SPaRTA: Spatial Reasoning in Text-to-SQL for Omics Analysis

Anonymous ACL submission

Abstract

Spatial-omics datasets have transformed biomedical research by providing clinicians and pathologists access to multi-omics measurements, such as transcriptomic, proteomic, and metabolomic maps of tissue, at unprecedented resolution and scale. These massive datasets, now being generated and deposited worldwide, have the potential to revolutionize the discovery, diagnosis, and prevention of life-threatening diseases. However, despite their biological richness, these datasets remain largely inaccessible to clinicians and translational researchers who lack computational expertise. In this paper, we develop SPaRTA (Spatial Reasoning in Text-to-SQL for Omics Analysis) an end-to-end language-to-spatial-query translation system that uses a natural-language interface to translate free-text questions into executable spatial queries, dramatically lowering the barrier to interacting with complex tissue datasets. By adapting open-weight large language models (LLMs) (e.g., LLaMA and Mistral families) to publicly available real-world spatial-omics datasets, and comparing against proprietary closed-weighted models (e.g., GPT-4-class systems), we demonstrate that our system can construct syntactically correct SQL queries achieving performance comparable to proprietary alternatives. In addition, we introduce a benchmark dataset (SPaRTA-Bench) consisting of a reference schema and manually validated free-text question–SQL pairs, which can serve as a standardized resource for evaluating future research in this area.

1 Introduction

Recent advancements in imaging and sequencing technologies (Ståhl et al., 2016; de Oliveira et al., 2025; 10x Genomics, 2026; Rodriques et al., 2019; Stickels et al., 2021) have enabled profiling of biological tissues at unprecedented scale and resolution. These datasets are highly information-rich, containing high-dimensional measurements of thousands of genes across millions of cells, thereby creating comprehensive tissue maps that reveal important insights into underlying biological processes (Song et al., 2025). Such data are transformative because measuring molecular properties of individual cells within their native spatial context can help uncover key biological questions, such as how tumors interact with immune cells or which tissue regions are most susceptible to treatment response. These measurements further capture multiple molecular modalities, including transcriptomics, proteomics, and metabolomics. Collectively, these datasets are referred to as spatial-omics data.

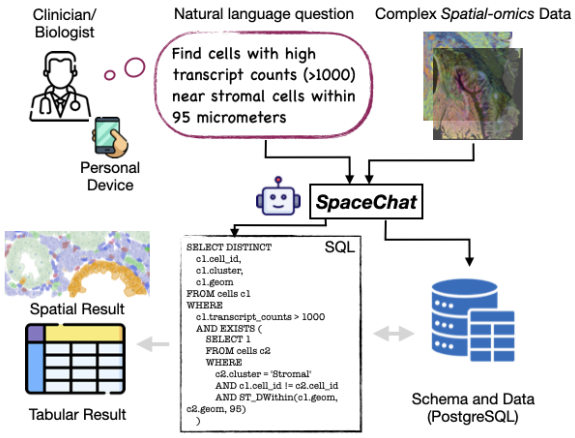

Figure 1: Overview of SPaRTA, a schema-grounded natural language interface for querying spatial-omics databases. Given a free-text biomedical question, SPaRTAtranslates the query into executable Spatial-SQL grounded in the underlying PostgreSQL/PostGIS schema, enabling spatial reasoning over complex tissue datasets and returning both tabular and spatially resolved results.

While such spatial molecular maps have the potential to transform the diagnosis and treatment of complex diseases such as cancer, effective in-

teraction with these datasets remains challenging for non-computational end users, including clinicians, pathologists, and biologists. First, unlike more familiar spatial modalities such as imaging, each spatial location in spatial-omics data is associated with a high-dimensional molecular profile, making direct visual inspection difficult. Second, individual datasets often span several gigabytes, which makes loading and interactively exploring impractical on standard computing devices. Finally, answering biologically meaningful questions often requires writing complex computational programs and database queries, creating technical barriers for users without data science or bioinformatics expertise.

As LLM-s become increasingly integrated into daily life, powering applications such as chatbots and healthcare assistants (Goyal et al., 2024), researchers are increasingly using tools such as ChatGPT to interact with clinical data. However, directly using such chatbots for spatial-omics datasets is not feasible for two main reasons. First, clinical datasets are often both sensitive and extremely large; therefore, uploading them to external generative AI systems is impractical and raises serious data privacy and security concerns (Faustini et al., 2025; Safarlou et al., 2021). Second, generative models are known to fabricate or hallucinate text and images (Huang et al., 2025; Bang et al., 2025; Ravichander et al., 2025), which in biomedical settings can lead to catastrophic consequences. For example, an interaction within a patient sample may be incorrectly annotated as cancerous due to a hallucinated threshold generated by the LLM. Moreover, unlike consumer-facing image generation tasks, biomedical imaging is not intended for cognitive or aesthetic satisfaction; therefore, a fabricated tissue segment holds no diagnostic value even if it appears visually realistic.

To address these bottlenecks, are increasingly being used to convert natural language questions into structured query language (SQL) queries, a task commonly referred to as NL2SQL. Designing NL2SQL methods is an active field of research (Liu et al., 2025), and related methods can broadly be divided into two classes: pre-LLM (Li et al., 2023a; Fu et al., 2023; Gao et al., 2022) and post-LLM (Talaei et al., 2024; Pourreza and Rafiei, 2024; Wang et al., 2023; Pourreza and Rafiei, 2023; Li et al., 2024). There are also schema-free methods (Qu et al., 2024; Li et al., 2023a) that rely on an intermediate representation with less restrictive grammar than that of SQL. Recent comprehensive benchmarks (Li et al., 2023b; Lei et al., 2025) (recent results are available in bird-bench[1]) show that recent LLM-based NL2SQL systems (Shkapenyuk et al., 2025) perform the best.

Srikanth et al., 2025 suggest that these improvements in NL2SQL models arise from an iterative process of identifying weaknesses and optimizing models to address them. However, applying off-the-shelf LLM as an interface to spatial-omics data is not straightforward. General SQL ground-truth datasets used for prompting and benchmarking are categorically different from the types of questions clinicians are interested in. For example, a pathologist examining breast cancer spatial-omics data may ask:

> "Which spatial domains show the highest fraction of cytotoxic T cells expressing GZMB that are proximal to proliferating tumor cell clusters?"

Contrasting this with:

> "Find the names of students who are enrolled in more than 3 courses and have a GPA above 3.5"

we observe that the former contains nuances unique to the biomedical domain, such as implicit biological knowledge and the use of spatial predicates.

