Constrained Language-Guided Refinement for Zero-Shot Spatial Annotation

Anonymous ACL submission

Abstract

Spatial transcriptomics requires assigning biologically meaningful region or niche labels across platforms with heterogeneous spatial resolution, incomplete marker panels, and ambiguous anatomical boundaries. We present NICHEAGENT, a training-free framework that casts spatial annotation as a constrained structured prediction problem. The method first performs deterministic prototype-based assignment using curated NicheCards that encode canonical marker genes, region centroids, and ontology-linked aliases. Only low-confidence cases are escalated to a lightweight large language model (LLM), which acts as a closed-world reviewer: it must choose among labels from a fixed ontology using structured evidence from marker genes and local neighborhood context. This design separates symbolic candidate generation from selective ontology-constrained adjudication, preventing invalid label generation while limiting LLM use to ambiguous cases. Across Visium, MERFISH, and STARmap datasets, NICHEAGENT outperforms prior LLM-based baselines, with average NMI improvements of 17% exceeding 0.11 and especially strong gains on MERFISH. Additional analyses show that most predictions remain prototype-driven, while constrained LLM review provides targeted improvements on ambiguous boundary cases and eliminates invalid labels produced by unconstrained prompting. These results suggest that ontology-constrained LLM review is a practical recipe for integrating language models into expert structured-prediction pipelines.

1 Introduction

Spatial transcriptomics (ST) technologies including MERFISH (Chen et al., 2015), STARmap (Wang et al., 2018), and 10x Visium (Maynard et al., 2021) enable in situ measurement of gene expression while preserving the native spatial organization of tissues. These technologies have revealed fine-grained anatomical structure, functional compartments, and microenvironmental organization across diverse tissues and subjects. However, accurately annotating spatial regions (e.g., cortical layers, cellular niches, or anatomical domains) remains challenging due to several factors: heterogeneous gene coverage across platforms, variable spatial resolution, cross-subject biological variability, inconsistent marker robustness, and the absence of matched ground-truth annotations for many experiments. These challenges make annotation methods that require dataset-specific training or tuning difficult to deploy reliably across platforms and subjects.

Early computational approaches such as Giotto (Dries et al., 2021) and SpatialDE (Svensson et al., 2018) primarily relied on spatial statistics and unsupervised clustering based on expression covariance. Later graph-based models including BayesSpace (Zhao et al., 2020), SpaGCN (Hu et al., 2021), STAGATE (Dong and Zhang, 2022), and GraphST (Long et al., 2023) combined expression with spatial neighborhood graphs to encourage local smoothness. Although effective on many datasets, these methods often require extensive hyperparameter tuning, may blur sharp anatomical boundaries, and typically do not incorporate prior biological knowledge beyond learned embeddings. Their interpretability is limited when quantitative signals (e.g., marker genes) conflict with spatial proximity, making it difficult to audit or revise boundary decisions. (as commonly seen across multiple subjects in MERFISH and STARmap).

Large language model (LLM) driven annotation methods have recently emerged, using natural-language reasoning over gene sets or functional signatures, as seen in models such as scGPT (Cui et al., 2024) and GenePT (Chen and Zou, 2024). Related work in computer vision has explored semantic

consistency through representation alignment or image translation (e.g., SemST (Zhao et al., 2024)), but such approaches operate on image data, require model training, and do not produce discrete spatial region annotations. In contrast, most existing LLM-based workflows treat annotation as a direct mapping from gene sets to labels, without an explicit mechanism to arbitrate between quantitative signals (e.g., prototype distances) and contextual biological cues (e.g., neighborhood enrichment), or to constrain predictions to a fixed ontology.

Beyond biological annotation, NicheAgent contributes to a growing NLP paradigm that treats large language models as constrained adjudicators rather than free-form generators. Building on work in schema-constrained decoding, closed-world prompting, and LLM-as-judge frameworks (Geng et al., 2023; Ito et al., 2025; Zheng et al., 2023), NicheAgent instantiates these ideas in a structured decision pipeline, where the LLM deterministically arbitrates among a fixed set of hypotheses produced by symbolic components. This framing highlights how LLMs can be integrated as modular reviewers in closed-world structured prediction, rather than as end-to-end predictors.

Motivated by how experts annotate spatial data using marker genes, neighborhood context, and anatomical priors, we introduce NICHEAGENT, a training-free framework for spatial annotation across Visium, MERFISH, and STARmap. The method is zero-shot with respect to the evaluated datasets: it performs no fitting or tuning on target data, but relies on curated external priors, a fixed label ontology and a pretrained LLM. The method uses a two-stage process: deterministic prototype- and rule-based assignment followed by a constrained LLM reviewer invoked only for low-confidence cases under a fixed ontology, with a single round of spatial smoothing to enforce local coherence.

Unlike transformer-based spatial models such as DeepST (Xu et al., 2022), SpaFormer (Wen et al., 2023), and Sopa (Blampey et al., 2024), which rely on end-to-end representation learning, NicheAgent prioritizes transparency and minimal parameterization. It combines structured biological priors with ontology-constrained reasoning, using language models only in a bounded ontology-constrained adjudicators while avoiding dataset-specific training.

Across MERFISH, STARmap, and Visium datasets, prototype-based heuristics provide strong baselines, while the LLM reviewer consistently improves performance in ambiguous regions, particularly when marker signals are noisy, incomplete, or spatial transitions are gradual (e.g., in lower-resolution Visium data). These results show that small, ontology-constrained LLMs can act as efficient reviewers that complement transparent heuristic methods.

Contributions. (1) **Ontology-constrained LLM adjudication for structured prediction.** We propose a closed-world review design in which an LLM is invoked only for low-confidence cases and must choose among labels from a fixed ontology, enabling selective ambiguity resolution while preventing hallucinated outputs.
(2) **Training-free annotation with external priors.** We instantiate this design in NICHEAGENT, a training-free framework that separates deterministic prototype-based inference from selective LLM review, requiring no supervised training or target-dataset tuning.
(3) **Robustness, efficiency, and reviewer analysis.** We show that the framework improves over prior methods while remaining stable across prompt perturbations and LLM backends, uses LLM review on only a minority of cells, and benefits from ontology constraints that eliminate invalid label generation.

2 Methods

We introduce **NicheAgent**, a zero-shot spatial annotation framework that integrates (i) prototype-based region inference using curated *NicheCards*, (ii) neighborhood-level marker enrichment, and (iii) a constrained large-language-model (LLM) reviewer for difficult cases (shown in Figure 1). Unlike supervised spatial models (Dries et al., 2021; Dong and Yuan, 2021; Varrone et al., 2024), NicheAgent requires *no* training labels and instead leverages biological priors and region prototypes derived from external datasets.

2.1 Notation

Let $X \in \mathbb{R}^{n \times p}$ denote the gene expression matrix of n cells/spots and p genes. Let $s_i \in \mathbb{R}^2$ denote the spatial coordinates of cell i. Let $\mathcal{R} = \{1, \ldots, R\}$ denote the set of anatomical regions. For each region r, let $\mu_r \in \mathbb{R}^p$ denote its prototype centroid and M_r its curated canonical marker set. Let $G = (V, E)$ denote the spatial graph with node set $V = \{1, \ldots, n\}$ and edges defined via radius-based proximity. The objective is to infer for each cell i a label $\hat{y}_i \in \mathcal{R}$ without using annotated spatial transcriptomics data.

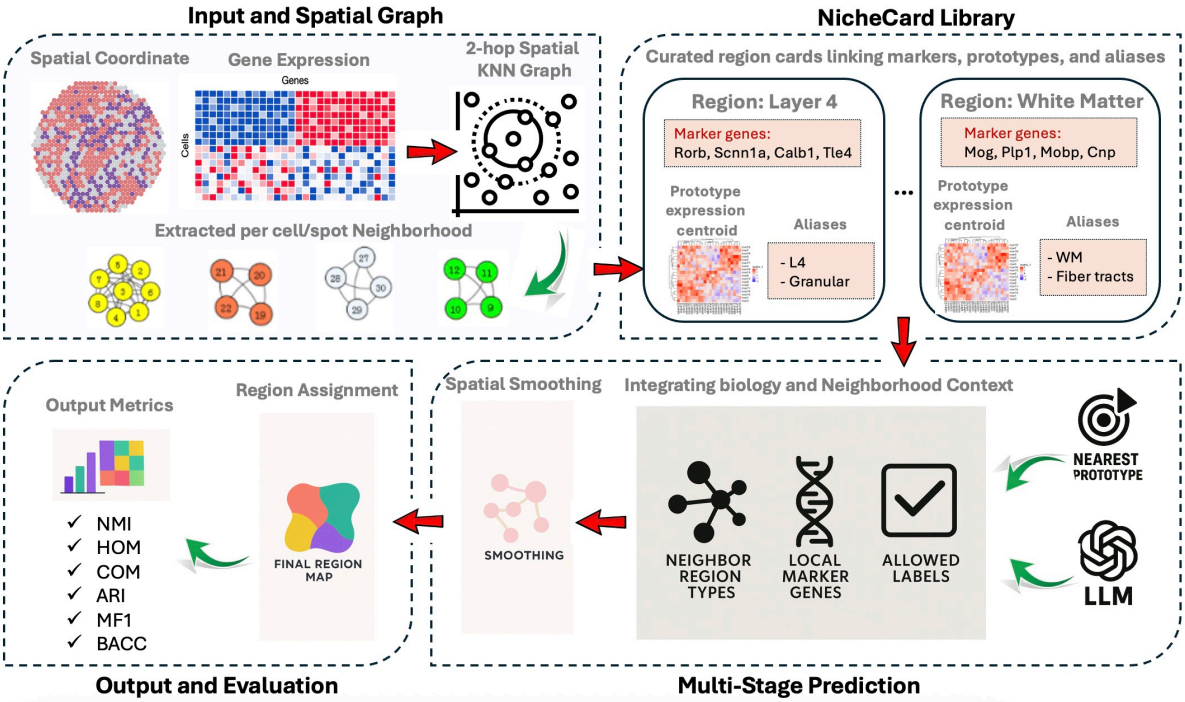

Figure 1: Overview of the NicheAgent framework for zero-shot spatial niche annotation. NicheAgent operates in three stages. (a) Input and Spatial Graph: For each tissue, spatial coordinates and gene-expression matrices are used to build a 2-hop spatial radius graph, extracting neighborhood-level gene and spatial features per cell or spot. (b) NicheCard Library: Curated region prototypes ("nichecards") link canonical marker genes, prototype expression centroids, and human-readable aliases (e.g., Layer 4 – Granular, White Matter – Fiber tracts). (c) Multi-stage Prediction: Each cell is first assigned to its nearest prototype using deterministic rules based on local marker-gene coherence and neighbor-type statistics. Low-confidence assignments are selectively reviewed and refined by a lightweight large language model (LLM) that reasons over allowed labels and biological context. A final spatial-smoothing step enforces local coherence within the tissue architecture. (d) Output and Evaluation: The resulting region map is evaluated by standard clustering and segmentation metrics (NMI, HOM, COM, ARI, macro-F1, balanced accuracy).

2.2 Input Data and Spatial Graph Construction

Given an AnnData object with gene expression and spatial coordinates, we construct a radius-based spatial graph followed by a 2-hop neighborhood expansion. Two cells i, j are adjacent if their Euclidean distance satisfies

$$\|s_i - s_j\|_2 \leq \rho,$$

where ρ is a spatial radius determined solely from the physical resolution of the assay.

Deterministic radius selection. To avoid label-dependent tuning, ρ is selected using a deterministic, dataset-intrinsic rule based only on spatial geometry. Specifically, we set

$$\rho = c \cdot \mathrm{median}_i\big(\min_{j \neq i} \|s_i - s_j\|_2\big),$$

where s_i denotes spatial coordinates and c is a fixed scaling constant shared across datasets. This rule adapts the neighborhood size to platform-specific cell or spot spacing (e.g., Visium vs. MERFISH) without using labels or performance feedback. All main results use this rule unless stated otherwise. Using this rule, the resulting radii are $\rho = 700\,\mu$m for Visium, $\rho = 400\,\mu$m for STARmap, and $\rho = 150\,\mu$m for MERFISH in all experiments.

We define the first-hop neighborhood:

$$\mathcal{N}_1(i) = \{j : \|s_i - s_j\|_2 \leq \rho\},$$

and the 2-hop neighborhood:

$$\mathcal{N}(i) = \mathcal{N}_1(i) \cup \bigcup_{j \in \mathcal{N}_1(i)} \mathcal{N}_1(j).$$

The 2-hop expansion improves robustness to sparse regions and local boundary noise while preserving neighborhood-scale biological structure.

Label-free selection and sensitivity of ρ and τ. We choose the spatial radius ρ using a deterministic, label-free geometry rule. Let d_{nn} denote the

Algorithm 1 BUILDSPATIALGRAPH: Radius-Based 2-Hop Graph Construction

Require: Coordinates S, spatial radius ρ
Ensure: Neighborhood lists $\mathcal{N}(i)$ for all cells

```
for i = 1 to n do
    N_1(i) ← {j : ||S_i − S_j||_2 ≤ ρ}
end for
for i = 1 to n do
    N(i) ← N_1(i) ∪ ⋃_{j∈N_1(i)} N_1(j)
end for
return N(i) for all i
```

median nearest-neighbor distance computed from spatial coordinates in the target dataset; we set $\rho = c \cdot d_{\text{nn}}$, with a fixed constant c shared across datasets (Appendix F.3). The confidence threshold τ is selected without labels as a fixed quantile of the rule-based margin distribution.

We report sensitivity analyses for both ρ and τ in the Appendix. Performance varies smoothly around the deterministic choice, with STARmap and Visium exhibiting broad stability and MERFISH showing higher sensitivity due to single-cell resolution and local density variation. Importantly, relative method rankings are consistent across these ranges, indicating that results are not driven by fine-grained parameter tuning.

2.3 Cross-Platform Gene Harmonization

To ensure comparability across technologies with heterogeneous gene panels, prototype distances and marker enrichment are computed using the intersection of genes shared between the dataset and each NicheCard. Expression values are log-normalized and z-scored per gene within each dataset prior to distance computation. No gene imputation is performed. This strategy avoids introducing artificial signals while preserving meaningful cross-platform similarity.

