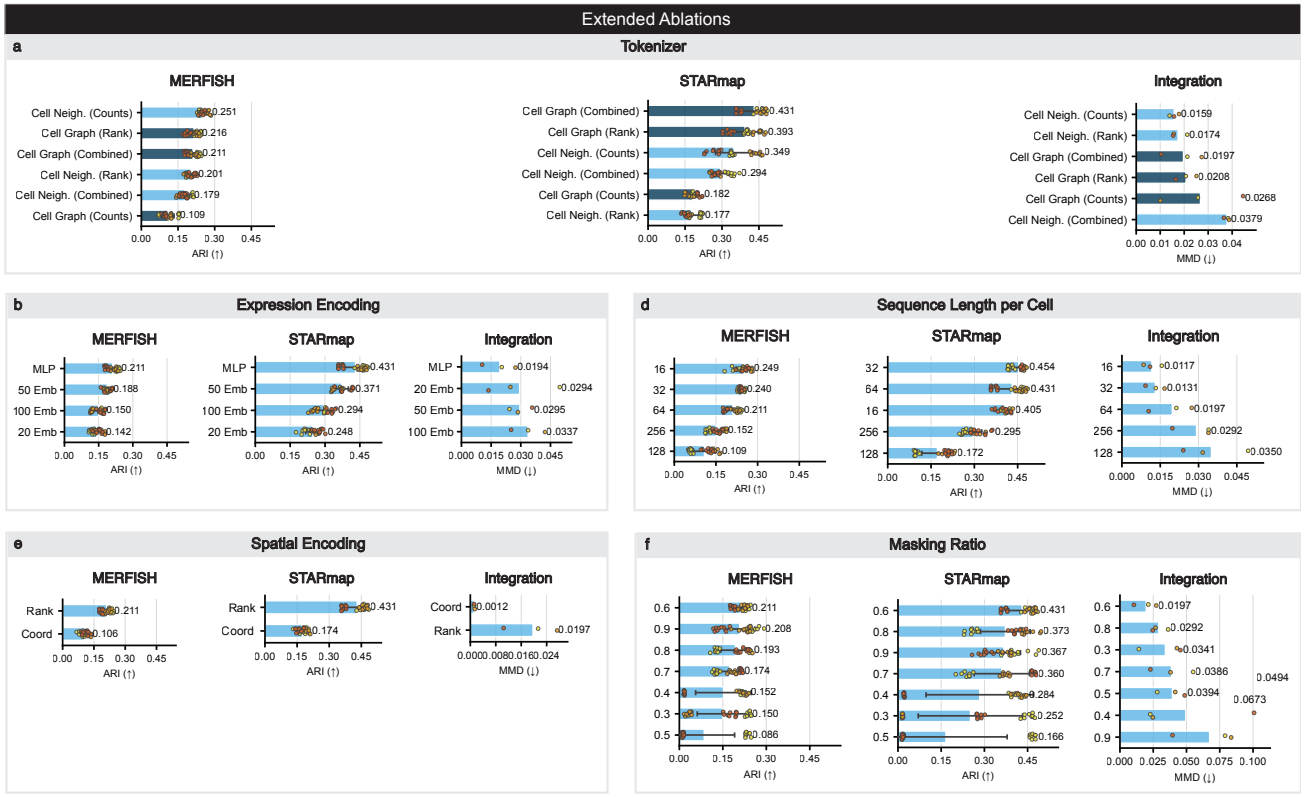

Figure S3. **Extended ablation results.** Complementary to Figure 4 (which reports NMI and iLISI), this figure shows per-technology ARI (MERFISH, STARmap) and cross-technology MMD across six design axes. **(a)** Graph tokenization variants. **(b)** Feature encoding. **(c)** Graph positional encoding. **(d)** Sequence length ($L = 16$–128). **(e)** Positional encoding (expanded). **(f)** Masking ratio ($\rho = 0.3$–0.9).

D. Normalization Strategy Ablation

Gene expression normalization is a critical preprocessing step, yet its interaction with learned graph tokenization schemes is poorly understood. We systematically evaluate 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. The key finding is that no normalization (raw values) achieves the best niche identification across both graphs. This is likely because rank-based tokenization already provides implicit normalization—ranking features by value is inherently robust to scaling differences. Applying additional normalization (particularly Seurat V3 or Pearson residuals) distorts the rank ordering and degrades performance, though Seurat V3 achieves the best integration (iLISI=0.69).