1.1 Contributions

In this paper, we present SPaRTA: **Spa**tial **R**easoning in **T**ext-to-SQL for Omics **A**nalysis, that packages (1) a framework for extracting the database schema from spatial-omics data; (2) a semi-supervised agent-based model for generating relevant benchmark ground truth dataset; and (3) a tuning scheme for improving open-weight models to perform NL2SQL tasks. SPaRTA draws ideas of domain adaptation (Bogdanov et al., 2026), iterative improvement (Liu et al., 2022) in existing literature (Shi et al., 2025). We make the following concrete contributions:

A benchmark for spatial-omics NL2SQL. Grounded in geospatial text-and-SQL pairs (Hou et al., 2025), we introduce a benchmark SPaRTA-Bench from real Xenium data that captures this semantic complexity across six biologically motivated query categories $\{\kappa_j\}_{j=1}^{6}$, from simple spatial proximity to multi-hop reasoning over expression, neighborhood composition, and tissue

[1] https://bird-bench.github.io/

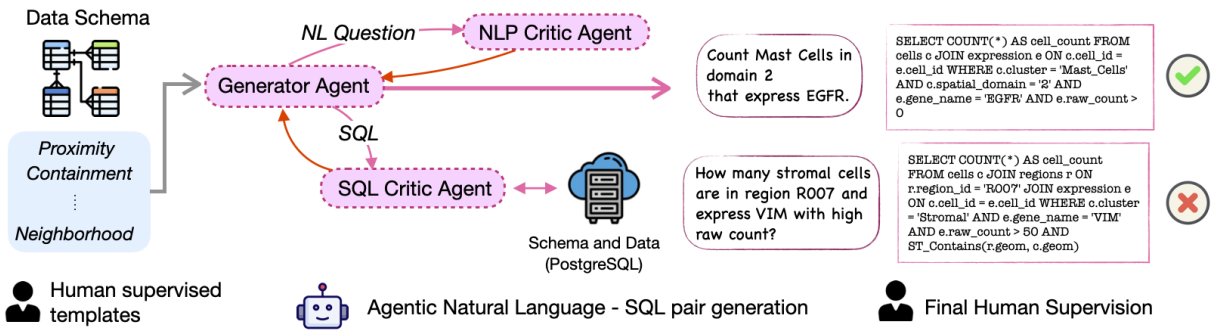

Figure 2: Overview of the agentic benchmark generation pipeline used in SPaRTA. Starting from human-supervised spatial query templates and database schemas, a generator agent produces natural language–SpatialSQL pairs conditioned on spatial reasoning operators such as proximity, containment, and neighborhood relationships. Dedicated NLP and SQL critic agents iteratively verify semantic faithfulness, schema consistency, and executability against the underlying PostgreSQL database. The generated pairs are subsequently filtered through final human supervision to construct a high-quality benchmark of executable and biologically meaningful NL2SQL pairs.

architecture. These datasets are generated using an agent-based framework, and finally evaluated by human evaluators with domain knowledge (see Figure 2).

Characterization of how NL2SQL models fail in biological domains. We show that off-the-shelf LLM-based models not only produce wrong SQL, but, they often produce plausible SQL that execute without an error but produce biologically incorrect results. We identify 19 such failure modes around three broad notions - structurally wrong SQL, faithfulness error (query runs and provides a different answer than what was asked), error in biological fact finding. Our findings point out that benchmarking just by successful execution is an unreliable metric for judging NL2SQL tasks.

A fine-tuning scheme for improving accuracy of open-weight LLMs. We fine-tuned five open-weight models on SPaRTA-Bench using low-rank adaptation (LoRA), (Hu et al., 2022) and demonstrate comparable performance with proprietary models.

2 Design Considerations

The NL2SQL query pairs required to test any spatial-omics interface task is fundamentally different from other existing benchmarks such as SPIDER (Yu et al., 2018; Lei et al., 2025). Therefore, SPaRTA made several design choices to make the benchmark data and later the fine-tuning realistic. We followed three principles to achieve this.

- **Implicit biological concepts.** When a pathologist asks about *infiltrating immune cells near a DCIS lesion*, the question presupposes knowledge of cell type hierarchies, marker-cluster correspondences, and tissue architecture that exists entirely outside the database schema. Unlike general-purpose NL2SQL where entities are associated by their syntactic relationship, spatial-omics queries require resolving implicit biological concepts, such as, which cell types constitute *immune cells*, which clusters represent *DCIS* (there can be cell sub-types within *DCIS*, named by *DCIS_1*, *DCIS_2* etc), what radius constitutes *near* etc., before a valid SQL query can even be formulated. A benchmark that does not test these nuances is not sufficient for assessing these interface.
- **Correctness of queries.** In standard benchmarks, a query that executes and returns the correct rows is correct. This assumption breaks down in spatial-omics. Consider a query from SPaRTA-Bench , *"compare average GZMB expression of CD8+ T cells within 50 micrometers of invasive tumor cells versus those that are not."* The generated SQL computes `AVG(e.raw_count)` via an inner join on the expression table. It executes without error and returns two numbers: one for proximal cells, one for distal cells. But because single-cell expression data is stored sparsely — absent rows encode zero counts, not missing data — the inner join silently drops all *CD8+ T* cells with no *GZMB* entry from both groups. The average is computed only over cells that express *GZMB* at all, inflating both values. This inflation could be asymmetric: if proximal *CD8+ T* cells have a higher proportion of zero-expression than distal ones, the proximal group will appear artificially less cytotoxically active, **potentially reversing the scientific conclusion about immune activation near the tumor**, leading to a wrong diagnosis. We call these *silent*

failures: errors invisible to execution-based evaluation that directly affect scientific interpretation.

- **Semantical difference between general language and biological questions.** The gap between the general purpose NL2SQL and spatial-omics related NL2SQL is not limited to the unfamiliarity of SQL constructs present in biomedical questions, or new schema. We observe that biological terminology carries precise technical definitions that are not analogous to general languages, and a model without domain grounding will produce incorrect result even when the generated SQL is syntactically valid. Consider the question *find invasive tumor cells with CD8+ T cell infiltration within 60 micrometers."* The word *infiltration* in natural language implies movement or presence; in spatial pathology it denotes a specific quantitative relationship, the density of cytotoxic lymphocytes within a defined radius of a tumor cell, that must be present as a neighborhood count, not a simple proximity check. As a result, domain adaptation cannot be achieved by sharing the schema alone, but requires grounding the model in conventions of spatial biology.

2.1 Evaluated Models

We investigate five proprietary model families: GPT (OpenAI, 2024); Gemini (Comanici et al., 2025); Grok (xAI, 2025); Claude (Anthropic, 2025); and DeepSeek (Bi et al., 2024). For each model family, we consider the following models (version details are in the Appendix): GPT (*GPT-5*); Gemini (*Gemini-2.5-flash*); Grok (*Grok-4*); Claude (*Claude-opus-4*); and DeepSeek (*Deepseek-reasoner*). In addition, we consider five open-weight models Qwen-7B-Instruct (Yang et al., 2025), GPT-OSS-20B ((Agarwal et al., 2025)), Llama 3.1 8B-Instruct (Grattafiori et al., 2024), Gemma-4-4B-Instruct ((Team et al., 2025)), and Mistral ((Mistral AI Team, 2024)) (Appendix C contains further details).

3 Dataset and Benchmark Construction

After taking the design considerations into account we followed a biologically grounded 5 step approach to construct a benchmark NL2SQL dataset that contains queries from different categories spread across 3 categories of complexity: easy, medium, and hard, depending on the complicacy of generated the SQL query.