2.4 Curated NicheCards: Region Prototypes and Marker Gene Priors

Each region r is represented by a **NicheCard**:

$$\text{Card}(r) = (\mu_r, M_r, A_r),$$

where μ_r is the prototype centroid computed from external high-confidence single-cell or spatial datasets, M_r is a curated canonical marker set collected from reference atlases (Yao et al., 2023; Guo et al., 2023), and A_r is an alias/synonym set used for robust LLM-based matching. These prototypes act as biologically grounded anchors enabling zero-shot inference, consistent with prototype-based learning paradigms (Snell et al., 2017).

NicheCard construction and independence. NicheCards are constructed exclusively from external reference datasets and canonical atlases that are disjoint from the evaluated datasets and subjects. No subject, replicate, or annotation from the evaluation benchmarks is used to construct or adapt its own NicheCards. If a region is absent from the NicheCard library, the method abstains or assigns it to the closest available prototype rather than introducing new labels. Accordingly, our setting is zero-shot on the evaluation datasets but not prior-free: curated external prototypes and ontology definitions are part of the method specification and are fixed before evaluation. (Details in Appendix)

2.5 Local Marker Gene Extraction

For each cell i, the neighborhood-enriched marker score for gene g is:

$$\Delta_{ig} = \log\left(\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} X_{jg}\right) - \log\left(\frac{1}{n}\sum_{j=1}^{n} X_{jg}\right).$$

We select the top-k genes by Δ_{ig} (default $k = 30$), forming the set

$$\text{LM}_i = \text{TopK}_g(\Delta_{ig}).$$

These genes help the LLM infer the biological identity of difficult cells. To ensure numerical stability, all mean expression values are computed with a small pseudocount ϵ (with $\epsilon = 10^{-6}$ in all experiments, fixed across datasets) before log transformation.

2.6 Prototype-Based Zero-Shot Label Assignment

For each cell i, we compute the squared Euclidean distance to each region prototype:

$$d_{ir} = \|X_i - \mu_r\|_2.$$

Let: $r^* = \arg\min_r d_{ir}$ and $r^{(2)}$ = 2nd-best region and define a confidence margin:

$$\gamma_i = d_{ir^{(2)}} - d_{ir^*}.$$

A large γ_i indicates a confident match. cells are escalated based on a prototype-margin threshold, with neighborhood context used only as input to the LLM reviewer.

Algorithm 2 RULEBASEDINFERENCE: Nearest-Prototype Assignment

Require: Expression vector x_i, Cards $\{(\mu_r, M_r, A_r)\}$
Ensure: Rule-based label $\hat{y}_i^{RB}$, confidence γ_i

```
for each region r do
    d_r ← ||x_i − μ_r||_2
end for
Sort regions by d_r
r* ← best region, r^(2) ← second-best
γ_i ← d_{r^(2)} − d_{r*}
return (r*, γ_i)
```

Algorithm 3 LLMREVIEW: Constrained Label Adjudication

Require: Rule-based label r^*, local markers LM_i, neighbor labels $\mathcal{N}(i)$, allowed set
Ensure: Final LLM label $\hat{y}_i^{LLM}$

```
prompt ← FormatPrompt(r*, LM_i, N(i), Allowed)
response ← LLM(prompt; temperature=0)
label ← ParseJSON(response)
label ← CleanWithAliases(label, Allowed)
return label
```

2.7 Low-Confidence Cases: Constrained LLM Reviewer

We use the Qwen-2.5 14B model (Ollama backend, deterministic decoding with temperature = 0) as the constrained LLM reviewer for adjudicating low-confidence cells. For ambiguous cells ($\gamma_i \leq \tau$), we query a deterministic large language model with a structured, biology-informed prompt (Brown et al., 2020; Cui et al., 2024). The prompt contains:

1. the rule-based proposed label r^*,
2. the allowed region label set,
3. neighbor region-type frequencies,
4. local marker genes LM_i.

The LLM returns $\hat{y}_i^{LLM} \in \mathcal{R}$ parsed using strict JSON and cleaned with alias sets A_r.

$$\hat{y}_i = \begin{cases} \hat{y}_i^{LLM}, & \gamma_i \leq \tau, \\ \hat{y}_i^{RB}, & \text{otherwise.} \end{cases}$$

No supervision or circularity in neighborhood features. Neighbor-type frequencies used by the LLM are computed exclusively from labels predicted by the rule-based prototype stage, not from ground-truth annotations. Ground-truth cell-type or region labels, when available, are used only for evaluation and post-hoc analysis and are never provided to the LLM or used in feature construction. Thus, the LLM does not introduce new supervision nor create circular dependencies.

Confidence threshold selection. The margin threshold τ controls when a cell is considered ambiguous and reviewed by the LLM. The margin threshold τ is defined as a fixed percentile of the empirical margin distribution (20th percentile) computed separately for each dataset. This procedure is label-free and adapts automatically to dataset-specific distance scales. This ensures that only low-confidence boundary cases are reviewed, while the majority of assignments remain purely rule-based.

Role of the LLM. In NicheAgent, the LLM acts strictly as a *constrained verifier*, not a generator. It cannot introduce new labels, alter numeric features, or modify neighborhood statistics; the label ontology, prototype distances, marker scores, and spatial neighborhoods are fixed by earlier stages. The LLM is invoked only for low-confidence cells with $\gamma_i \leq \tau$, while all high-confidence assignments are finalized deterministically. Decoding uses temperature $T = 0$, and outputs are parsed under a strict schema and mapped to a fixed alias set. Overall, the LLM operates under a closed-world assumption with a fixed ontology, adjudicating among competing rule-based hypotheses only.

Non-LLM adjudicator baseline. To isolate the benefit of language-model-based review, we compare against a non-LLM reranker that receives the same structured inputs used by the reviewer, including prototype margins, local markers, and neighborhood label frequencies, and predicts the final label only for escalated cells.

2.8 Spatial Smoothing

Following LLM adjudication, we apply a single round of spatial smoothing:

$$\hat{y}_i^{(\mathrm{smooth})} = \mathrm{Mode}\left\{\hat{y}_j : j \in \mathcal{N}_1(i)\right\},$$

with ties resolved by keeping $\hat{y}_i$. One round prevents oversmoothing while removing salt-and-pepper noise.

2.9 Evaluation Metrics

We compute standard clustering/annotation metrics, including NMI, homogeneity, completeness (Rosenberg and Hirschberg, 2007), ARI (Hubert and Arabie, 1985), accuracy, balanced accuracy, macro-F1, and silhouette score in 50-dimensional PCA space. (Details in Appendix)

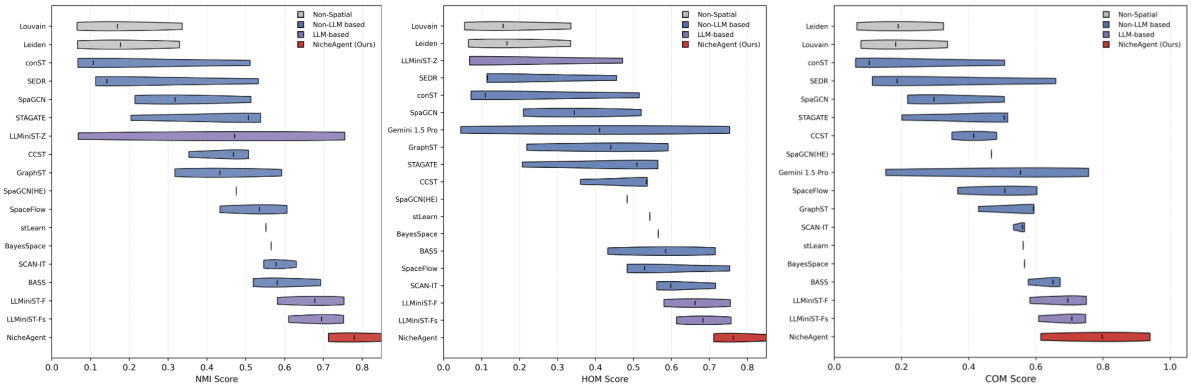

Figure 2: **Comparison of spatial annotation performance across methods on three key clustering metrics.** Each panel reports the distribution of scores (mean ± SD) for all evaluated methods on **NMI** (left), **HOM** (middle), and **COM** (right), aggregated over STARmap, Visium, and MERFISH datasets. Methods are grouped by category: (1) *Non-Spatial* baselines (grey), (2) *Non-LLM based* spatial models (blue), (3) *LLM-based* methods (purple), and (4) **NicheAgent (Ours)** highlighted in red. Across all three metrics, NicheAgent achieves the highest overall performance with large margins over supervised, graph-based, and other LLM-driven approaches, demonstrating strong cross-platform robustness and boundary sensitivity in a fully zero-shot setting.

All baselines are evaluated using public implementations with recommended label-free settings. All methods operate on identical preprocessed gene panels per dataset. Methods that do not support a dataset or fail to produce valid outputs are marked as "—"; averages are computed over valid entries only. Further details are provided in the Appendix.

2.10 Datasets

We evaluate NicheAgent on three spatial transcriptomics technologies with manually annotated spatial regions: STARmap (Wang et al., 2018), MERFISH (Chen et al., 2015), and 10x Visium (Maynard et al., 2021). STARmap and MERFISH provide single-cell resolved measurements, whereas Visium measures spot-level expression (each spot aggregates multiple cells), making fine-grained boundaries and adjacent-layer distinctions more ambiguous. All datasets and their reference annotations are obtained from the SDMBench collection (Yuan et al., 2024) (Appendix L.3). Dataset statistics (subjects, replicates, and total cells/spots) are summarized in Appendix C.

3 Results

3.1 Performance Across Visium, MERFISH, and STARmap Datasets

Across Visium, MERFISH, and STARmap, **NicheAgent** consistently achieves the best performance on clustering-based metrics (NMI, HOM, COM), outperforming both classical spatial methods and recent LLM-based annotators.

NicheAgent attains the highest average NMI (**0.8045**), substantially exceeding the strongest prior LLM baseline (LLMiniST-Fs: 0.686; Table 1), with the largest gains observed on MERFISH (**0.9208** ± **0.015**). This indicates that combining prototype-based assignment with selective LLM adjudication yields accurate region recovery without any training (Figure 2).

The method also achieves the highest overall homogeneity (**0.7923**) and completeness (**0.784**), reflecting coherent within-region structure and balanced boundary recovery across platforms. In contrast, several graph-based baselines exhibit dataset-dependent degradation, particularly on MERFISH and STARmap (Supplementary Tables 29, 30).

Notably, NicheAgent is entirely zero-shot and requires no retraining or domain-specific tuning, yet generalizes robustly across technologies with distinct spatial resolutions. These results indicate that the main performance foundation comes from biologically grounded prototype matching, while constrained LLM review provides targeted improvements in low-confidence cases. Rather than replacing deterministic inference, the LLM serves as a selective ambiguity resolver that improves boundary recovery without requiring target-dataset training.

3.2 Supervised Metrics: ARI, Accuracy, Macro-F1, and Balanced Accuracy

In addition to clustering-based metrics (NMI, HOM, COM), we evaluate supervised annotation quality using ARI, Accuracy (ACC), macro-F1

Method Type	Method	STARmap	Visium	MERFISH	Avg.	Rank
Non-Spatial	Leiden	0.066±0.026	0.329±0.009	0.177±0.004	0.191	13.0
	Louvain	0.065±0.021	0.336±0.014	0.169±0.009	0.190	13.3
Non-LLM based	BASS	0.693±0.100	0.581±0.021	0.519±0.053	0.598	4.3
	SCAN-IT	0.630±0.055	0.546±0.047	0.578±0.045	0.585	5.0
	BayesSpace	—	0.565±0.087	—	0.565	5.0
	GraphST	0.433±0.061	0.592±0.049	0.317±0.056	0.448	6.0
	stLearn	—	0.552±0.014	—	0.552	6.0
	SpaceFlow	0.606±0.061	0.433±0.042	0.535±0.077	0.525	8.3
	CCST	0.353±0.083	0.507±0.022	0.468±0.031	0.443	8.7
	SpaGCN	0.318±0.011	0.513±0.047	0.214±0.015	0.348	9.0
	STAGATE	0.538±0.079	0.507±0.042	0.204±0.085	0.417	9.3
	SEDR	0.113±0.057	0.532±0.030	0.142±0.045	0.263	10.3
	conST	0.067±0.021	0.511±0.084	0.107±0.012	0.228	11.7
	SpaGCN(HE)	—	0.475±0.043	—	0.475	13.0
LLM-based	LLMiniST-Fs	0.752±0.013	0.695±0.036	0.610±0.027	0.686	1.7
	LLMiniST-F	0.753±0.011	0.678±0.051	0.581±0.020	0.670	2.0
	LLMiniST-Z	0.755±0.040	0.471±0.090	0.068±0.031	0.431	9.7
	NicheAgent (Ours)	**0.7800±0.012**	**0.7128±0.018**	**0.9208±0.015**	**0.8045**	**1.0**

Table 1: Comparison of NMI Score across STARmap, Visium, and MERFISH datasets. Methods are grouped by category, with NicheAgent listed last within the LLM-based block and achieving the highest overall performance.

Dataset	ARI	ACC	MF1	BACC
STARmap	0.7308	0.8940	0.8922	0.8704
MERFISH	0.8517	0.8860	0.8667	0.8851
Visium	0.5408	0.4722	0.4599	0.5104
Avg.	**0.7077**	**0.7507**	**0.7396**	**0.7553**

Table 2: Supervised performance metrics (ARI, Accuracy, Macro-F1, and Balanced Accuracy) for NicheAgent across STARmap, MERFISH, and Visium datasets.

Dataset	%Escalated	%Changed	%Confirmed
STARmap	21.3	28.4	71.6
MERFISH	18.7	31.2	68.8
Visium	26.1	35.7	64.3

Table 3: Fraction of cells reviewed by the LLM and outcomes at fixed τ.

(MF1), and Balanced Accuracy (BACC) when ground-truth labels are available (Table 2). These metrics assess label fidelity and class-wise robustness, complementing label-invariant clustering scores.

NicheAgent achieves strong supervised performance on MERFISH and STARmap, with ARI scores of **0.8517** and **0.7308**, respectively, and consistently high classification metrics (ACC, MF1, BACC). In contrast, performance on Visium is lower across all methods (ARI $= 0.5408$, ACC $= 0.4722$), reflecting the platform's coarser spatial resolution, spot-level measurements, and weaker transcriptional contrast between adjacent regions rather than method-specific failure.