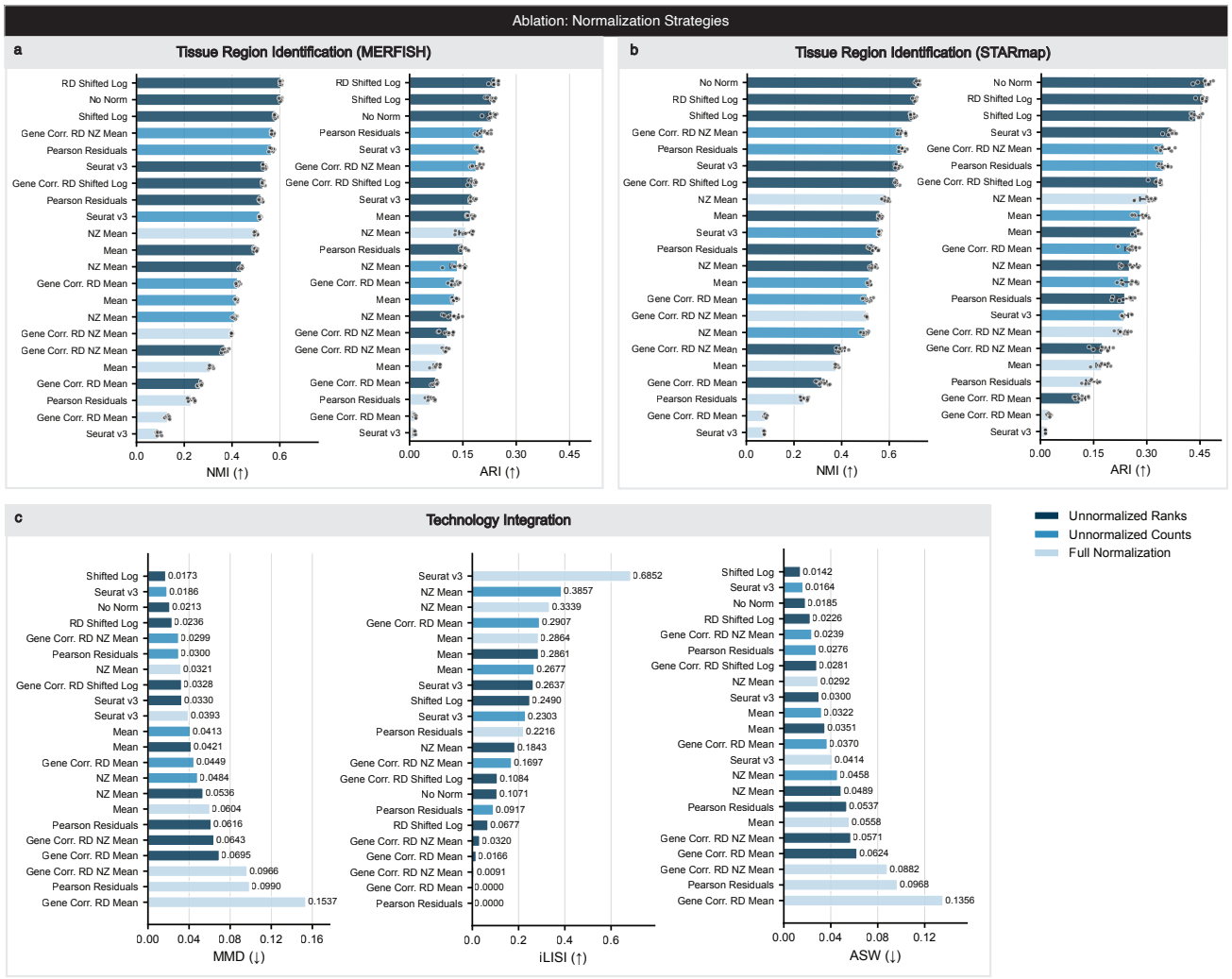

Figure S4. **Normalization strategy ablation.** Comprehensive comparison of normalization strategies for node feature encoding. All variants use combined tokenization (rank + counts); normalizations can be applied independently to either component. Color coding: unnormalized ranks (dark), unnormalized counts (medium), full normalization (light). **(a)** MERFISH niche identification (NMI, ARI). **(b)** STARmap niche identification (NMI, ARI). **(c)** Cross-graph integration (MMD, iLISI, ASW). Raw values with combined tokenization achieve the best niche identification; Seurat V3 achieves the best integration.