- **Phase 1 Few-shot foundation.** We used GeoSQL-Bench (Hou et al., 2025), which contains over $7,488$ NL2SQL pairs $\{(q_i, s_i)\}$ spanning 299 geometric functions. In practice, the few-shot grounding process restricts to a small subset of 12 PostGIS functions $\mathcal{F}$ (Figure 3c) centered on proximity (ST_DWithin, ST_Distance), containment (ST_Contains, ST_Within), region properties (ST_Area, ST_Boundary), and spatial aggregation (ST_Collect, ST_ClusterDBSCAN, ST_ConvexHull). This restriction is motivated by the fact that spatial transcriptomics operates on a geometrically constrained schema where point geometries and polygon geometries are sufficient for delineating cells and tissue regions respectively. Other spatial operations such as routing, raster operations, coordinate transformations, and topological predicates are less relevant to biological questions.
- **Phase 2 Schema extraction.** We derived the ground-truth schema $\mathcal{S}$ directly from an h5ad file containing Xenium breast cancer spatial transcriptomics data (Janesick et al., 2023). The extraction produces two conceptually distinct types of spatial annotation. First, *spatial domain labels* $\mathcal{D}$ are per-cell categorical annotations (column spatial_domain in the cells table) encoding biologically meaningful groupings such as cell-type neighborhoods or niches. These are labels on cells, not geometric objects. Second, *regions* are spatially contiguous polygon geometries stored in a separate regions table, derived by computing polygons over cell coordinates per domain or loaded from an external GeoJSON annotation. Cell coordinates, cluster labels $\mathcal{C}$, a sparse expression matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{n_c \times n_g}$ (n_c cells, n_g genes), domain labels $\mathcal{D}$, and region geometries were loaded into a PostGIS database, producing the ground-truth tables (cells, expression, regions) and schema $\mathcal{S}$. The entity vocabulary $\mathcal{V} = (\mathcal{C}, \mathcal{G}, \mathcal{D})$ of cluster labels, gene names $\mathcal{G}$, and domain labels was derived in Phase 3 by reading the extracted data files directly, ensuring that slot values in generated templates reference only entities present in the actual dataset.
- **Phase 3 Template generation** We defined a six-category query classification $\{\kappa_j\}_{j=1}^{6}$ covering spatial proximity, containment, biological filtering, neighborhood aggregation, region-level statistics, and compositional multi-hop queries. Each category κ_j was explicitly assigned a subset of $\mathcal{F}$, constraining the generator to only the functions relevant to that query type. A small set of seed templates $\mathcal{T}_{\text{seed}}$ were authored manually using real values from $\mathcal{V}$; an LLM (Claude Haiku 4.5) then

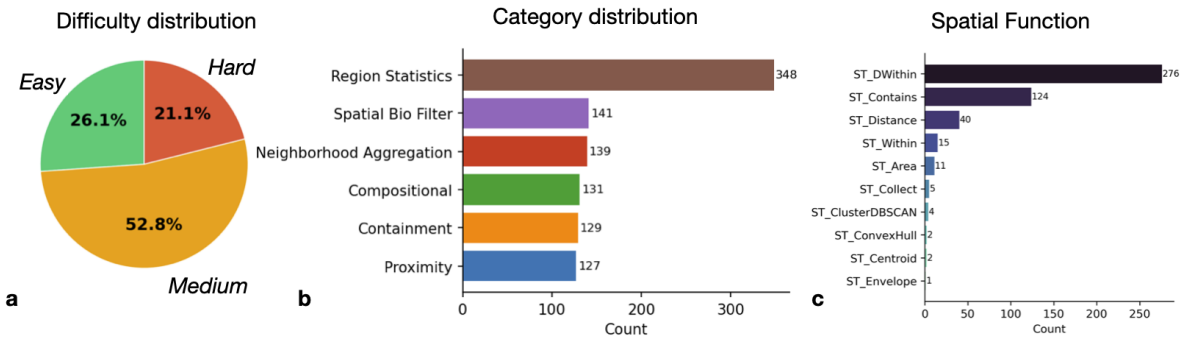

Figure 3: Characteristics of SPaRTA-Bench . (**a**) Distribution of query difficulty levels, showing that the benchmark contains a substantial proportion of medium and hard examples. (**b**) Distribution of high-level query categories, including region statistics, spatial biological filtering, neighborhood aggregation, compositional analysis, containment, and proximity-based reasoning. (**c**) Frequency of spatial functions used in generated SpatialSQL queries, highlighting the prevalence of geometric predicates such as ST_DWithin and ST_Contains.

generated additional templates $\mathcal{T}_{\text{gen}}$ per category conditioned on $(\mathcal{S}, \mathcal{V})$ and the few-shot examples from Phase 1 matched by PostGIS function. We have made a few representative candidates by using each template is a pair (q_t, s_t) of a natural-language question and its corresponding SQL query, with placeholders for cell types, gene names, spatial domains, and distance thresholds drawn from $\mathcal{V}$. The representative set is further refined by experimental biologists, and pathology students.

- **Phase 4 Initial NL2SQL pairs construction.** Each template t was instantiated $k = 20$ times by sampling from σ_t using a controlled strategy that varied entity frequency (common vs. rare cluster labels), spatial radii $r \in \{25, 50, 100, 200\}$ μm, and expression thresholds, producing a candidate set $\mathcal{B}_{\text{cand}}$. Every candidate SQL query was executed against the live PostGIS database to verify executability and record row count $|\mathcal{R}_i|$. Each example was assigned a difficulty label $\delta_i \in \{\text{easy}, \text{medium}, \text{hard}\}$ and an expected result type $\rho_i \in \{\text{cells}, \text{counts}, \text{aggregated_table}, \text{ranked_table}, \text{scalar}\}$.
- **Phase 5 Agentic review loop.** A generator agent, $\mathcal{A}_G$, produces 20 candidate pairs per template. We applied a near-duplicate filter using token-level Jaccard similarity on normalized questions, discarding any candidate for which $J(q_i, q_j) > 0.85$ against previously accepted questions within the same category template. The normalization process replaces numbers and domain-specific entities (e.g., cluster labels, gene names, and domain identifiers) with placeholders.

The surviving candidates were evaluated by two independent review agents (Figure 2) with complementary responsibilities. A semantic reviewer, $\mathcal{A}_R^{\text{sem}}$, acts as an NLP critic agent and assesses each candidate across six different dimensions: SQL faithfulness, biological plausibility, realism of the natural language, non-triviality, and value as a benchmark candidate. $\mathcal{A}_R^{\text{sem}}$ assigns an integer score from 0–5 for each dimension and returns a decision of *keep*, *revise*, or *reject* based on predefined thresholds. A syntactic validator, $\mathcal{A}_R^{\text{syn}}$, acting as an SQL critic agent, independently executes each candidate SQL query against the live PostGIS database, separate from the semantic review process. Rows not receiving a *keep* decision from either agent entered a targeted repair loop for up to a predefined maximum number of rounds. In each round, SQL-failed rows were repaired by re-prompting $\mathcal{A}_G$ with the exact execution error; *revise* rows had their natural language rewritten while preserving the SQL query; and *reject* rows were fully regenerated from the same template. A row entered the final benchmark $\mathcal{B}$ only if: (*i*) $\mathcal{A}_R^{\text{syn}}$ confirmed SQL execution, and the relevant scores are over predefined threshold. The resulting benchmark Text-SQL pairs are distributed across diverse complexities, categories (Figure 3a,b). Evan after going through this rigorous agent based review cycle spanning $\mathcal{A}_G$, $\mathcal{A}_R^{\text{sem}}$, and $\mathcal{A}_R^{\text{syn}}$, upon manual inspection we observe many failure modes.