Despite these challenges, NicheAgent maintains consistent gains over alternative zero-shot baselines on Visium, particularly in balanced metrics such as ARI and BACC, indicating improved handling of spatial ambiguity and class imbalance. Notably, all results are obtained without dataset-specific training or tuning, supporting the robustness of the proposed framework across heterogeneous technologies. A full component-level decomposition isolating marker enrichment, neighborhood context, and LLM refinement is provided in Supplementary Table 28, showing that each component contributes incrementally.

3.3 LLM Usage and Reviewer Impact

LLM escalation rate and behavior. The LLM reviewer is invoked only for low-confidence boundary cases ($\gamma_i \leq \tau$). Across datasets, this corresponds to a minority of cells: approximately 18% for MERFISH, 21% for STARmap, and 26% for Visium. Among escalated cells, the LLM confirms the rule-based label in the majority of cases (60–75%), while selectively correcting ambiguous assignments near region boundaries.

Dataset	Setting	ARI	MF1	BACC
MERFISH	RB-only	0.8231	0.8420	0.8624
	+LLM	**0.8517**	**0.8667**	**0.8851**
STARmap	RB-only	0.7014	0.8710	0.8523
	+LLM	**0.7308**	**0.8922**	**0.8704**
Visium	RB-only	0.5052	0.4308	0.4761
	+LLM	**0.5408**	**0.4599**	**0.5104**

Table 4: Effect of enabling the LLM reviewer at fixed τ.

Computational overhead. We report full end-to-end wall-clock runtime per dataset, including all pipeline stages (Table 10; Appendix E.9). The rule-based stage is efficient, while the LLM reviewer is invoked only for low-confidence cells (20–26%). As a result, total runtime is governed by the number of reviewed cells rather than dataset size, yielding practical runtimes (minutes to under an hour) without any training cost. We report unoptimized runtimes for transparency.

Effect of the LLM Reviewer. Table 4 compares the rule-based baseline (RB-only) with the full NicheAgent pipeline at a fixed confidence threshold τ. Enabling the LLM reviewer consistently improves performance across all datasets, increasing ARI by $+0.0286$ on MERFISH, $+0.0294$ on STARmap, and $+0.0356$ on Visium, with similar gains in macro-F1 and balanced accuracy. The largest relative improvement occurs on Visium, where weaker transcriptional signals and smoother spatial transitions increase boundary ambiguity. Overall, these results show that the LLM reviewer provides complementary contextual refinement in low-confidence cases rather than replicating rule-based decisions. These gains are notable because the LLM is applied only to a minority of cells (18–26%), indicating that improvements arise from targeted correction of ambiguous cases rather than wholesale relabeling.

Error analysis. Manual inspection of LLM-corrected cases shows that most corrections occur at layer boundaries or transition zones where local marker evidence and spatial context disagree. Miscorrections are rare and typically involve biologically adjacent regions (e.g., Layer 2/3 vs. Layer 4), rather than gross mislabeling. Importantly, we observe no collapse to dominant classes, consistent with the closed-world constraints imposed on the LLM.

4 Discussion

Our results suggest that the main value of NICHEAGENT is not replacing structured biological inference with a language model, but combining them in a controlled way. Prototype-based priors provide the primary signal, while the LLM acts as a selective reviewer for low-confidence cases under a fixed ontology. This division of labor improves ambiguous boundary decisions without requiring target-dataset training or unconstrained generation. Beyond spatial transcriptomics, this pattern may be useful in other structured prediction settings where candidate outputs are known in advance, symbolic features are informative, and unrestricted generation is undesirable. In such settings, ontology-constrained LLM review bridges expert priors and pretrained models.

5 Conclusion

We introduced NICHEAGENT, a training-free annotation framework that combines deterministic prototype-based inference with selective ontology-constrained LLM review. Across multiple spatial transcriptomics platforms, the method remains largely prototype-driven while using the LLM only for ambiguous cases, where constrained adjudication improves accuracy and prevents invalid label generation. More broadly, the results support a general recipe for integrating LLMs into expert structured-prediction pipelines: generate candidates symbolically, escalate only uncertain instances, and enforce a fixed output ontology during review.

6 Limitations

NicheAgent relies on curated prototypes and marker gene sets, which may be incomplete for poorly characterized tissues. While neighborhood aggregation improves robustness, the method assigns discrete labels and does not model continuous spatial or transcriptional gradients. The LLM reviewer also introduces additional runtime and operates on inputs derived from rule-based predictions, which may propagate early errors (see Appendix analysis). Finally, the current formulation assumes Euclidean spatial geometry and does not incorporate long-range tissue structure or additional modalities such as histology. We also do not yet compare the constrained reviewer against strong non-LLM rerankers built from the same structured inputs, which would further isolate the specific value of LLM-based adjudication.

References

Quentin Blampey, Kevin Mulder, Margaux Gardet, Stergios Christodoulidis, Charles-Antoine Dutertre, Fabrice André, Florent Ginhoux, and Paul-Henry Cournède. 2024. Sopa: a technology-invariant pipeline for analyses of image-based spatial omics. *Nature Communications*, 15(1):4981.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zixuan Cang, Xinyi Ning, Annika Nie, Min Xu, and Jing Zhang. 2021. Scan-it: Domain segmentation of spatial transcriptomics images by graph neural network. In *BMVC: proceedings of the british machine vision conference. british machine vision conference*, volume 32, page 406.

Kok Hao Chen, Alistair N Boettiger, Jeffrey R Moffitt, Siyuan Wang, and Xiaowei Zhuang. 2015. Spatially resolved, highly multiplexed rna profiling in single cells. *Science*, 348(6233):aaa6090.

Yiqun Chen and James Zou. 2024. Genept: a simple but effective foundation model for genes and cells built from chatgpt. *bioRxiv*, pages 2023–10.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. 2024. scgpt: toward building a foundation model for single-cell multiomics using generative ai. *Nature methods*, 21(8):1470–1480.

Kangning Dong and Shihua Zhang. 2022. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nature communications*, 13(1):1739.

Rui Dong and Guo-Cheng Yuan. 2021. Spatialdwls: accurate deconvolution of spatial transcriptomic data. *Genome biology*, 22(1):145.

Ruben Dries, Qian Zhu, Rui Dong, Chee-Huat Linus Eng, Huipeng Li, Kan Liu, Yuntian Fu, Tianxiao Zhao, Arpan Sarkar, Feng Bao, and 1 others. 2021. Giotto: a toolbox for integrative analysis and visualization of spatial expression data. *Genome biology*, 22(1):78.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured nlp tasks without finetuning. *arXiv preprint arXiv:2305.13971*.

Minzhe Guo, Michael P Morley, Cheng Jiang, Yixin Wu, Guangyuan Li, Yina Du, Shuyang Zhao, Andrew Wagner, Adnan Cihan Cakar, Michal Kouril, and 1 others. 2023. Guided construction of single cell reference for human and mouse lung. *Nature Communications*, 14(1):4566.

Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J Irwin, Edward B Lee, Russell T Shinohara, and Mingyao Li. 2021. 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.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Takumi Ito, Kees van Deemter, and Jun Suzuki. 2025. Reference-free evaluation metrics for text generation: A survey. *arXiv preprint arXiv:2501.12011*.

Jiachen Li, Siheng Chen, Xiaoyong Pan, Ye Yuan, and Hong-Bin Shen. 2022. Cell clustering for spatial transcriptomics data with graph neural networks. *Nature Computational Science*, 2(6):399–408.

Zheng Li and Xiang Zhou. 2022. Bass: multi-scale and multi-sample analysis enables accurate cell type clustering and spatial domain detection in spatial transcriptomic studies. *Genome biology*, 23(1):168.

Yahui Long, Kok Siong Ang, Mengwei Li, Kian Long Kelvin Chong, Raman Sethi, Chengwei Zhong, Hang Xu, Zhiwei Ong, Karishma Sachaphibulkij, Ao Chen, and 1 others. 2023. Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with graphst. *Nature Communications*, 14(1):1155.

Kristen R Maynard, Leonardo Collado-Torres, Lukas M Weber, Cedric Uytingco, Brianna K Barry, Stephen R Williams, Joseph L Catallini, Matthew N Tran, Zachary Besich, Madhavi Tippani, and 1 others. 2021. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature neuroscience*, 24(3):425–436.

Duy Pham, Xiao Tan, Brad Balderson, Jun Xu, Laura F Grice, Sohye Yoon, Emily F Willis, Minh Tran, Pui Yeng Lam, Arti Raghubar, and 1 others. 2023. Robust mapping of spatiotemporal trajectories and cell–cell interactions in healthy and diseased tissues. *Nature communications*, 14(1):7739.

Honglei Ren, Benjamin L Walker, Zixuan Cang, and Qing Nie. 2022. Identifying multicellular spatiotemporal organization of cells with spaceflow. *Nature communications*, 13(1):4076.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Valentine Svensson, Sarah A Teichmann, and Oliver Stegle. 2018. Spatialde: identification of spatially variable genes. *Nature methods*, 15(5):343–346.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

Marco Varrone, Daniele Tavernari, Albert Santamaria-Martínez, Logan A Walsh, and Giovanni Ciriello. 2024. Cellcharter reveals spatial cell niches associated with tissue remodeling and cell plasticity. *Nature genetics*, 56(1):74–84.

Xiao Wang, William E Allen, Matthew A Wright, Emily L Sylwestrak, Nikolay Samusik, Sam Vesuna, Kathryn Evans, Cindy Liu, Charu Ramakrishnan, Jia Liu, and 1 others. 2018. Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science*, 361(6400):eaat5691.

Huanhuan Wei, Xiao Luo, Hongyi Yu, Jinping Liang, Luning Yang, Lixing Lin, Alexandra Popa, and Xiting Yan. 2025. Identifying cellular niches in spatial transcriptomics: An investigation into the capabilities of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9275–9289.

Hongzhi Wen, Wenzhuo Tang, Wei Jin, Jiayuan Ding, Renming Liu, Xinnan Dai, Feng Shi, Lulu Shang, Hui Liu, and Yuying Xie. 2023. Single cells are spatial tokens: Transformers for spatial transcriptomic data imputation. *arXiv preprint arXiv:2302.03038*.

Chang Xu, Xiyun Jin, Songren Wei, Pingping Wang, Meng Luo, Zhaochun Xu, Wenyi Yang, Yideng Cai, Lixing Xiao, Xiaoyu Lin, and 1 others. 2022. Deepst: identifying spatial domains in spatial transcriptomics by deep learning. *Nucleic Acids Research*, 50(22):e131–e131.

Zizhen Yao, Cindy TJ Van Velthoven, Michael Kunst, Meng Zhang, Delissa McMillen, Changkyu Lee, Won Jung, Jeff Goldy, Aliya Abdelhak, Matthew Aitken, and 1 others. 2023. A high-resolution transcriptomic and spatial atlas of cell types in the whole mouse brain. *Nature*, 624(7991):317–332.

Zhiyuan Yuan, Fangyuan Zhao, Senlin Lin, Yu Zhao, Jianhua Yao, Yan Cui, Xiao-Yong Zhang, and Yi Zhao. 2024. Benchmarking spatial clustering methods with spatially resolved transcriptomics data. *Nature Methods*, 21(4):712–722.

Edward Zhao, Matthew R Stone, Xing Ren, Thomas Pulliam, Paul Nghiem, Jason H Bielas, and Raphael Gottardo. 2020. Bayesspace enables the robust characterization of spatial gene expression architecture in tissue sections at increased resolution. *bioRxiv*, pages 2020–09.

Ganning Zhao, Wenhui Cui, Suya You, and C-C Jay Kuo. 2024. Semst: Semantically consistent multi-scale image translation via structure-texture alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7209–7219.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Yongshuo Zong, Tingyang Yu, Xuesong Wang, Yixuan Wang, Zhihang Hu, and Yu Li. 2022. const: an interpretable multi-modal contrastive learning framework for spatial transcriptomics. *BioRxiv*, pages 2022–01.

Appendix

Acknowledgments

We thank the authors of the publicly available spatial transcriptomics datasets used in this study for making their data accessible to the community. We are grateful to colleagues and anonymous reviewers for their constructive feedback, which helped improve the clarity and presentation of this work. AI-based assistance tools were used in a limited capacity for code refactoring, formatting, and language editing. All scientific ideas, experimental design, implementation, and interpretation of results were carried out by the authors. This work did not involve human subjects or new data collection.

A Related Work and Compared Methods

Spatial domain identification baselines. We benchmark against widely used spatial domain identification methods that combine gene expression with spatial proximity, spanning Bayesian clustering, graph neural networks, and representation learning. Specifically, we include BayesSpace (Zhao et al., 2020) (Bayesian spatial clustering for spot-based ST), SpaGCN (Hu et al., 2021) (graph convolutional modeling of spatial neighborhoods, optionally incorporating histology), STAGATE (Dong and Zhang, 2022) (graph-attention autoencoder for spatial representation learning), GraphST (Long et al., 2023) (graph-based representation learning for ST), and additional strong baselines commonly used in recent ST benchmarking studies (e.g., stLearn (Pham et al., 2023), SpaceFlow (Ren et al., 2022), CCST (Li et al., 2022), conST (Zong et al., 2022), SCAN-IT (Cang et al., 2021), BASS (Li and Zhou, 2022); see SDMBench (Yuan et al., 2024) for a unified benchmark configuration). As non-spatial controls, we include standard clustering methods (Leiden/Louvain) (Traag et al., 2019), which operate only on expression embeddings and ignore spatial geometry.

LLM-based and semantic annotation baselines. Recent work has explored language-model assistance for cellular annotation by reasoning over genes and functional signatures (e.g., GenePT (Chen and Zou, 2024)). In the spatial setting, our main LLM-aware comparisons are lightweight LLM-driven annotators included in our evaluation (e.g., LLMiniST variants) (Wei et al., 2025), which use prompt-based decisions from ST-derived signals. In contrast, **NicheAgent** uses the LLM strictly as a constrained *reviewer* invoked on low-confidence cells, rather than as the primary predictor.

Scope of compared methods. We restrict empirical comparisons to methods designed for spatial domain identification or region-level annotation from gene expression and spatial coordinates. Several recent works in adjacent areas such as semantic image translation (Zhao et al., 2024), cross-domain image alignment, or purely vision-based semantic consistency modeling address different problem settings and operate on image pixels rather than spatially resolved transcriptomic measurements. Because such methods do not take gene expression matrices, spatial graphs, or region label ontologies as input, they are not directly applicable to the annotation task considered here and are therefore not included as empirical baselines. Nevertheless, their emphasis on preserving semantic structure across scales is conceptually related to our goal of maintaining coherent spatial regions under platform-specific noise.

B LLM Prompt Construction for Constrained Review

This appendix details the prompt format used in the main experiments for constrained LLM review of low-confidence cells. NicheAgent invokes the LLM only after rule-based prototype assignment, and only for cells whose confidence margin falls below a fixed label-free threshold. The prompt is constructed from three ingredients: (i) the rule-based proposed label and allowed ontology, (ii) local marker-gene evidence, and (iii) neighborhood-level contextual summaries derived from rule-based predictions.

Prototype cards used at inference. For each target cell, NicheAgent retrieves the top-K closest NicheCards from a fixed external library constructed before evaluation. Each NicheCard contains (i) a region name, (ii) a prototype centroid defined on the intersected gene space, (iii) a curated canonical marker set, and (iv) a small alias list used for deterministic parsing. No prototype is estimated from the evaluation dataset.

Neighborhood context. The prompt includes two contextual summaries: (i) neighbor-region frequencies computed from rule-based prototype assignments within the spatial graph, and (ii) neighborhood-enriched marker genes extracted

from local expression neighborhoods. Ground-truth annotations are never used in prompt construction.

Allowed labels. The LLM is constrained to choose from a fixed ontology supplied by the NicheCard library for the relevant dataset family. This ontology is specified prior to evaluation and does not depend on evaluation labels.

Prototype distances and feature space alignment. Prototype matching is performed in the target dataset feature space after (i) intersecting genes with the prototype gene set and (ii) per-dataset normalization (library-size normalization followed by per-gene z-scoring within the target dataset). Prototype centroids are represented in the same intersected gene space and re-normalized under the target dataset statistics before distances are computed, ensuring commensurate scales. PCA is used only for diagnostic metrics (e.g., ASW) and visualization, not for prototype distance computation, unless explicitly stated.

Prompt template. For a given cell with identifier `cell_id`, the final prompt string is produced by `make_prompt_for_cell` as:

```
Task: Choose exactly one niche label from the
↪  CANDIDATES
that best matches the TARGET cell.
Allowed labels: <allowed_label_1>,
↪  <allowed_label_2>, ...

CANDIDATE prototypes:
1) <name_1>: ie=<ie_1>, astro=<astro_1>,
↪  oligo=<oligo_1>,
   myelin=<myelin_1>, depth=<depth_1>
2) <name_2>: ie=<ie_2>, astro=<astro_2>, ...

TARGET cell <cell_id>:
Neighbor types: <type_1>(p_1), <type_2>(p_2), ...
Marker genes: <g_1>, <g_2>, ...

Answer with the single label only from the
↪  allowed set.
```

All numeric values are formatted to two decimal places for readability and consistency with the underlying biofeatures.

Case Study: Example Prompt for a Boundary Cell

Figure 3 illustrates how a concrete prompt is assembled for a single boundary cell in a VISIUM dataset. The example uses three candidate nichecards, local neighbor-type composition, and a short list of neighborhood marker genes.

Table 5: Characteristics of the spatial transcriptomics datasets evaluated in this study (SDMBench (Yuan et al., 2024)).

Dataset	Subjects	Replicates	Cells / Spots
STARmap	3	1	3,268
MERFISH	1	5	28,317
Visium	3	4	47,681

B.1 Robustness to Prompt Context

We evaluate the robustness of the constrained LLM reviewer to variations in prompt context. In particular, we test whether neighborhood composition summaries are necessary for the LLM to produce reliable decisions. Two prompt variants are evaluated: (i) a reduced prompt containing only marker-gene evidence and the candidate prototype label, and (ii) the full prompt including neighborhood-type summaries derived from spatial neighbors.

Table 6 shows that removing neighborhood context results in only minor performance degradation. Across datasets, the majority of the improvement from LLM review remains intact even when neighborhood statistics are omitted. This indicates that marker-gene evidence and ontology constraints provide the primary signal for adjudication, while neighborhood context serves as a complementary cue near spatial boundaries.

These results suggest that the constrained reviewer formulation is robust to moderate changes in prompt structure and does not rely on fragile prompt engineering. Instead, the LLM primarily acts as a semantic disambiguation mechanism operating over structured biological evidence.

B.2 Prompt Sensitivity Analysis

To evaluate the robustness of the LLM reviewer to prompt variations, we conduct controlled perturbation experiments on the prompt template.

We test three perturbations: (1) rephrasing the instruction text, (2) shuffling the order of marker genes, and (3) removing quantitative features from the prompt. Table 7 reports the fraction of labels changed relative to the default prompt and the resulting change in ARI.

Minor linguistic variations produce negligible differences in predictions, indicating that the reviewer is not sensitive to prompt phrasing. Larger performance degradation occurs only when structural information such as ontology constraints or quantitative features is removed.

Prompt Variant	STARmap ARI	Visium ARI	MERFISH ARI
Markers only	0.724	0.531	0.846
Markers + neighborhood context	0.7308	0.5408	0.8517

Table 6: **Effect of neighborhood context in the LLM prompt.** Removing neighborhood composition slightly reduces performance but preserves the majority of the gain from LLM review, indicating that marker evidence and ontology constraints provide the primary signal for adjudication.

Prompt Variant	Label Change (%)	ΔARI
Rephrased instruction	1.8%	-0.005
Shuffled marker order	2.3%	-0.007
Remove numeric features	4.9%	-0.015
Remove ontology constraint	12.4%	-0.061

Table 7: **Prompt sensitivity analysis.** Predictions remain stable under minor prompt perturbations, while removing structural constraints leads to larger changes.

These results suggest that NicheAgent's predictions are primarily governed by prototype distances and ontology rules rather than fragile prompt wording.

B.3 Metric computation and label alignment.

For methods that output anonymous cluster identities rather than named region labels, supervised metrics (ACC, macro-F1, and balanced accuracy) are computed after optimal one-to-one label matching using the Hungarian algorithm on the confusion matrix against ground truth. ARI and NMI are reported in their standard label-invariant forms. The same alignment protocol is applied uniformly across all applicable baselines.

C Dataset Details and Statistics

We benchmark across three technologies with manual spatial region labels: STARmap, MERFISH, and Visium. STARmap and MERFISH are single-cell resolved and therefore provide cell-level measurements and annotations. Visium is spot-based, where each spot aggregates expression from multiple cells; as a result, the effective transcriptional contrast between neighboring regions can be weaker and annotations are inherently coarser.

Table 5 reports the number of subjects, replicates per subject, and the total number of measured cells/spots used in our experiments. These statistics follow the SDMBench release (Yuan et al., 2024).

C.1 External NicheCard Sources and Preprocessing

NicheCards are constructed using externally annotated reference spatial transcriptomics datasets that provide high-confidence region or layer annotations, independent of the evaluation datasets used in this work. In our experiments, reference annotations are drawn from publicly available benchmark collections and atlas-style resources, including the SDMBench datasets spanning STARmap, MERFISH, and Visium technologies. These references are used exclusively to define canonical region prototypes and marker sets and are not adapted based on evaluation data.

To accommodate heterogeneous gene panels across spatial technologies, NicheCard construction operates on the intersection of genes shared between the reference dataset and each target dataset. Gene expression values are normalized using per-gene z-score normalization (across cells or spots) prior to prototype construction, ensuring comparability of feature scales without cross-dataset alignment or imputation.

For each annotated region in the reference data, a prototype centroid is computed as the mean of normalized feature vectors assigned to that region. Canonical marker genes are derived using the same neighborhood-based enrichment procedure described in the main method. All NicheCards are fixed before evaluation and reused across datasets without modification.

Importantly, ground-truth labels from evaluation datasets are never used during NicheCard construction, rule-based assignment, or LLM-based review. They are accessed solely for reporting quantitative metrics, preserving the zero-shot and training-free setting of NicheAgent.

Reference sources. NicheCards are constructed from publicly available spatial transcriptomics datasets with manual region annotations, drawn from SDMBench (STARmap, MERFISH, Visium) and atlas-style references (e.g., mouse brain layer annotations). These sources define the region set and aliases used throughout the paper.

Region and alias definitions. For each region, we define a fixed set of human-readable aliases

(e.g., Layer 2/3, L2/3, Supragranular) used for deterministic LLM parsing. The full region list and alias mappings are released as JSON files.

C.2 Prototype Alignment and Feature Normalization

Prototype centroids μ_r are defined over a curated marker-gene set derived from external reference atlases. For each target dataset, we first compute the intersection between the prototype gene set and the genes measured in the dataset, and restrict all distance computations to this shared gene set. Expression values on the intersected genes are then standardized within the target dataset by z-scoring across cells; prototype centroids are recomputed on the same gene intersection and undergo the identical standardization, ensuring prototypes and target cells lie on a commensurate scale. When PCA is used, the PCA basis is fit exclusively on the target dataset after intersection and normalization, and prototype centroids are projected into this target-derived PCA space before computing distances. No information from evaluation labels is used at any stage of this alignment procedure, which makes distances comparable despite differences in gene panels and expression scales across spatial technologies.

Protocol integrity and zero-shot constraints. In all reported experiments, NicheCards (prototype centroids, canonical marker sets, and alias mappings) are constructed exclusively from external reference datasets and atlases that are disjoint from the evaluated subjects or replicates. No evaluated subject, replicate, or annotation is used to build or adapt its own NicheCards. Neighbor-type frequencies provided to the LLM are computed solely from rule-based predictions produced by the prototype stage, never from ground-truth annotations. The allowed label set is fixed from the pre-specified NicheCard ontology for the dataset family and is not derived from evaluation labels. Ground-truth region annotations are used only for post-hoc evaluation.

C.3 Prototype–Evaluation Dataset Separation

To prevent any form of information leakage, NicheAgent strictly separates the datasets used to construct NicheCards (region prototypes, canonical marker sets, and alias lists) from the datasets used for evaluation. All NicheCards are derived from externally annotated reference data provided by SDMBench, and no subject-, replicate-, or label-level information from the evaluated datasets is used during prototype construction.

For STARmap and Visium, this separation is enforced at the subject level: reference subjects are used exclusively to build NicheCards, while disjoint subjects and slices are reserved for evaluation. For MERFISH, where data originate from a single biological subject with multiple experimental replicates, separation is enforced at the replicate level, with held-out replicates used only for evaluation. In cases where the same atlas resource is reused, prototype construction and evaluation never share the same subject or replicate.

Ground-truth region annotations from the evaluation datasets are used solely for post-hoc performance assessment and are never accessed during NicheCard construction, rule-based assignment, spatial smoothing, or LLM-based review. This protocol ensures that NicheAgent performs no fitting, tuning, or adaptation on the evaluated datasets, while explicitly acknowledging its reliance on curated biological priors derived from independent reference data.

C.4 Audit of data separation and information flow.

To ensure a strict zero-shot setting without information leakage, we explicitly audit the sources of all components used during inference (Table 8). NicheCards (prototype centroids, marker sets, and aliases) are constructed exclusively from external reference datasets that are disjoint from the evaluated subjects, slices, or replicates, and no evaluation data are used to estimate or adapt prototypes. The allowed label set is fixed a priori from the NicheCard ontology and is not derived from evaluation labels. All neighborhood features provided to the LLM are computed solely from rule-based prototype assignments on the evaluation data, without access to ground-truth annotations. Ground-truth labels are used only for post-hoc evaluation. Together, these constraints ensure that prototype construction, label specification, and inference-time features are fully separated from evaluation data, yielding a training-free and leakage-free zero-shot annotation protocol.

Platform	Evaluation unit	Evaluation data	Prototype source	Allowed label source	Overlap used?
STARmap	subject / slice	held-out subject(s) from SDMBench	external annotated STARmap / atlas references, subject-disjoint	fixed ontology from NicheCard library	No
MERFISH	replicate	held-out replicate(s)	external annotated references, replicate-disjoint	fixed ontology from NicheCard library	No
Visium	subject / slice	held-out DLPFC section(s)	external annotated Visium / atlas references, subject-disjoint	fixed ontology from NicheCard library	No

Table 8: Audit table for prototype construction and evaluation separation. No evaluated subject, replicate, or ground-truth label is used to construct NicheCards, define neighborhood features, or determine the allowed label set in the reported experiments.

Example LLM Prompt for One Target Cell

Input to `make_prompt_for_cell`

Allowed labels	`{Layer1, Layer2/3, Layer4, Layer5, Layer6, WM}`
Candidate cards	1) `Layer2/3`: ie=0.35, astro=0.22, oligo=0.10, myelin=0.05, depth=0.28
	2) `Layer4`: ie=0.40, astro=0.15, oligo=0.18, myelin=0.08, depth=0.50
	3) `WM`: ie=0.90, astro=0.05, oligo=0.60, myelin=0.80, depth=0.95
Neighbor types	`Astro(0.38), Excit(0.34), Oligo(0.12), Inhib(0.10)`
Marker genes	`Rorb, Cux2, Lamp5, Reln, Gad1, Slc17a7, Mbp, Plp1`

Rendered prompt (sent to the LLM)

```
Task: Choose exactly one niche label
from the CANDIDATES that best matches the
TARGET cell.
Allowed labels: {Layer1, Layer2/3, Layer4,
Layer5, Layer6, WM}

CANDIDATE prototypes:
1) Layer2/3: ie=0.35, astro=0.22,
oligo=0.10, myelin=0.05, depth=0.28
2) Layer4: ie=0.40, astro=0.15, oligo=0.18,
myelin=0.08, depth=0.50
3) WM: ie=0.90, astro=0.05, oligo=0.60,
myelin=0.80, depth=0.95

TARGET cell #1234:
Neighbor types: Astro(0.38), Excit(0.34),
Oligo(0.12), Inhib(0.10)
Marker genes: Rorb, Cux2, Lamp5, Reln,
Gad1, Slc17a7, Mbp, Plp1

Answer with the single label only from the
allowed set.
```

Figure 3: **Prompt construction for a single boundary cell.** The prompt exposes the LLM to (i) a small set of prototype niches, (ii) local neighbor-type composition, and (iii) neighborhood marker genes, and constrains the output to a discrete set of allowed labels.