4 Failure Mode Analysis and Recovery

Manual inspection of cases where the SQL was syntactically valid and executable but produced results inconsistent with the natural language intent revealed approximately 19 categories of errors (see Appendix Table 4). In what follows, we present a

few interesting examples.

We observe that failures in LLM-based generation arise from three broad areas. The most prevalent category is *structural failure*, which causes the generation of incorrect SQL constructions independent of the intended question. One of the most impactful examples is the sparse HAVING threshold bug (Table 1): for a question asking to list stromal cells with average *EPCAM* expression > 3 among neighboring *DCIS_1* cells, the model uses an inner join that silently excludes cells with zero expression, thereby overestimating the average. A similar error also occurs when calculating denominators, especially when the term "fraction" is mentioned in the question. For example, when asked to rank spatial domains by the fraction of GZMB-expressing CD8+_T_Cells near tumors, the model divides by total_cd8 — all CD8+_T_Cells — rather than only GZMB-expressing cells, asymmetrically understating the fraction across domains and potentially reversing the clinical ranking.

Faithfulness failures occur when the model hallucinates conditions that are not present in the question. A representative case is the phantom gene join: for the question *"find proliferating invasive tumor cells in domain 2,"* the model joins the expression table and filters on e.gene_name = 'MKI67' — a gene never mentioned in the question. This false association may arise from the implicit connection the model makes between the concept of "proliferative" and the gene *MKI67*.

Another class of errors, termed *semantic failures*, reflects a mismatch between biological language and SQL semantics. Broad category terms such as *immune cells* or *DCIS cells* are mapped to incomplete cluster lists, spatial qualifiers such as *cluster* are silently reduced to single-cell proximity checks, and region identifiers are hallucinated. In all three categories, the queries execute without error and return plausible results — the defining property that makes these failures invisible to purely execution-based evaluation.

For queries that failed human inspection, we applied a targeted repair step: the identified failure category and a corrected SQL example were provided as few-shot demonstrations to claude-opus-4-6, which then generated a revised query. This two-pass process of automatic generation followed by failure-aware repair forms the basis of the validation framework described in Appendix B. This process ultimately recovered 1015 NL2SQL pairs (SPaRTA-Bench), termed as $\mathcal{B}_{\text{gold}} \subseteq \mathcal{B}$ that were used for benchmarking.

5 Experiments

5.1 NL2SQL Generation using Different LLMs

All models were evaluated in a zero-shot setting. No few-shot examples, retrieval augmentation, or chain-of-thought scaffolding were used. Each question was issued as an independent call with no shared context across queries, so results reflect each model's raw generation capability given only the schema and the question.

- **Prompt construction.** Each call consists of two parts: a system message containing the database schema $\mathcal{S}$ and a fixed instruction, and a user message containing the natural language question q. The system message serves two roles: it grounds the model in the concrete table and column definitions (preventing hallucinated schema references), and it constrains the output format to bare SQL with no surrounding explanation or markdown. The full system prompt used across all models and all experiments is shown in Figure 6.
- **Model-specific prompt handling.** For open-weight models, the system message is passed via the model's native chat template. Most models accept a dedicated system role; otherwise, the schema is prepended to the first user turn. GPT-oss-20B uses the developer role per its Harmony chat format, and thinking models have chain-of-thought disabled at tokenization time to prevent reasoning traces from appearing in the output. All open-weight models are run locally with greedy decoding (temperature 0) ensuring deterministic outputs. For propr models accessed via API, we use the lowest available temperature setting and report the response as-is; internal retrieval augmentation or system-level instructions injected by the provider cannot be ruled out, so results for these models should be interpreted as reflecting the provider's deployed system rather than the raw model.

5.2 Finetuning and domain adaptation

We adapt each open-weight model using Low-Rank Adaptation (LoRA; Hu et al. 2022). $\mathcal{B}_{\text{gold}}$ was split 75/25 into training and held-out test sets, with the training portion further divided 90/10 for epoch-level validation. All splits used a fixed random seed. Fine-tuned models are evaluated on the held-out test split only. Each example is formatted using the model's native chat template, with the full database schema $\mathcal{S}$ as a system prompt instructing the model

Failure Mode	Question	Generated SQL (abbreviated)
Sparse AVG + HAVING threshold	List stromal cells with average EPCAM expression > 3 among neighboring DCIS_1 cells within 60 μm.	... JOIN expression e ON e.gene_name = 'EPCAM' GROUP BY c1.cell_id HAVING AVG(e.raw_count) > 3
Hallucinated expression filter	Find proliferating invasive tumor cells in domain 2.	... WHERE c.cluster = 'Prolif_Invasive_Tumor' AND e.gene_name = 'MKI67' AND e.raw_count > 0
Wrong denominator	Rank spatial domains by the fraction of CD8+_T_Cells expressing GZMB that reside within 100 μm of proliferating invasive tumors.	WITH cd8_gzmb_near_pit AS (. . .), all_cd8 AS (SELECT spatial_domain, COUNT(DISTINCT cell_id) AS total_cd8 FROM cells WHERE cluster = 'CD8+_T_Cells' GROUP BY . . .) SELECT . . . COUNT(. . .) / NULLIF(total_cd8 , 0) AS fraction

Table 1: Three catastrophic NL2SQL failure modes identified in SPaRTA. All three queries execute without error and return numerically plausible results, making them invisible to execution-based evaluation. Highlighted regions indicate the erroneous SQL fragment in each case.

to return bare SQL only. For GPT-oss-20B the schema is passed under the developer role per its Harmony chat format; for models without a dedicated system role, the schema is prepended to the user turn. For thinking models, chain-of-thought is disabled at tokenization time.

5.3 SPaRTA Evaluation Framework

Subtle differences in SQL queries, such as the order of rows, names of columns, or retrieval of additional columns, can drive divergence in the resulting table when compared to the ground truth. When the evaluation metric does not account for these differences, the performance of an NL2SQL model is underestimated, even when the query leads to partial retrieval of the correct result. In SPaRTA, we use several performance metrics (Lei et al., 2025; Sutanto et al., 2025) that capture different degrees of overlap between the ground-truth table T_{gold} and the candidate table T_{pred} produced by the predicted query from the NL2SQL models. Apart from using the exact-match metrics such as execution accuracy, or soft F1[2] commonly used in existing literature (Yu et al., 2018; Li et al., 2023b), we perform a finer comparison between T_{gold} and T_{pred} by separately calculating overlap between the corresponding sets of rows $R(.)$, where $r \in R(.)$ represents a tuple of row entries, and column names $C(.)$, and then combining them to obtain the final score, termed SPaRTA-F1. Formally,

- **Execution Accuracy (EA)**: It's a binary metric that measures the exact match between the flatten version of output. A row is regarded as a table.

$$\text{EA} = \mathbf{1}[R(T_{\text{gold}}) = R(T_{\text{pred}})]$$

- **Soft F1**[2]: After deduplication for each row pair (r_i^p, r_i^g) from T_{gold} and T_{pred} respectively, we count how many values in r_i^p appear in r_i^g (as a set) and vice versa, accumulating TP, FP, and FN counts. Unpaired rows due to length mismatch contribute to FP or FN. The final score is:

$$\text{Soft-F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{1}$$

This metric assumes column order matches between T_{pred} and T_{gold}.