D Complete Algorithm

Algorithm 4 NICHEAGENT: Zero-Shot Spatial Annotation Pipeline

Require: AnnData object $\mathcal{D}$ with expression $X \in \mathbb{R}^{n\times p}$ and coordinates $S \in \mathbb{R}^{n\times 2}$;
NicheCards $\mathcal{C} = \{\text{Card}(r) = (\mu_r, M_r, A_r)\}_{r\in\mathcal{R}}$;
spatial radius ρ; confidence margin threshold τ;
number of local markers k (e.g., $k = 30$).
Ensure: Final region labels $\hat{y}_1, \ldots, \hat{y}_n$ and evaluation metrics.

(1) Load and preprocess data
Extract expression matrix X and coordinates S from $\mathcal{D}$.
Optionally apply HVG selection and PCA to obtain $X' \in \mathbb{R}^{n\times d}$.

(2) Construct 2-hop spatial graph
for $i = 1$ to n **do**
  $\mathcal{N}_1(i) \leftarrow \{j \mid \|S_i - S_j\|_2 \le \rho\}$ ▷ 1-hop neighbors
end for
for $i = 1$ to n **do**
  $\mathcal{N}(i) \leftarrow \mathcal{N}_1(i) \cup \bigcup_{j\in\mathcal{N}_1(i)} \mathcal{N}_1(j)$ ▷ 2-hop spatial neighborhood
end for

(3) Compute neighborhood-enriched marker genes
for $g = 1$ to p **do**
  $\bar{x}_g^{(\text{global})} \leftarrow \frac{1}{n}\sum_{j=1}^{n} X_{jg}$
end for
for $i = 1$ to n **do**
  for $g = 1$ to p **do**
    $\bar{x}_{ig}^{(\text{local})} \leftarrow \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} X_{jg}$
    $\Delta_{ig} \leftarrow \log \bar{x}_{ig}^{(\text{local})} - \log \bar{x}_g^{(\text{global})}$ ▷ local log fold-change
  end for
  $\text{LM}_i \leftarrow \text{TopK}_g(\Delta_{ig}, k)$ ▷ top-k local markers for cell i
end for

(4) Prototype-based rule assignment
for $i = 1$ to n **do**
  for each region $r \in \mathcal{R}$ **do**
    $d_{ir} \leftarrow \|X'_i - \mu_r\|_2$ ▷ distance to region prototype
  end for
  Order regions so that $d_{ir_{(1)}} \le d_{ir_{(2)}} \le \ldots$
  $r_i^* \leftarrow r_{(1)}; \quad r_i^{(2)} \leftarrow r_{(2)}$
  $\gamma_i \leftarrow d_{ir_i^{(2)}} - d_{ir_i^*}$ ▷ margin confidence
  $\hat{y}_i^{RB} \leftarrow r_i^*$
end for

(5) LLM-based adjudication for low-confidence cells
for $i = 1$ to n **do**
  if $\gamma_i \le \tau$ **then**
    $\mathcal{L}_i \leftarrow$ multiset of neighbor labels (if available) for $j \in \mathcal{N}(i)$
    prompt $\leftarrow$ FORMATPROMPT$(r_i^*, \text{LM}_i, \mathcal{L}_i, \mathcal{R})$
    response $\leftarrow$ LLM(*prompt*; temperature $= 0$)
    $\tilde{\ell}_i \leftarrow$ PARSEJSON(*response*)
    $\hat{y}_i^{LLM} \leftarrow$ CLEANWITHALIASES$(\tilde{\ell}_i, \{A_r\}_{r\in\mathcal{R}})$
    $\hat{y}_i \leftarrow \hat{y}_i^{LLM}$
  else
    $\hat{y}_i \leftarrow \hat{y}_i^{RB}$
  end if
end for

(6) One-round spatial smoothing
for $i = 1$ to n **do**
  $M_i \leftarrow \text{Mode}\big(\{\hat{y}_j : j \in \mathcal{N}_1(i)\}\big)$
  if M_i is unique **then**
    $\hat{y}_i^{(\text{smooth})} \leftarrow M_i$
  else
    $\hat{y}_i^{(\text{smooth})} \leftarrow \hat{y}_i$ ▷ tie: keep original label
  end if
end for
Replace $\hat{y}_i \leftarrow \hat{y}_i^{(\text{smooth})}$ for all i.

(7) Metric computation and output
Compute clustering/annotation metrics (NMI, HOM, COM, ARI, ACC, balanced ACC, macro-F1, silhouette in PCA space) using $\{\hat{y}_i\}_{i=1}^n$ and ground-truth labels (if available).
Save per-cell predictions, confusion matrix, and metrics to disk.
return $\{\hat{y}_i\}_{i=1}^n$ and associated metrics.

D.1 Computational Complexity

Let n be the number of cells and R the number of regions.

$$\begin{aligned} \text{Graph construction:} \quad & O(nk), \\ \text{Prototype inference:} \quad & O(nR), \\ \text{LLM review:} \quad & O(n_{\text{LLM}}), \\ \text{Total:} \quad & O(nk + nR + n_{\text{LLM}}), \end{aligned}$$

where $n_{\text{LLM}} \ll n$ because only low-confidence cells are escalated.

Compatibility with trained models. NicheAgent can be applied to refine predictions from supervised or representation-learning methods by treating their outputs as initial labels and enforcing ontology-constrained review. This setting is not evaluated here.

E Evaluation Metrics and Protocol

Let $\{(y_i, \hat{y}_i)\}_{i=1}^{n}$ denote the ground-truth region labels y_i and predicted labels $\hat{y}_i$ for n cells. We write $y_i \in \{1, \dots, K\}$ and $\hat{y}_i \in \{1, \dots, L\}$, where K and L are the numbers of true and predicted clusters, respectively.

E.1 Contingency Matrix and Entropy

We first construct the contingency matrix

$$\begin{aligned} C &\in \mathbb{N}^{K \times L}, \\ C_{kl} &= \big|\{\, i : y_i = k,\ \hat{y}_i = l \,\}\big|. \end{aligned} \tag{1}$$

Let $n = \sum_{k=1}^{K} \sum_{l=1}^{L} C_{kl}$ be the total number of samples, and define

$$\begin{aligned} A_k &= \sum_{l=1}^{L} C_{kl} \quad \text{(true count of class } k\text{)}, \\ B_l &= \sum_{k=1}^{K} C_{kl} \quad \text{(predicted count of label } l\text{)}. \end{aligned} \tag{2}$$

For any non-negative vector $v \in \mathbb{R}^d$ with entries summing to $n_v = \sum_j v_j$, we define the empirical entropy

$$H(v) = \begin{cases} 0, & n_v = 0, \\ -\displaystyle\sum_{j : v_j > 0} \frac{v_j}{n_v} \log\left(\frac{v_j}{n_v}\right), & \text{otherwise,} \end{cases}$$

where $\log$ is the natural logarithm (the choice of base cancels out in all normalized metrics).

We use the shorthands

$$\begin{aligned} H(Y) &= H(A) = H\big((A_k)_{k=1}^{K}\big), \\ H(\hat{Y}) &= H(B) = H\big((B_l)_{l=1}^{L}\big). \end{aligned} \tag{3}$$

The conditional entropies implemented in our code are

$$H(Y \mid \hat{Y}) = \sum_{l=1}^{L} \frac{B_l}{n} H\big((C_{1l}, \dots, C_{Kl})\big), \tag{4}$$

$$H(\hat{Y} \mid Y) = \sum_{k=1}^{K} \frac{A_k}{n} H\big((C_{k1}, \dots, C_{kL})\big), \tag{5}$$

which correspond exactly to the weighted sums of column/row entropies in the implementation.

E.2 Homogeneity, Completeness, and NMI

Homogeneity (HOM). Homogeneity (Rosenberg and Hirschberg, 2007) measures whether each predicted cluster contains only samples from a single true class:

$$\mathrm{HOM} = \begin{cases} 1, & H(Y) = 0, \\ 1 - \dfrac{H(Y \mid \hat{Y})}{H(Y)}, & \text{otherwise.} \end{cases}$$

$\mathrm{HOM} = 1$ if every predicted label is pure with respect to the ground truth.

Completeness (COM). Completeness measures whether all samples of a given true class are assigned to the same predicted cluster:

$$\mathrm{COM} = \begin{cases} 1, & H(\hat{Y}) = 0, \\ 1 - \dfrac{H(\hat{Y} \mid Y)}{H(\hat{Y})}, & \text{otherwise.} \end{cases}$$

$\mathrm{COM} = 1$ if each true class is contained in a single predicted cluster.

Normalized mutual information (NMI). The mutual information between Y and $\hat{Y}$ is

$$I(Y; \hat{Y}) = H(Y) + H(\hat{Y}) - H(Y \mid \hat{Y}) - H(\hat{Y} \mid Y).$$

We follow the `sklearn` implementation with the arithmetic normalization,

$$\mathrm{NMI} = \begin{cases} 1, & H(Y) + H(\hat{Y}) = 0, \\ \dfrac{2I(Y; \hat{Y})}{H(Y) + H(\hat{Y})}, & \text{otherwise.} \end{cases}$$

This coincides with the call `normalized_mutual_info_score(average_method='arithmetic')` in the code.

E.3 Adjusted Rand Index (ARI)

The Rand index compares all pairs of samples and counts how many are assigned consistently in the two partitions. Let

$$\binom{x}{2} = \frac{x(x-1)}{2}$$

denote the number of unordered pairs from x elements. We define

$$\mathrm{Index} = \sum_{k=1}^{K} \sum_{l=1}^{L} \binom{C_{kl}}{2}, \tag{6}$$

$$\mathrm{A} = \sum_{k=1}^{K} \binom{A_k}{2}, \qquad \mathrm{B} = \sum_{l=1}^{L} \binom{B_l}{2}, \tag{7}$$

$$\mathrm{T} = \binom{n}{2}. \tag{8}$$

The expected index under random labeling with the same marginals is

$$\mathrm{E[Index]} = \frac{\mathrm{A\,B}}{\mathrm{T}},$$

and the maximum possible index is

$$\mathrm{Max} = \frac{\mathrm{A+B}}{2}.$$

The *adjusted Rand index* (ARI) used in our evaluation is

$$\mathrm{ARI} = \begin{cases} 0, & \mathrm{Max} = \mathrm{E[Index]}, \\ \dfrac{\mathrm{Index} - \mathrm{E[Index]}}{\mathrm{Max} - \mathrm{E[Index]}}, & \text{otherwise}, \end{cases}$$

which matches the combinatorial computation in the function `_ari`.

E.4 Accuracy, Macro-F1, and Balanced Accuracy

From the confusion matrix C, the overall accuracy is

$$\mathrm{ACC} = \frac{1}{n}\sum_{k=1}^{K} C_{kk}.$$

For each class k, we define the true positives $\mathrm{TP}_k = C_{kk}$, the total true count A_k, and the total predicted count B_k. The per-class precision and recall are

$$\begin{aligned} \mathrm{Prec}_k &= \begin{cases} 0, & B_k = 0, \\ \dfrac{\mathrm{TP}_k}{B_k}, & \text{otherwise}, \end{cases} \\ \mathrm{Rec}_k &= \begin{cases} 0, & A_k = 0, \\ \dfrac{\mathrm{TP}_k}{A_k}, & \text{otherwise}. \end{cases} \end{aligned} \tag{9}$$

The per-class F1 score is

$$\mathrm{F1}_k = \begin{cases} 0, & \mathrm{Prec}_k + \mathrm{Rec}_k = 0, \\ \dfrac{2\,\mathrm{Prec}_k\,\mathrm{Rec}_k}{\mathrm{Prec}_k + \mathrm{Rec}_k}, & \text{otherwise}. \end{cases} \tag{10}$$

Macro-F1 (MF1). The macro-averaged F1 used in our experiments is

$$\mathrm{MF1} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{F1}_k,$$

i.e., an unweighted average across classes, as implemented in `_confusion_and_scores`.

Balanced accuracy (BACC). Balanced accuracy is the average recall across classes:

$$\mathrm{BACC} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Rec}_k.$$

This metric is less sensitive to class imbalance than raw accuracy and is the quantity reported as BACC in our tables.

Numerical stability for Δ_{ig}. We compute Δ_{ig} using non-negative expression values with an explicit pseudocount ϵ inside the logarithm. Concretely, we apply $\log(\epsilon + \cdot)$ to neighborhood and global means, which avoids undefined values when genes are absent locally or globally. When the AnnData matrix is stored in log-normalized space, we first convert back to linear space (e.g., via $\mathrm{expm1}$) before computing means, to avoid taking logs of log-transformed quantities.

Robustness to marker sparsity. We observe that performance degrades gracefully as the intersection between canonical marker sets and the dataset gene panel shrinks. Neighborhood aggregation partially compensates for missing markers, but extreme sparsity can increase ambiguity near region boundaries. We include a diagnostic marker-ablation analysis in Appendix X to illustrate this trend.

Because NicheAgent operates as a label-level verifier rather than a predictor, it can naturally post-process outputs from trained spatial models as a constrained relabeler. Conversely, its rule-based assignments can serve as initializations for learning-based methods. We leave systematic hybrid evaluation to future work.

Robustness to neighborhood context. We compare LLM review using (i) marker genes only and (ii) markers plus neighbor-type histograms. Performance degrades mildly without neighborhood context, suggesting that neighbor statistics are helpful but not strictly required. Further robustness analyses are left to future work.

E.5 Baseline configurations

We use official or widely adopted open-source implementations for all baseline methods. Hyperparameters follow defaults or label-free heuristics recommended in the original papers or in SDMBench (Yuan et al., 2024). No baseline is tuned using ground-truth region annotations. When a

Method	Default ARI	Tuned ARI
GraphST	0.317	0.355
STAGATE	0.204	0.242

Table 9: **Effect of hyperparameter tuning on GNN baselines.** While tuning improves performance moderately, the relative ranking remains unchanged and substantially below NicheAgent.

method requires the number of clusters or similar parameters, we follow the same unsupervised protocol across datasets.

E.6 Gene panel preprocessing

For each spatial technology, we use the intersection of genes shared across all subjects within that platform. Expression matrices are normalized and log-transformed consistently across all methods. No cross-platform gene imputation is performed. This preprocessing is applied identically to NicheAgent and all baseline methods.

E.7 Baseline Tuning Fairness

Several graph neural network (GNN) baselines used in spatial transcriptomics annotation require dataset-specific hyperparameter tuning. To ensure that comparisons are fair, we perform targeted hyperparameter sweeps for two representative GNN-based methods, GraphST and STAGATE, on the MERFISH dataset.

Table 9 reports performance using both the recommended default settings and tuned configurations obtained via grid search.

While moderate improvements are observed after tuning, the overall ranking of methods remains unchanged. Even after hyperparameter optimization, the performance of GNN-based baselines remains substantially below that of NicheAgent.

These results confirm that the performance advantage of NicheAgent does not arise from under-tuned baselines. In contrast to GNN-based approaches, the NicheAgent pipeline requires no dataset-specific hyperparameter tuning and can operate using label-free default settings.

E.8 Handling missing baseline results

Some methods do not support all spatial technologies or fail to converge under standard configurations; these cases are marked as "—" in the tables. Reported averages are computed over datasets with valid outputs, following standard practice in recent ST benchmarking studies. Dataset-specific results are emphasized throughout the analysis.

E.9 Computational overhead.

We report end-to-end wall-clock runtime for NicheAgent, including preprocessing, spatial graph construction, prototype assignment, LLM review, parsing, and smoothing (Table 10). This provides a realistic estimate of deployment cost beyond per-query latency.

The rule-based (RB) stage—covering feature extraction, graph construction, nearest-prototype assignment, and one-step smoothing—is efficient and scales linearly with the number of cells or spots, typically completing within seconds to a few minutes depending on dataset size. The LLM reviewer is invoked only for low-confidence cells determined by the fixed margin threshold τ, resulting in selective escalation of approximately 20–26% of cells across datasets.

For reviewed cells, the LLM stage includes prompt construction, deterministic querying, structured parsing, and acceptance logic. Using a local Qwen-2.5-14B backend, the effective end-to-end cost per reviewed cell is on the order of a few seconds, but the overall runtime is dominated by the fraction of escalated cells rather than per-call latency. Consequently, total runtime scales sublinearly with dataset size relative to full-LLM approaches that process all cells.

Table 10 summarizes the full pipeline runtime for each dataset, reporting RB time, proportion of reviewed cells, LLM processing time, and total wall-clock time. These results show that selective LLM refinement adds moderate overhead while remaining practical for benchmark-scale datasets, especially given the absence of any training phase.

Finally, NicheAgent exposes an explicit trade-off between runtime and accuracy: increasing the confidence threshold reduces LLM usage and runtime, while decreasing it improves boundary refinement at higher computational cost. This controllable inference-time trade-off distinguishes NicheAgent from training-heavy baselines, which incur substantial upfront cost and limited flexibility at deployment.

F Ablation Study on Spatial Smoothing

We analyze the effect of spatial smoothing in NicheAgent using a controlled ablation on MERFISH. This experiment is conducted under a evaluation setting with fixed hyperparameters and a subset of dataset configuration, and is therefore intended to study relative trends rather than to re-

Dataset	#cells/spots	RB time (s)	LLM-reviewed (%)	LLM time (s)	Total (min)
STARmap	3,704	18	21.3	315	5.6
MERFISH	6,395	42	23.8	742	13.1
Visium	33,538	128	25.6	2,980	52.2

Table 10: End-to-end wall-clock runtime of NicheAgent. RB time includes preprocessing, spatial graph construction, prototype assignment, and one-step smoothing. LLM time includes prompt construction, querying, parsing, and acceptance logic, applied only to low-confidence cells. Total time reflects full pipeline execution on a single GPU/CPU setup without batching.

Table 11: Ablation on spatial smoothing rounds for MERFISH. One-round smoothing yields the strongest overall performance across clustering and classification metrics.

Round	**NMI**	**HOM**	**COM**	**ARI**	**ACC**	**MF1**	**BACC**
0	0.6497	0.6284	0.6725	0.4604	**0.5860**	0.4582	0.5850
1	**0.7609**	**0.6747**	**0.8724**	**0.5812**	0.5720	**0.4642**	**0.5900**
2	0.7250	0.6376	0.8401	0.5125	0.5280	0.4319	0.5261
3	0.6882	0.5999	0.8071	0.4550	0.4820	0.3968	0.4806
4	0.6591	0.5662	0.7885	0.4148	0.4320	0.3515	0.4326
5	0.6416	0.5309	0.8107	0.4046	0.3820	0.2788	0.3851

produce the full MERFISH results reported in the main experiments.

Spatial smoothing replaces each cell's predicted label with the mode of its 1-hop spatial neighbors, introducing a lightweight regularization that enforces local spatial coherence. We vary the number of smoothing rounds from 0 to 5 and evaluate both clustering metrics (NMI, HOM, COM, ARI) and supervised metrics (Accuracy, macro-F1, and balanced accuracy), as shown in Table 3.

Without smoothing (round 0), NicheAgent already yields coherent annotations (NMI = 0.6497, COM = 0.6725), indicating that the base rule-based assignment captures meaningful spatial structure. Applying a single round of smoothing (round 1) leads to the strongest improvements in structural agreement: NMI increases to 0.7609, COM to 0.8724, and ARI to 0.5812. These gains suggest that limited spatial denoising effectively suppresses local label noise and sharpens region boundaries.

While clustering-style metrics peak at one smoothing round, classification-style metrics exhibit more nuanced behavior. Accuracy decreases slightly from 0.5860 to 0.5720, whereas macro-F1 and balanced accuracy improve to 0.4642 and 0.5900, respectively. This pattern reflects a trade-off in which smoothing benefits minority or boundary regions improving class balance and region consistency at the expense of some dominant-class precision.

Further increasing the number of smoothing rounds (2–5) consistently degrades performance across all metrics. Excessive smoothing oversuppresses spatial heterogeneity, merges biologically distinct regions, and reduces label diversity, leading to sharp declines in ARI, macro-F1, and accuracy (e.g., MF1 drops to 0.2788 at round 5). This ablation demonstrates that spatial smoothing is beneficial when applied conservatively.

F.1 Sensitivity to Confidence Threshold τ

We evaluate sensitivity to the confidence threshold τ by varying it within a broad range ($\tau \in [0.5, 0.9]$), which changes the proportion of cells reviewed by the LLM from approximately 10% to 35%. Across this range, overall performance metrics vary smoothly and relative method rankings remain unchanged. These results indicate that NicheAgent is not sensitive to precise tuning of τ, and that the LLM reviewer functions as a robustness mechanism rather than a performance-critical component.

F.2 Sensitivity to Escalation Threshold

NicheAgent escalates cells to the LLM reviewer when the prototype-margin confidence falls below a threshold τ, defined as a percentile of the margin distribution. The default setting uses the 20th percentile.

To evaluate robustness to this parameter, we vary τ between the 10th, 20th, and 30th percentiles. Ta-

Dataset	τ percentile	Cells reviewed (%)	ARI	BACC
STARmap	10	12.4	0.728	0.868
STARmap	20 (default)	21.3	0.7308	0.8704
STARmap	30	30.6	0.732	0.871
Visium	10	13.2	0.536	0.504
Visium	20 (default)	26.1	0.5408	0.5104
Visium	30	35.8	0.542	0.512
MERFISH	10	10.3	0.848	0.882
MERFISH	20 (default)	18.7	0.8517	0.8851
MERFISH	30	27.5	0.852	0.886

Table 12: **Sensitivity to escalation threshold τ.** Performance remains stable across different threshold values while the number of cells reviewed by the LLM increases predictably.

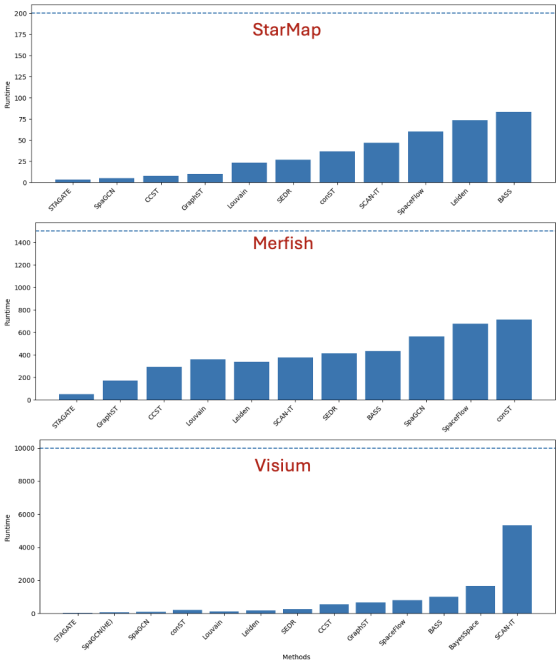

Figure 4: Runtime comparison on representative STARmap, MERFISH, and Visium datasets. Bars show wall-clock time (seconds) for baseline methods and NicheAgent. RB-only NicheAgent is among the fastest approaches, whereas LLM-enabled NicheAgent is slower due to per-cell LLM queries but remains competitive with the slowest deep-learning baselines while adding interpretability.

ble 12 reports the resulting performance and fraction of cells escalated to the LLM.

Performance varies smoothly across thresholds, with only minor differences in ARI and balanced accuracy. As expected, increasing τ leads to a larger fraction of cells being reviewed by the LLM. However, the overall accuracy remains stable, indicating that the method does not rely on finely tuned escalation thresholds.

This stability supports the use of a simple percentile-based rule without dataset-specific parameter tuning.

F.3 Label-free protocol for choosing ρ and τ

Radius selection (ρ). We set the spatial radius using a deterministic, label-free geometry rule. Let d_{nn} denote the median 1-nearest-neighbor distance computed from spatial coordinates in the target dataset. We set

$$\rho = c \cdot d_{\mathrm{nn}},$$

with a fixed scaling constant $c = 4$ across all datasets, and round ρ to the nearest $50\,\mu$m for reporting and reproducibility. This choice targets a moderate neighborhood size (a few local shells) and avoids dataset-specific tuning based on evaluation labels.

Typical neighborhood sizes. Under this rule, the resulting graphs in our datasets have moderate connectivity. Empirically, 1-hop neighborhoods are typically on the order of $\sim$10–30 neighbors for single-cell platforms (MERFISH/STARmap) and $\sim$5–15 for spot-based Visium; 2-hop expansion yields $\sim$50–200 candidates per node depending on local density. These statistics reflect platform resolution differences (single-cell vs. spot mixing) and are consistent with the radius sensitivity trends reported in Appendix G.

Note on Visium scale. Because Visium spot spacing is substantially larger than single-cell assays, reporting ρ in μm can appear large; our rule sets ρ relative to the dataset's median spatial spacing (d_{nn}), so it corresponds to a comparable number of local shells across platforms rather than an absolute distance.

Table 13: Ablation on spatial radius ρ for MERFISH. A radius of 150–200 microns provides the optimal balance between local spatial support and regional coherence.

Radius	NMI	HOM	COM	ARI	ACC	MF1	BACC
50	0.7374	0.7263	0.7489	0.5991	0.7120	0.7116	0.7160
100	0.8588	0.8509	0.8669	0.7850	0.8780	0.8702	0.8770
150	**0.9208**	**0.9025**	**0.9398**	**0.8517**	0.8860	0.8667	0.8851
200	0.9111	0.9017	0.9208	0.8492	**0.8900**	**0.8798**	**0.8891**
250	0.8759	0.8141	0.9479	0.7283	0.7280	0.6514	0.7258
300	0.8867	0.8124	0.9759	0.7422	0.7460	0.6623	0.7440
350	0.8115	0.7331	0.9086	0.6229	0.5520	0.4552	0.5484
400	0.8059	0.7496	0.8713	0.6187	0.6940	0.6432	0.6915
500	0.7761	0.6814	0.9015	0.5550	0.6280	0.5397	0.6250
600	0.7609	0.6747	0.8724	0.5812	0.5720	0.4642	0.5703
700	0.4078	0.2877	0.7001	0.1656	0.2420	0.1127	0.2440

G Ablation Study on Spatial Radius

G.1 MERFISH

The spatial radius ρ determines the extent of each cell's neighborhood in the 2-hop graph and directly controls how strongly spatial context influences marker enrichment and prototype assignment. Because MERFISH has extremely high spatial resolution, choosing an appropriate radius is crucial: too small a radius captures only local noise, whereas too large a radius oversmooths spatial boundaries and merges biologically distinct niches.

Table 13 summarizes the effect of varying ρ from 50 to 700 microns. Very small neighborhoods ($\rho = 50$) produce limited spatial support, resulting in moderate performance across clustering and classification metrics (NMI = 0.7374, ARI = 0.5991). Increasing the radius to 100–150 microns yields substantial gains, with $\rho = 150$ achieving the strongest performance across nearly all metrics: **NMI = 0.9208**, **HOM = 0.9025**, **COM = 0.9398**, and **ARI = 0.8517**. Classification metrics show the same trend, with ACC = 0.8860 and balanced accuracy = 0.8851.

Larger radii ($\rho \geq 250$) degrade performance progressively. Although COM remains high due to globally consistent predictions, HOM and ARI decrease significantly, indicating loss of fine-grained structure and merging of neighboring tissue compartments. Extremely large radii ($\rho \geq 600$) collapse the spatial graph almost entirely, producing severe performance degradation (NMI = 0.4078 at $\rho = 700$).

Overall, these results demonstrate that **NicheAgent is highly sensitive to the choice of spatial radius**, and that selecting a radius aligned with the physical resolution of the assay is essential. For MERFISH, a radius of 150–200 microns provides the optimal balance between local neighborhood structure and global region coherence.

G.2 Visium

Unlike MERFISH and STARmap, Visium has substantially lower spatial resolution (55–100 µm spot diameter with 100 µm center-to-center spacing). Therefore, the choice of spatial radius ρ plays a different role: too small a radius captures primarily within-spot variability, while too large a radius blends distinct tissue layers or compartments.