- **SPaRTA F1** For SPaRTA, we first align columns via an injective mapping $\phi : C(T_{\text{pred}}) \rightarrow C(T_{\text{gold}})$. For each unmatched column c_p, we assign $\phi(c_p) = \arg\max_{c_g} s(c_p, c_g)$ where $s(c_p, c_g) = J(V_{c_p}, V_{c_g}) + \alpha \cdot \mathbf{1}[V_{c_p} \subseteq V_{c_g}]$, with J the Jaccard similarity and $\alpha > 1$. Let $k = |\phi|$. The column F1 is:

$$\text{F1}_{\text{col}} = \frac{2k}{|C(T_{\text{pred}})| + |C(T_{\text{gold}})|} \tag{2}$$

After projecting both tables to aligned columns, row precision and recall are computed symmetrically, and the row F1 is:

$$\text{F1}_{\text{row}} = \frac{2 \cdot |R(T_{\text{pred}}) \cap R(T_{\text{gold}})|}{|R(T_{\text{pred}})| + |R(T_{\text{gold}})|} \tag{3}$$

The final score is:

$$\text{SPaRTA F1} = \text{F1}_{\text{col}} \times \text{F1}_{\text{row}} \tag{4}$$

The multiplicative combination ensures failure on either axis drives the score to zero.

[2] https://github.com/bird-bench/mini_dev/

Model	SPaRTA-F1	Std	Median
Closed-weight models			
GPT-5.5	**0.6028**	0.3750	0.6667
Gemini 2.5 Flash	0.5804	0.3848	0.6667
Claude Sonnet 4.6	0.5746	0.3892	0.6667
DeepSeek-V4 Flash	0.5696	0.3900	0.6667
Mistral Large 25.12	0.5358	0.3984	0.6089
Grok-4.3	0.4538	0.4101	0.4000
Open-weight models			
GPT-OSS-20B	0.4716	0.4101	0.4667
Gemma-4 4B-it	0.4523	0.4035	0.4000
Qwen-7B-Instruct	0.3145	0.3941	0.0000
Gemma-2 9B-Instruct	0.3139	0.4035	0.0000
Llama-3.1 8B-Instruct	0.1627	0.3195	0.0000
Mistral-7B-Instruct-v0.3	0.1123	0.2712	0.0000

Table 2: SPaRTA-F1 performance of proprietary and open-weight NL2SQL models on the SPaRTA benchmark.

5.4 Results

On benchmark NL2SQL pairs $\mathcal{B}_{\text{gold}}$, closed-weight models achieve SPaRTA-F1 scores between 0.54–0.60, with no single model dominating decisively (Table 2). First, we note that barring Grok-4.3, the zero-shot performance of all other proprietary models are better than that of the open-weight models with the best-performing proprietary model (GPT-5.5) performing considerably better than open-weight counterparts. On the other hand, open-weight models in their base configuration lag substantially behind. We observe a pattern to the failure mode of open-weight models prompted zero-shot. `Llama-3.1-8B` and `Mistral-7B-Instruct-v0.3` fail to produce executable SQL on 61.4% and 69.0% of queries respectively, with SPaRTA-F1 scores below 0.17. The failure modes here are predominantly structural: models generate syntactically invalid SQL or hallucinate table and column names absent from the schema. `Gemma-4-4B` and `GPT-oss-20B` show much lower failure rates (9.6% and 12.6%), indicating better schema adherence in the base configuration despite weaker semantic accuracy.

However, all open-weight models show considerable improvement upon fine-tuning (Figure 4) on $\mathcal{B}_{\text{gold}}$, with `Mistral-7B` showing the largest absolute gain. Several fine-tuned models reach or exceed the SPaRTA-F1 of `Claude` Sonnet 4.6 and `Gemini` 2.5 Flash.

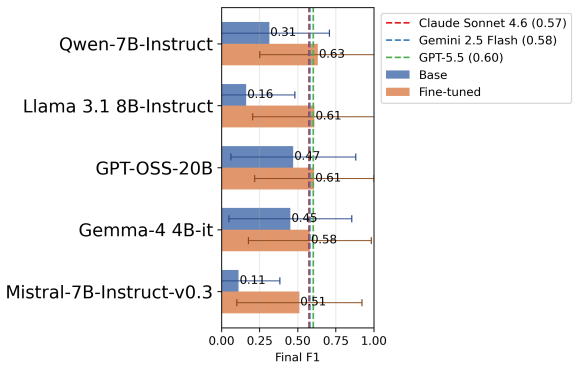

Figure 4: Comparison of SPaRTA-F1 score base and fine-tuned open-weight NL2SQL models on the SPaRTA benchmark.

6 Conclusion

Just like geospatial maps, spatial-omics data could transform medical diagnosis and biological research by changing how diseased tissue samples are explored. However, these datasets are complex, high-dimensional, and sensitive with respect to data privacy. Moreover, biological questions are semantically and syntactically different from conversational language, making existing NL2SQL interfaces and benchmark datasets insufficient for spatial-omics applications.

In this manuscript, we address this gap through SPaRTA, a benchmark dataset with verified NL2SQL pairs for spatial-omics analysis. Starting from an expert-curated template library, SPaRTA employs an agent-based generate-review pipeline to ensure both biological groundedness and scalability. We further identify multiple failure modes that can lead to *silent failures*, where generated SQL queries are executable yet produce incorrect biological conclusions due to subtle numerical or semantic errors.

To evaluate current systems, we benchmarked 12 leading LLM models using a standardized prompt and additionally fine-tuned 5 open-weight models on the proposed dataset. The results show that several open-weight models improve substantially after fine-tuning and achieve performance comparable to closed-weight systems. These findings suggest a practical path toward privacy-preserving natural language interfaces for large-scale spatial-omics data, enabling clinicians to interact with tissue maps similarly to geospatial maps.

7 Limitations

Our paper has following limitations,

7.1 Data size limitation

The benchmark dataset $\mathcal{B}_{\text{gold}}$ is limited in size and built on a single spatial-omics dataset from breast cancer. This dataset represents only one technology, `Xenium`, whereas a number of spatial transcriptomics platforms exist with varying spatial resolution and gene panels. The use of a single breast cancer sample further constrains biological variability in terms of cell types and expressed genes. Generalizing `SPaRTA` to other biological samples and conditions will require expanding the underlying data to cover additional technologies, tissue types, and disease contexts.