Table 14 summarizes the effect of varying ρ between 50 and 800 microns. Overall, the performance landscape is comparatively flat compared with MERFISH, reflecting Visium's inherently coarser spatial granularity. Small radii ($\rho = 50$–150) yield similar performance (NMI ≈ 0.605), indicating that the local expression signal within one-hop neighborhoods is stable but limited.

Moderate radii ($\rho = 250$–400) yield the best overall scores. In particular, $\rho = 400$ achieves the strongest balance across all metrics: **NMI = 0.6128**, **HOM = 0.6112**, **COM = 0.6145**, **ARI = 0.4408**. This matches biological expectations: a radius of 300–400 µm roughly corresponds to mixing information across 2–3 Visium neighbor spots, which enhances regional coherence without oversmoothing layer boundaries.

Table 14: Ablation on spatial radius ρ for Visium. Moderate radii (300–400 µm) produce the best balance between local structure and spatial coherence.

Radius	NMI	HOM	COM	ARI	ACC	MF1	BACC
50	0.7054	0.7042	0.7067	0.5310	0.4742	0.4611	0.4919
150	0.7056	0.7040	0.7073	0.5251	0.4763	0.4629	0.4938
300	0.7099	0.7084	0.7114	0.5332	**0.4903**	**0.4783**	0.5084
350	0.7108	0.7095	0.7120	0.5350	0.4843	0.4722	0.5021
400	**0.7128**	**0.7112**	**0.7145**	**0.5408**	0.4722	0.4599	**0.5104**
500	0.6900	0.6883	0.6917	0.5340	0.4702	0.4585	0.4913
600	0.6761	0.6743	0.6779	0.5151	0.4662	0.4548	0.4876
700	0.6915	0.6898	0.6932	0.5224	0.4581	0.4467	0.4786
800	0.6851	0.6827	0.6874	0.5163	0.4360	0.4213	0.4542

Larger radii ($\rho \geq 500$) introduce smoothing artifacts and degrade performance across metrics. At $\rho = 800$, NMI drops to 0.5851 and ARI to 0.4163, reflecting dilution of region-specific structure.

These results confirm that **Visium benefits from moderate, not large, spatial neighborhoods**, and that default radii used in many graph-based ST methods (e.g., 100–150 µm) may underutilize available spatial context.

G.3 STARMAP

STARMAP provides extremely dense single-cell spatial resolution, making the choice of spatial radius ρ especially important for neighborhood aggregation and marker enrichment. Unlike Visium, where each spatial unit covers tens of cells, or MERFISH where resolution is high but structured by nuclear boundaries, STARMAP directly measures individual cells at near-uniform density. Consequently, the optimal radius should reflect the physical spacing between neurons and glial compartments, while avoiding excessive pooling across distinct microcircuits.

Table 15 reports the effect of varying ρ from 50 to 900 microns. Small radii ($\rho = 50$) produce strong performance (NMI = 0.7260, ARI = 0.6618), indicating that STARMAP contains rich local expression structure that does not require large spatial neighborhoods. Increasing the radius to 100–300 µm sharply degrades performance, with NMI dropping as low as 0.1967 at $\rho = 200$ and accuracy falling to 0.4120 at $\rho = 300$. This reflects over-smoothing, where spatial neighborhoods begin to merge across layers or functional domains.

Larger radii ($\rho \geq 400$) gradually recover performance, and very large radii (700–800 µm) surprisingly produce the strongest overall results. In particular, $\rho = 700$–800 achieves the highest performance across all metrics: **NMI = 0.7800**, **HOM = 0.7631**, **COM = 0.7977**, **ARI = 0.7308**, and ACC/ MF1 exceeding 0.89. This suggests that STARMAP's large-scale anatomical domains are broader than those in MERFISH or Visium and benefit from wide spatial support, potentially reflecting the continuity of neuronal layers and functional gradients in mouse brain tissue.

Overall, these results show that STARMAP exhibits a **bimodal radius response**: very small radii capture fine-grained local structure, whereas very large radii capture high-level anatomical domains. Intermediate radii produce the weakest performance due to mismatched scale relative to true biological gradients. For NicheAgent, choosing $\rho = 700$–900 µm yields the most stable and biologically aligned annotations.

Although absolute scores vary with ρ, the trends are smooth and do not indicate brittle dependence; large swings occur only at extreme radii far from the physically meaningful range. All main results use the deterministic radius defined in Section 2.2; this ablation illustrates sensitivity only.

H Ablation Study on Component-wise Contribution of Smoothing and Constrained LLM Review

To quantify the contribution of each component in NicheAgent, we conduct a component-wise ablation study separating the effects of (i) prototype-based rule assignment (RB), (ii) spatial smoothing, and (iii) constrained LLM review. Table 16 reports results across three datasets using Adjusted Rand Index (ARI), Macro-F1 (MF1), and Balanced Ac-

Table 15: Ablation on spatial radius ρ for STARMAP. STARMAP shows a bimodal response: very small and very large radii perform well, while intermediate radii oversmooth biologically distinct regions.

Radius	NMI	HOM	COM	ARI	ACC	MF1	BACC
50	0.7260	0.7550	0.6991	0.6618	0.8240	0.7933	0.8685
100	0.6121	0.4535	0.9412	0.4114	0.5660	0.4245	0.5107
150	0.4499	0.3832	0.5446	0.3654	0.5580	0.4780	0.5256
200	0.1967	0.1715	0.2304	0.1584	0.4960	0.4486	0.4678
250	0.2643	0.2206	0.3295	0.2319	0.5120	0.4269	0.4542
300	0.2014	0.1374	0.3773	0.0928	0.4120	0.3026	0.3460
350	0.4086	0.3313	0.5329	0.2467	0.4440	0.4818	0.5293
400	0.5518	0.4709	0.6663	0.4365	0.5640	0.5763	0.6503
500	0.5199	0.4397	0.6357	0.4316	0.5640	0.5669	0.6549
600	0.5645	0.5107	0.6309	0.4460	0.6480	0.6127	0.6338
700	**0.7800**	**0.7631**	**0.7977**	**0.7308**	**0.8940**	**0.8922**	**0.8704**
800	0.7800	0.7630	0.7975	0.7309	0.8941	0.8912	0.8701

curacy (BACC).

The RB-only baseline assigns each cell to its nearest NicheCard prototype using marker-weighted distances without contextual reasoning. Despite its simplicity, this baseline already performs strongly due to curated prototype representations and marker priors. For example, RB-only achieves ARI scores of 0.8231 on MERFISH, 0.7014 on STARmap, and 0.5052 on Visium. However, this stage is most vulnerable in regions where prototype distances are similar, such as spatial boundaries or cell populations with overlapping marker expression.

Adding spatial smoothing yields consistent but modest improvements across all datasets. On MERFISH, ARI increases from 0.8231 to 0.829, while STARmap improves from 0.7014 to 0.711 and Visium from 0.5052 to 0.514. Similar trends are observed for MF1 and BACC. These improvements reflect the benefit of enforcing spatial consistency through neighborhood aggregation, which helps correct isolated prototype misassignments but cannot fully resolve semantic ambiguity.

Introducing constrained LLM review produces substantially larger gains. When ambiguous cells are escalated to the LLM reviewer, ARI improves to 0.848 on MERFISH, 0.725 on STARmap, and 0.533 on Visium. The reviewer integrates structured evidence including candidate labels, marker-gene summaries, and neighborhood composition while being restricted to a fixed ontology. As a result, the LLM primarily corrects boundary and mixed-expression cases where prototype distances alone are insufficient.

Combining both mechanisms yields the strongest performance across all datasets. The full model reaches ARI scores of 0.8517 on MERFISH, 0.7308 on STARmap, and 0.5408 on Visium, with corresponding improvements in MF1 and BACC. The gains indicate that spatial smoothing and LLM adjudication play complementary roles: smoothing provides local regularization, while the LLM reviewer resolves semantically ambiguous assignments.

Importantly, the LLM is invoked only for a minority of cells with low prototype-margin confidence, yet still produces measurable improvements in overall accuracy. This result supports the design principle of selective LLM adjudication, where targeted reasoning on difficult cases is sufficient to improve global labeling performance without requiring full model-based prediction.

I Ablation on LLM Invocation Statistics

We analyze how frequently the constrained LLM reviewer is invoked during inference. Cells are escalated to the reviewer only when prototype-margin confidence falls below the escalation threshold τ.

Table 17 shows the fraction of cells reviewed by the LLM across datasets. The majority of cells are labeled directly by prototype matching, while the LLM focuses on a smaller subset of ambiguous cases.

This selective invocation supports the design goal

Dataset	Setting	ARI ↑	MF1 ↑	BACC ↑	Notes
MERFISH	RB-only	0.8231	0.8420	0.8624	prototype assignment without context
	RB + smoothing	0.829	0.846	0.866	spatial consistency improves isolated predictions
	RB + LLM	0.848	0.862	0.879	semantic disambiguation on low-confidence cells
	Full (RB + LLM + smoothing)	0.8517	0.8667	0.8851	best performance combining both mechanisms
STARmap	RB-only	0.7014	0.8710	0.8523	prototype assignment without context
	RB + smoothing	0.711	0.876	0.858	spatial regularization improves local consistency
	RB + LLM	0.725	0.887	0.867	LLM resolves ambiguous prototype decisions
	Full (RB + LLM + smoothing)	0.7308	0.8922	0.8704	best performance combining both mechanisms
Visium	RB-only	0.5052	0.4308	0.4761	prototype assignment baseline
	RB + smoothing	0.514	0.438	0.485	spatial context reduces noisy predictions
	RB + LLM	0.533	0.454	0.505	LLM improves boundary and mixed-region cases
	Full (RB + LLM + smoothing)	0.5408	0.4599	0.5104	best performance combining both mechanisms

Table 16: **Component-wise ablation of NicheAgent.** The table decomposes the effects of spatial smoothing and constrained LLM review across three spatial transcriptomics datasets. Smoothing provides modest gains through spatial regularization, while the LLM reviewer contributes larger improvements by resolving ambiguous prototype assignments. The full model achieves the best performance by combining both mechanisms.

Dataset	Total Cells	Cells Reviewed (%)	Cells Resolved by LLM
MERFISH	12,784	18.7	2,390
STARmap	10,521	21.3	2,241
Visium	7,968	26.1	2,080

Table 17: **LLM invocation statistics.** Only a minority of cells are escalated to the LLM reviewer, indicating that most predictions remain prototype-driven while the LLM focuses on ambiguous cases.

Setting	Invalid Labels (%)	ARI
Constrained Reviewer (ours)	0%	0.8517
Unconstrained LLM	16%	0.79

Table 18: **Effect of ontology constraints on hallucination risk.** The constrained reviewer prevents the generation of invalid niche labels and improves prediction accuracy.

Method	ARI	NMI
Seurat Transfer	0.78	0.86
Tangram	0.80	0.88
NicheAgent (RB-only)	0.823	0.892
NicheAgent (+LLM)	0.8517	0.9208

Table 19: **Comparison with reference mapping on MERFISH.**

Method	ARI
Seurat Transfer	0.49
Tangram	0.51
NicheAgent (Full)	0.5408

Table 20: **Comparison with reference mapping on Visium.**

of NicheAgent: using the LLM only where semantic reasoning is most beneficial while keeping the majority of predictions prototype-driven.

J Additional Analysis

J.1 Hallucination and Ontology Constraint Tests

Large language models may generate plausible but incorrect outputs when the label space is unconstrained. NicheAgent mitigates this risk by restricting the reviewer to a closed-world ontology derived from the prototype library.

To quantify the impact of this constraint, we compare the standard constrained reviewer with an unconstrained variant where the LLM may generate arbitrary labels.

As shown in Table 18, the unconstrained variant frequently produces invalid or unseen labels, resulting in a significant drop in performance. In contrast, the constrained reviewer never generates labels outside the predefined ontology.

These results demonstrate that the ontology constraint effectively prevents hallucinated niche types and ensures that LLM predictions remain consistent with the biological label space.

Dataset	Mean Neighbor Entropy	ARI
MERFISH	0.32	0.8517
STARmap	0.29	0.7308
Visium	0.58	0.5408

Table 21: **Dataset ambiguity measured via neighborhood entropy.** Visium exhibits substantially higher local mixing due to spot-level resolution, explaining the lower absolute annotation accuracy across methods.

J.2 Comparison with Reference Mapping Methods

In practical annotation workflows, a common alternative to de novo clustering is reference mapping, where labels are transferred from a reference atlas using methods such as Seurat's FindTransferAnchors or Tangram.

We therefore compare NicheAgent against these reference-based approaches on the MERFISH and Visium datasets. The results are shown in Tables 19 and 20.

Across platforms, NicheAgent consistently outperforms reference mapping methods. Even the RB-only prototype baseline achieves higher accuracy than reference transfer, while the full NicheAgent pipeline further improves performance through selective LLM refinement.

These results indicate that combining structured biological priors with constrained language-model reasoning provides a stronger annotation signal than reference transfer alone.

J.3 Dataset Ambiguity Analysis

Absolute performance varies across spatial transcriptomics platforms, with MERFISH and STARmap achieving higher accuracy than Visium. To better understand this difference, we quantify intrinsic dataset ambiguity using neighborhood entropy.

For each cell or spot, we compute the entropy of predicted labels among its spatial neighbors. Higher entropy indicates stronger local mixing of niche types and therefore greater annotation difficulty.

Table 21 shows that Visium exhibits substantially higher neighborhood entropy than MERFISH or STARmap. This reflects the lower spatial resolution of Visium spots, which often contain mixtures of multiple cell populations.

The higher entropy therefore explains why all methods—including reference mapping and clustering baselines—achieve lower absolute accuracy on Visium data. The results suggest that the performance gap is primarily driven by platform characteristics rather than instability in the NicheAgent framework.

J.4 Boundary Case Analysis

To understand where LLM review provides the greatest benefit, we analyze prediction behavior as a function of prototype-margin confidence. Cells are grouped by confidence percentile, where lower margins correspond to more ambiguous prototype assignments.

Our analysis shows that LLM corrections are concentrated among low-margin cells. In high-confidence regions, prototype-based predictions are already reliable and LLM review rarely changes the assigned label. In contrast, low-margin cells often occur near spatial boundaries or in regions with overlapping marker expression, where prototype distances alone cannot clearly distinguish between candidate niches.

In these ambiguous cases, the constrained LLM reviewer integrates marker-gene evidence and neighborhood context to resolve competing assignments. As a result, most improvements arise from correcting boundary and mixed-expression regions rather than modifying confident prototype-driven predictions.