7.2 Template specificity and semantic bias

We use a template centric (6 templates) initiation process. As a result the dataset lacks comprehensibility. Biologically motivated questions that arise organically in clinical practice, such as queries involving multi-step reasoning across cell states, time-series comparisons, or differential analysis across multiple tissues are absent because they do not fit naturally into the limited template vocabulary. More broadly, the full space of clinically relevant spatial-omics queries can only be characterized through continuous engagement with pathologists and clinicians over extended periods of time.

7.3 Human annotation gap

The failure modes in `SPaRTA` were identified through expert developer analysis of model-generated queries rather than through engagement with practicing pathologists or clinicians. Clinicians reason about tissue biology in terms of diagnostic categories, spatial gradients, and cell state transitions that do not map naturally to `SQL` queries. As a result, the failure modes we characterize likely represent an incomplete picture. Sustained collaboration with clinical users will be necessary to surface broader diagnostic failure patterns and improve the coverage of the benchmark.

8 Ethical considerations

`SPaRTA`uses the publicly available `GeoSQL-eval` for grounding the generative process, and use a publicly available breast cancer (Janesick et al., 2023) available from 10x genomics website. The generation of the benchmark uses proprietary models accessed with API keys. However, we have shown that with fine tuning the open-weight free models achieve comparable performance to NL2SQL-task. All experimental biologists involved in template selection has extensive knowledge of producing and working with spatial-omics datasets. The NL2SQL-validation was carried out by students with `SQL`and AI knowledge under the supervision of senior researchers ensuring there is no overlap between student annotators and students who ran modeling experiments. Student annotators were given precise set of instructions. Finally, we used *ChatGPT* as a writing assistant for flow and grammar. All scientific decisions, interpretations, and core arguments were made by the authors.

References

10x Genomics. 2026. Xenium in situ platform. https://www.10xgenomics.com/platforms/xenium. Accessed: 2026-05-24.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Anthropic. 2025. Introducing claude 4. Technical report, Anthropic.

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. 2025. HalluLens: LLM hallucination benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24128–24156, Vienna, Austria. Association for Computational Linguistics.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Olena Bogdanov, Yeunji Jung, Chandra Dhir, Pareekshitreddy Gaddam, Saurabh Jain, Lakshmi Tumati, Vijay Parthasarathy, and Anup Shirgaonkar. 2026. Agentnlq: A general-purpose agent for natural language to sql. *arXiv preprint arXiv:2605.19010*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, and Others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Michelli Faria de Oliveira, Juan Pablo Romero, Meii Chung, Stephen R. Williams, Andrew D. Gottscho, Anushka Gupta, Susan E. Pilipauskas, Seayar Mohabbat, Nandhini Raman, David J. Sukovich, and 1

others. 2025. High-definition spatial transcriptomic profiling of immune cell populations in colorectal cancer. *Nature Genetics*, 57(6):1512–1523. Introduces and applies the Visium HD spatial transcriptomics platform in human colorectal cancer.

Pedro Faustini, Annabelle McIver, Ryan Sullivan, and Mark Dras. 2025. De-identification of clinical data: a systematic review of free text, image and tabular data approaches. *International Journal of Medical Informatics*, page 106225.

Han Fu, Chang Liu, Bin Wu, Feifei Li, Jian Tan, and Jianling Sun. 2023. Catsql: Towards real world natural language to sql applications. *Proceedings of the VLDB Endowment*, 16(6):1534–1547.

Chang Gao, Bowen Li, Wenxuan Zhang, Wai Lam, Binhua Li, Fei Huang, Luo Si, and Yongbin Li. 2022. Towards generalizable and robust text-to-sql parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2113–2125.

Sagar Goyal, Eti Rastogi, Sree Prasanna Rajagopal, Dong Yuan, Fen Zhao, Jai Chintagunta, Gautam Naik, and Jeff Ward. 2024. Healai: A healthcare LLM for effective medical documentation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 1167–1168.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Shuyang Hou, Haoyue Jiao, Ziqi Liu, Lutong Xie, Guanyu Chen, Shaowen Wu, Xuefeng Guan, and Huayi Wu. 2025. Geosql-eval: First evaluation of llms on postgis-based nl2geosql queries. *arXiv preprint arXiv:2509.25264*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

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

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, and 1 others. 2025. Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows. In *International Conference on Learning Representations*, volume 2025, pages 28691–28735.

Boyan Li, Yuyu Luo, Chengliang Chai, Guoliang Li, and Nan Tang. 2024. The dawn of natural language to sql: Are we fully ready? *arXiv preprint arXiv:2406.01265*.

Haoyang Li, Jing Zhang, Cuiping Li, and Hong Chen. 2023a. Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13067–13075.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, and 1 others. 2023b. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36:42330–42357.

Aiwei Liu, Xuming Hu, Li Lin, and Lijie Wen. 2022. Semantic enhanced text-to-sql parsing via iteratively learning schema linking graph. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1021–1030.

Xinyu Liu, Shuyu Shen, Boyan Li, Peixian Ma, Runzhi Jiang, Yuxin Zhang, Ju Fan, Guoliang Li, Nan Tang, and Yuyu Luo. 2025. A survey of text-to-sql in the era of llms: Where are we, and where are we going? *IEEE Transactions on Knowledge and Data Engineering*.

Mistral AI Team. 2024. Au large: Mistral large, our new cutting-edge text generation model. Mistral AI blog. Accessed: 2025-08-01.

OpenAI. 2024. Gpt-4o technical report. Accessed: 2025-08-02.

Mohammadreza Pourreza and Davood Rafiei. 2023. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *Advances in neural information processing systems*, 36:36339–36348.

Mohammadreza Pourreza and Davood Rafiei. 2024. Dts-sql: Decomposed text-to-sql with small large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8212–8220.

Ge Qu, Jinyang Li, Bowen Li, Bowen Qin, Nan Huo, Chenhao Ma, and Reynold Cheng. 2024. Before generation, align it! a novel and effective strategy for mitigating hallucinations in text-to-sql generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5456–5471.

Abhilasha Ravichander, Shrusti Ghela, David Wadden, and Yejin Choi. 2025. HALoGEN: Fantastic LLM hallucinations and where to find them. In *Proceedings of the 63rd Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 1402–1425, Vienna, Austria. Association for Computational Linguistics.

Samuel G. Rodriques, Robert R. Stickels, Aleksandrina Goeva, Bryan Martin, Elior Murray, Charles R. Vanderburg, Joshua Welch, Linlin M. Chen, Fei Chen, and Evan Z. Macosko. 2019. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463–1467.

CW Safarlou, AL Bredenoord, R Vermeulen, and KR Jongsma. 2021. Scrutinizing privacy in multi-omics research: how to provide ethical grounding for the identification of privacy-relevant data properties. *The American Journal of Bioethics*, 21(12):73–75.

Liang Shi, Zhengju Tang, Nan Zhang, Xiaotong Zhang, and Zhi Yang. 2025. A survey on employing large language models for text-to-sql tasks. *ACM Computing Surveys*, 58(2):1–37.