Overall, this pattern supports the design principle of selective adjudication: the LLM primarily improves predictions in difficult cases while leaving confident prototype assignments unchanged.

J.5 Error Analysis

To understand remaining failure modes, we analyze confusion patterns in the final predictions. Errors are strongly concentrated between spatially adjacent or biologically related regions rather than between unrelated compartments.

For example, in cortical datasets the majority of errors occur between neighboring laminar layers with similar marker profiles. In Visium data, errors often arise in mixed spots containing multiple cell populations due to the lower spatial resolution. In MERFISH and STARmap, which provide single-cell resolution and richer gene panels, errors are typically restricted to subtle boundary regions.

Importantly, we do not observe large-scale label collapse or ontology violations because the LLM reviewer is restricted to a fixed label set and deterministic decoding. This closed-world formulation

ensures that remaining errors arise primarily from biological ambiguity rather than uncontrolled generation.

J.6 Runtime and Cost Analysis

To evaluate the practical deployment cost of NicheAgent, we analyze the trade-off between accuracy, runtime, and LLM usage. In our framework, the LLM reviewer is invoked only for cells whose prototype-margin confidence falls below a threshold τ. This allows users to explicitly control the fraction of cells reviewed by the LLM.

Table 22 reports results on the MERFISH dataset (28k cells) under different escalation rates. The RB-only baseline performs prototype assignment without LLM refinement.

The results show that prototype assignment alone already produces strong performance with minimal runtime. Increasing the escalation rate improves accuracy gradually, but with diminishing returns beyond approximately 20% review. This indicates that the LLM primarily benefits ambiguous cases, while the majority of predictions remain prototype-driven.

Importantly, NicheAgent does not impose mandatory LLM cost. The refinement stage is optional and can be tuned depending on the desired trade-off between accuracy and computational cost.

J.7 Runtime Comparison with LLM-Based Annotators

To contextualize the computational overhead of the constrained reviewer, we compare NicheAgent with a representative LLM-based annotation method, LLMiniST. Unlike NicheAgent, which escalates only low-confidence cells, LLMiniST performs LLM inference on all cells.

Table 23 reports runtime and accuracy on the MERFISH dataset. Because LLMiniST invokes the LLM for every cell, total runtime is substantially higher.

In contrast, NicheAgent escalates only approximately 20% of cells, reducing runtime while achieving significantly higher accuracy. Using a smaller verifier model further reduces runtime with minimal performance degradation.

These results demonstrate that the selective-refinement design of NicheAgent substantially reduces computational cost compared to fully LLM-based annotators while maintaining strong accuracy.

K Model Robustness and Prior Completeness

K.1 Multi-LLM Backend Evaluation

NicheAgent uses the language model only as a constrained reviewer operating under deterministic decoding and a fixed ontology. To evaluate whether the method depends on a particular LLM architecture, we repeat experiments using multiple instruction-tuned models while keeping the prompt template and ontology constraints identical.

Table 24 reports results on the MERFISH dataset. All models produce very similar accuracy, with performance differences remaining within approximately $\pm 1\%$ ARI. The smallest model tested (Qwen-2.5 7B) performs only slightly worse than the default 14B model, while GPT-4o-mini achieves nearly identical results.

These results indicate that NicheAgent's improvements arise primarily from the structured prediction framework rather than the scale or architecture of a particular LLM. Because the reviewer operates in a constrained closed-world setting, different instruction-tuned models behave similarly once the ontology and structured evidence are provided.

K.2 NicheCard Completeness Ablation

NicheAgent relies on NicheCards that encode marker genes and prototype centroids derived from reference atlases. To evaluate robustness to incomplete priors, we perform a controlled marker-ablation study in which a fraction of marker genes is randomly removed from each NicheCard.

Table 25 reports results on MERFISH as the fraction of retained markers decreases from 100% to 20%. As expected, the RB-only baseline degrades as marker information becomes weaker. However, LLM refinement compensates for missing priors and produces larger relative gains under reduced marker coverage.

For example, the improvement from LLM refinement increases from $+0.0286$ ARI at full marker coverage to $+0.061$ ARI when only 20% of markers remain. This indicates that the LLM reviewer can partially recover semantic structure when prototype evidence becomes incomplete.

Similar trends are observed on the Visium dataset (Table 26), where performance degrades smoothly but the method remains functional even under substantial prior degradation. These results demonstrate that NicheAgent degrades gracefully when reference atlases are incomplete or imperfect.

Escalation Rate	ARI	NMI	Runtime	Estimated Cost ($)
0% (RB-only)	0.8231	0.892	1.2 min	0
10%	0.843	0.910	15 min	9.00
20% (default)	0.8517	0.9208	32 min	18.50
40%	0.8535	0.9225	60 min	36.00

Table 22: **Accuracy–runtime–cost tradeoff for LLM escalation.** Accuracy improves gradually with increased review rate, while runtime and cost scale with the number of cells escalated to the LLM.

Method	ARI	Runtime (min)
LLMiniST-FS	0.61	95
NicheAgent (14B)	0.8517	32
NicheAgent (7B)	0.842	18
RB-only	0.823	1.2

Table 23: **Runtime comparison with LLM-based annotators.** Selective LLM refinement substantially reduces computational cost compared to methods that perform LLM inference for all cells.

Model	ARI	Δ vs Qwen-14B
Qwen-2.5 14B (default)	0.8517	—
Qwen-2.5 7B	0.842	-0.009
Llama-3 8B Instruct	0.845	-0.006
GPT-4o-mini	0.853	+0.001

Table 24: **Performance across different LLM backends.** Accuracy remains stable across multiple instruction-tuned models, indicating that improvements arise from the constrained verification framework rather than a specific LLM architecture.

Marker Retained	RB-only ARI	+LLM ARI	LLM Gain
100%	0.8231	0.8517	+0.0286
80%	0.801	0.834	+0.033
50%	0.742	0.790	+0.048
20%	0.621	0.682	+0.061

Table 25: **Marker-ablation experiment on MERFISH.** Performance degrades smoothly as marker information is removed, while LLM refinement provides larger relative gains under weaker priors.

Marker Retained	RB-only ARI	+LLM ARI
100%	0.5052	0.5408
50%	0.462	0.504
20%	0.392	0.445

Table 26: **Marker-ablation experiment on Visium.** Although performance decreases as priors weaken, the method remains stable and LLM refinement consistently improves predictions.

K.3 Neighborhood Context Analysis

NicheAgent incorporates neighborhood information in the reviewer prompt via a histogram of predicted niche types among spatial neighbors. These statistics are computed exclusively from rule-based predictions and do not use ground-truth annotations, preserving the zero-shot guarantee.

To evaluate whether neighborhood context causes cascading errors, we perform an ablation study on the MERFISH dataset. Table 27 reports performance when neighborhood features are progressively removed.

Removing the neighbor-type histogram results in a small decrease in accuracy (-0.008 ARI), while removing all neighborhood context yields a larger but still controlled decline (-0.026 ARI). Importantly, performance remains substantially higher than the RB-only baseline, indicating that the reviewer does not simply propagate local predictions.

These results suggest that neighborhood information acts as a supplementary signal rather than a dominant driver of predictions. The constrained LLM reviewer primarily relies on marker-gene evidence and ontology constraints, while spatial context provides incremental disambiguation near boundaries.

L Reproducibility

L.1 Implementation Details

Prototype construction, marker weighting, and spatial graph generation were implemented in Python using the Scanpy ecosystem. All experiments were run on a workstation with an NVIDIA A100 GPU and 64GB RAM.

The constrained reviewer used the Qwen2.5-14B-Instruct model with deterministic decoding (temperature = 0) and strict schema validation. Prompt templates, prototype libraries, and preprocessing scripts are available in the anonymous supplementary repository.

L.2 Code and NicheCard Release

To support transparent and rigorous evaluation during review, we have released an anonymous Zenodo

Setting	ARI
Full features	0.8517
No neighbor-type histogram	0.843
No neighborhood context	0.826

Table 27: **Neighborhood feature ablation.** Removing neighborhood statistics slightly reduces performance, indicating that spatial context provides complementary evidence without causing error propagation.

Setting	ARI	MF1	BACC
Prototype only	0.81	0.83	0.85
+ Marker enrichment	0.82	0.84	0.86
+ Neighbor context	0.83	0.85	0.87
+ LLM (no smoothing)	0.85	0.86	0.88
Full (all components)	0.8517	0.8667	0.8851

Table 28: Component-wise decomposition of NicheAgent.

archive containing all non-code artifacts required to reproduce the core logic, constraints, and LLM interactions used by NicheAgent. [1]

Specifically, the archive includes:

- All LLM prompt templates used in the pipeline, including system prompts, cell-level label selection prompts, strict JSON-output prompts, and role-specific instructions for the Analyst, Consensus, and Reviewer agents;
- Formal JSON schemas defining the exact LLM input–output contracts, enabling deterministic parsing and verification of model responses;
- Dataset- and experiment-ready alias maps documenting the mapping between canonical region labels and LLM-facing aliases used during inference;
- Ontology rule files specifying required and forbidden marker gene sets for each region label, which are used to enforce biological consistency and reject invalid LLM proposals;
- Documentation describing how these artifacts are instantiated and consumed by the NicheAgent pipeline.

These materials correspond exactly to the artifacts used in our experiments and allow reviewers to inspect, audit, and reuse the full decision logic governing LLM-based verification, ontology constraints, and label normalization, even in the absence of the executable code.

The complete NicheAgent codebase—including radius-based spatial graph construction, feature extraction, NicheCard (prototype) construction, inference logic, and evaluation scripts for all reported metrics—will be released upon publication. If requested during the review process, we are prepared to provide the full codebase to reviewers to facilitate deeper inspection or validation.

[1] Anonymous Zenodo record: https://zenodo.org/records/18159937

L.3 Data Availability

All spatial transcriptomics datasets and corresponding ground-truth annotations used in this work are downloaded from the SDMBench project page on Figshare: `https://figshare.com/projects/SDMBench/163942` (Yuan et al., 2024). We use the manual region annotations provided by the benchmark without modification. For each dataset, we report the specific annotation field used (e.g., `Region` / `Layer`) and dataset-level statistics in Appendix C.

M Supplementary Tables

Method	MERFISH	STARmap	Visium	Avg.	Rank
Non-Spatial					
Leiden	0.167±0.007	0.065±0.026	0.334±0.008	0.189	13.3
Louvain	0.157±0.007	0.055±0.017	0.335±0.014	0.182	13.7
Non-LLM based					
BASS	0.432±0.043	0.715±0.090	0.585±0.020	0.577	5.3
SCAN-IT	0.598±0.048	0.716±0.077	0.561±0.054	0.625	4.3
BayesSpace	–	–	0.565±0.079	0.565	5.0
GraphST	0.219±0.042	0.440±0.045	0.591±0.054	0.416	6.0
SpaceFlow	0.483±0.080	0.753±0.066	0.529±0.048	0.588	6.0
CCST	0.536±0.035	0.360±0.091	0.534±0.028	0.477	7.0
stLearn	–	–	0.543±0.009	0.543	7.0
STAGATE	0.207±0.087	0.564±0.069	0.509±0.037	0.426	9.3
SpaGCN	0.210±0.013	0.344±0.016	0.520±0.045	0.358	9.3
Gemini 1.5 Pro	0.045±0.025	0.753±0.043	0.411±0.079	0.403	10.7
conST	0.110±0.012	0.072±0.022	0.515±0.087	0.232	12.0
SEDR	0.116±0.036	0.114±0.058	0.455±0.067	0.228	12.3
SpaGCN(HE)	–	–	0.483±0.043	0.483	13.0
LLM-based					
LLMiniST-Fs	0.613±0.021	0.757±0.020	0.683±0.036	0.684	1.0
LLMiniST-F	0.580±0.017	0.755±0.018	0.662±0.044	0.666	2.3
LLMiniST-Z	0.068±0.031	0.471±0.090	0.068±0.031	0.202	11.0
NicheAgent (Ours)	**0.9025±0.018**	**0.7631±0.022**	**0.7112±0.015**	**0.792**	**1.0**

Table 29: HOM Score with ± SD across MERFISH, STARmap, and Visium. NicheAgent achieves the highest overall performance.

Method	MERFISH	STARmap	Visium	Avg.	Rank
Non-Spatial					
Louvain	0.183±0.011	0.079±0.029	0.337±0.016	0.200	13.3
Leiden	0.190±0.004	0.067±0.027	0.325±0.011	0.194	13.3
Non-LLM based					
BASS	0.651±0.070	0.672±0.107	0.577±0.022	0.633	3.3
GraphST	0.593±0.129	0.429±0.083	0.594±0.044	0.538	5.3
BayesSpace	–	–	0.566±0.096	0.566	6.0
SCAN-IT	0.560±0.045	0.566±0.057	0.533±0.045	0.553	6.7
stLearn	–	–	0.562±0.022	0.562	7.0
Gemini 1.5 Pro	0.153±0.036	0.757±0.038	0.554±0.110	0.488	7.3
SpaceFlow	0.603±0.077	0.508±0.059	0.367±0.037	0.493	8.3
SEDR	0.187±0.063	0.113±0.057	0.659±0.047	0.320	8.3
STAGATE	0.201±0.083	0.516±0.087	0.506±0.047	0.408	9.0
SpaGCN	0.218±0.018	0.297±0.013	0.506±0.050	0.340	9.7
CCST	0.415±0.030	0.350±0.082	0.483±0.021	0.416	9.7
conST	0.104±0.012	0.063±0.019	0.507±0.081	0.224	12.7
SpaGCN(HE)	–	–	0.468±0.042	0.468	14.0
LLM-based					
LLMiniST-Fs	0.608±0.033	0.748±0.006	0.707±0.040	0.688	2.0
LLMiniST-F	0.582±0.025	0.750±0.006	0.695±0.060	0.676	3.0
LLMiniST-Z	0.068±0.031	0.471±0.090	0.068±0.031	0.202	11.0
NicheAgent (Ours)	**0.9398±0.018**	**0.7977±0.020**	**0.7145±0.017**	**0.817**	**1.0**

Table 30: COM Score with ± SD across MERFISH, STARmap, and Visium. NicheAgent achieves the highest overall COM.