Vladislav Shkapenyuk, Divesh Srivastava, Theodore Johnson, and Parisa Ghane. 2025. Automatic metadata extraction for text-to-sql. *arXiv preprint arXiv:2505.19988*.

Liyang Song, Wenhao Chen, Junren Hou, Minmin Guo, and Jian Yang. 2025. Spatially resolved mapping of cells associated with human complex traits. *Nature*, 641(8064):932–941.

Neha Srikanth, Victor Bursztyn, Puneet Mathur, and Ani Nenkova. 2025. Sqlspace: A representation space for text-to-sql to discover and mitigate robustness gaps. *arXiv preprint arXiv:2510.27532*.

Patrik L. Ståhl, Fredrik Salmén, Sanja Vickovic, Anders Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O. Westholm, Linda Härndahl, and 1 others. 2016. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82.

Robert R. Stickels, Elior Murray, Pavan Kumar, Jean Li, Jeffrey L. Marshall, Daniel J. Di Bella, Paola Arlotta, Evan Z. Macosko, and Fei Chen. 2021. Highly sensitive spatial transcriptomics at near-cellular resolution with slide-seqv2. *Nature Biotechnology*, 39(3):313–319.

Patrick Sutanto, Jonathan Kenrick, Max Lorenz, and Joan Santoso. 2025. Llm-driven data generation and a novel soft metric for evaluating text-to-sql in aviation mro. *arXiv preprint arXiv:2506.13785*.

Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, and Amin Saberi. 2024. Chess: Contextual harnessing for efficient sql synthesis. *arXiv preprint arXiv:2405.16755*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Bing Wang, Changyu Ren, Jian Yang, Xinnian Liang, Jiaqi Bai, Qian-Wen Zhang, Zhao Yan, and Zhoujun Li. 2023. Mac-sql: A multi-agent collaborative framework for text-to-sql. corr abs/2312.11242 (2023). *arXiv preprint arXiv:2312.11242*, 10.

xAI. 2025. Grok 4. Technical report, xAI.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, and 1 others. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3911–3921.

A Template based Examples

We selected 6 templates as a basis for generating SQL queries. This templates span myriad of spatial queries -

- *Proximity*: Spatial closeness is one of the major biologically interesting properties captured by spatial-omics data. The genes that co-vary together or cell-types bordering each other, encode biological processes and suggest gene-gene interaction or tumor boundary that are resisting immune cells.

- *Containment*: Regions often signify localized activity in a tissue. Such queries reveal the cell-type composition of a biological compartment. Geometric property of these regions can suggest the changing landscape of a tissue organization in presence of a disease.

- *Spatial + Bio Filter*: The spatial gene expression pattern plays a crucial role in disease progression, reshaping the tumor microenvironment. Therefore, thresholding over a gene expression value and subsquently selecting a subset of spatial points is important.

- *Region Statistics*: Spatial density, autocorrelation (moran's I), etc. are informative features to study spatially varying genes, constellation of specific cell types, niche etc.

- *Compositional*: Complex query where a selection is made followed by a mathematical operation are crucial to benchmark on. The order of the SQL operations is important for getting the right result.

Together SPaRTA-Bench contains a rich set of NL2SQL pairs that coveres wide variety of spatial-omics questions.

B Complete NL2SQL Failure Modes in SPaRTA

The following table enumerates all 18 failure modes identified through systematic analysis of model-generated queries in SPaRTA. For each failure mode we provide the natural language question, the erroneous SQL (abbreviated), and a brief explanation. Highlighted fragments indicate the erroneous portion.

Category	NL Question	SQL Query
κ_1 Proximity	For each spatial domain, how many CD8+_T_Cells are located within 50 μm of invasive tumor cells?	SELECT c1.spatial_domain, COUNT(DISTINCT c1.cell_id) FROM cells c1 WHERE c1.cluster = 'CD8+_T_Cells' AND EXISTS (SELECT 1 FROM cells c2 WHERE c2.cluster = 'Invasive_Tumor' AND c1.cell_id != c2.cell_id AND ST_DWithin(c1.geom, c2.geom, 50)) GROUP BY c1.spatial_domain
κ_2 Containment	How many stromal cells in domain 3 are within 100 μm of endothelial cells?	SELECT COUNT(DISTINCT c1.cell_id) FROM cells c1 JOIN cells c2 ON ST_DWithin(c1.geom, c2.geom, 100) WHERE c1.spatial_domain = '3' AND c1.cluster = 'Stromal' AND c2.cluster = 'Endothelial'
κ_3 Spatial + Bio Filter	Find CD8+_T_Cells co-expressing CD3D and GZMB cytotoxic markers.	SELECT c.cell_id, e1.raw_count AS cd3d_count, e2.raw_count AS gzmb_count FROM cells c JOIN expression e1 ON e1.cell_id = c.cell_id AND e1.gene_name = 'CD3D' JOIN expression e2 ON e2.cell_id = c.cell_id AND e2.gene_name = 'GZMB' WHERE c.cluster = 'CD8+_T_Cells' AND e1.raw_count > 0 AND e2.raw_count > 0
κ_4 Neighborhood Aggregation	Which Stromal_&_T_Cell_Hybrid cells have more than 5 CD8+_T_Cell neighbors within 100 μm?	SELECT c1.cell_id, COUNT(c2.cell_id) AS cd8_count FROM cells c1 JOIN cells c2 ON ST_DWithin(c1.geom, c2.geom, 100) AND c1.cell_id != c2.cell_id WHERE c1.cluster = 'Stromal_&_T_Cell_Hybrid' AND c2.cluster = 'CD8+_T_Cells' GROUP BY c1.cell_id HAVING COUNT(c2.cell_id) > 5
κ_5 Region Statistics	What is the average cell area for IRF7+_DCs in each tissue region?	SELECT r.region_id, AVG(c.cell_area) AS avg_cell_area FROM regions r LEFT JOIN cells c ON ST_Contains(r.geom, c.geom) AND c.cluster = 'IRF7+_DCs' GROUP BY r.region_id ORDER BY avg_cell_area DESC
κ_6 Compositional	For each cell cluster in domain 3, calculate the average distance to the nearest stromal cell.	SELECT cluster, AVG(nearest_dist) FROM (SELECT c1.cell_id, c1.cluster, MIN(ST_Distance(c1.geom, c2.geom)) AS nearest_dist FROM cells c1 CROSS JOIN cells c2 WHERE c2.cluster = 'Stromal' AND c1.spatial_domain = '3' AND c1.cell_id != c2.cell_id GROUP BY c1.cell_id, c1.cluster) q GROUP BY cluster ORDER BY AVG(nearest_dist) ASC

Table 3: Representative NL2SQL pairs from SPaRTA-Bench , one per query category κ_j. All examples passed SQL execution.

#	Name	Question	Erroneous SQL + Explanation	Type
1	Self-match inflation	Profile the cellular neighborhood of Myoepi_ACTA2+ cells within 70 μm.	FROM cells c1 JOIN cells c2 ON ST_DWithin(c1.geom, c2.geom, 70) WHERE c2.cluster='Myoepi_ACTA2+' *Missing* c1.cell_id != c2.cell_id *— each cell counts itself as its own neighbor, inflating all proximity counts.*	Bug
2	Zero-excluding average	For each CD8+_T_Cell, what is the average GZMB expression in neighboring Invasive_Tumor cells within 50 μm?	JOIN expression e ON ... SELECT AVG(e.raw_count) *Inner* JOIN *drops neighbors with no* GZMB *entry under sparse storage, excluding zero-expressors from the denominator and inflating the average.*	Bug
3	Threshold on inflated average	List stromal cells with average EPCAM expression > 3 among neighboring DCIS_1 cells within 60 μm.	JOIN expression e ON ... HAVING AVG(e.raw_count) > 3 *Most impactful bug: inflated average from sparse storage causes wrong cells to pass the threshold, directly controlling which cells are returned.*	Bug
4	Zero-excluding median	Calculate median ERBB2 expression among neighboring Stromal cells for each DCIS_1 cell within 40 μm.	PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY e.raw_count) *Zeros excluded from ordered set; median substantially overestimated. Requires* COALESCE(e.raw_count, 0) *inside* PERCENTILE_CONT *with* LEFT JOIN.	Bug
5	Unindexed proximity filter	How many B_Cells in domain 2 have CD8+_T_Cell neighbors within 40 μm?	JOIN cells c2 ON ST_Distance(c1.geom, c2.geom) < 40 *Bypasses* GIST *spatial index; computes exact distances for all cell pairs before filtering. Orders of magnitude slower than* ST_DWithin(c1.geom, c2.geom, 40).	Perf.
6	Anchor fan-out inflation	Which B_Cells are located within 50 μm of any DCIS_1 cell?	SELECT DISTINCT c1.cell_id, ST_Distance(...) AS distance_um ORDER BY distance_um ASC *A* B_Cell *matching N* DCIS_1 *neighbors produces N rows;* DISTINCT *on* (cell_id, distance) *does not deduplicate — each neighbor yields a different distance.*	Bug
7	Negated outer join	How many Macrophages_1 cells in domain 2 express CD68 above background?	LEFT JOIN expression e ON ... WHERE e.raw_count > 0 WHERE *on the joined column negates the outer join — logically equivalent to inner join. Cells with no* CD68 *entry are incorrectly dropped. Use* JOIN *directly.*	Bug
8	Redundant outer join	Find DCIS_1 cells with ERBB2 > 3 that have at least one Endothelial neighbor within 40 μm.	LEFT JOIN cells c2 ON ... HAVING COUNT(DISTINCT c2.cell_id) > 0 HAVING COUNT > 0 *filters out all zero-match rows, making the* LEFT JOIN *equivalent to inner join. Contradictory pattern; use* JOIN *directly.*	Bug
9	MAX over expressors (correct)	For B_Cells in domain 2, what is the maximum CD8A expression among neighboring CD8+_T_Cells within 25 μm?	JOIN expression e ON ... SELECT MAX(e.raw_count) *Correct as-is. Maximum over expressing cells is the biologically meaningful quantity; including zero-expressors would only lower the maximum artificially.*	N/A
11	Window-aggregate conflict	Identify spatial clusters of Macrophages_1 in domains 1–2 with ≥ 3 cells ($\varepsilon = 20$ μm). Report cluster size, mean nucleus area, and mean CD68 expression.	WITH clustered AS (SELECT ST_ClusterDBSCAN(...) OVER () ...) SELECT ... GROUP BY spatial_cluster_id, spatial_domain ST_ClusterDBSCAN *is a window function computed in a CTE; the outer* GROUP BY *then groups by the assigned cluster ID, which is valid. Flag only when window function and* GROUP BY *are in the same query level.*	Bug
12	Phantom gene join	Find proliferating invasive tumor cells in domain 2.	WHERE c.cluster = 'Prolif_Invasive_Tumor' AND e.gene_name = 'MKI67' AND e.raw_count > 0 MKI67 *not mentioned in question. Cluster label already encodes proliferation; unrequested join silently drops cells absent from sparse table.*	Faith.

Continued on next page

(continued)

#	Name	Question	Erroneous SQL + Explanation	Type
13	Invented expression cutoff	Identify cells expressing *high* ESR1 with *high* ERBB2 co-expression.	HAVING MAX(...) > 10 *"High" is qualitative;* > 10 *is invented and biologically arbitrary. Correct approach:* PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY raw_count).	Faith.
14	Fraction–percentage confusion	What *fraction* of B_Cells are located in spatial domain 0?	SELECT ROUND(100.0 * COUNT(...) / COUNT(*), 2) *NL asks for fraction* $\in [0, 1]$ *but SQL returns percentage* $\in [0, 100]$ *via multiplication by* 100.0.	Faith.
15	Population scope error	Rank spatial domains by the fraction of CD8+_T_Cells expressing GZMB that reside within 100 μm of proliferating tumors.	COUNT(DISTINCT ng.cell_id) / NULLIF(total_cd8 , 0) total_cd8 *counts all* CD8+_T_Cells, *not* GZMB-*expressing only. Asymmetric understatement can reverse domain ranking and clinical conclusion.*	Faith.
16	Density qualifier omission	Identify B_Cells expressing elevated CD79A proximal to CD8+_T_Cell *clusters* within 50 μm.	EXISTS (SELECT 1 FROM cells c2 WHERE c2.cluster = 'CD8+_T_Cells' AND ST_DWithin(. . . , 50)) *"Clusters" requires density condition ($\geq N$ cells within radius r); checking proximity to any single cell is insufficient.*	Sem.
17	Partial category mapping	Find DCIS cells within 120 μm of infiltrating immune cells.	WHERE c1.cluster = 'DCIS_1' AND c2.cluster IN ('CD8+_T_Cells', 'CD4+_T_Cells',. . .) DCIS_2 *omitted from anchor;* B_Cells, DCs, Mast_Cells *omitted from immune set.*	Sem.
18	Underspecified filter scope	Identify immune cells within 100 μm of tumor cells *in domain* 3.	EXISTS (. . . WHERE . . . AND c2.spatial_domain = '3') *Domain filter applied to tumor cells only; ambiguous whether immune cells should also be restricted to domain 3. Both interpretations valid — flag for clarification.*	Sem.
19	Invented entity identifier	List all B_Cells inside the tumor microenvironment region.	JOIN regions r ON r.region_id = 'R002' R002 *is invented; valid* region_id *values are* 0–6. *Should filter by* region_type *semantic attribute instead.*	Sem.

Table 4: Complete taxonomy of 18 NL2SQL failure modes identified in SPaRTA. Each row shows the natural language question, the erroneous SQL fragment (abbreviated), a brief explanation in italics, and the failure type. Types: *Bug* = incorrect result; *Perf.* = correct but inefficient; *Faith.* = SQL answers a different question than asked; *Sem.* = biological domain knowledge misapplied; *N/A* = pattern is correct.

C Model Version Details

For our experiments, we use the following model snapshots.

- `GPT`: gpt-5.5-2026-04-23
- `Claude`: claude-sonnet-4-6
- `Gemini`: gemini-3.5-flash
- `Grok`: grok-4.3
- `DeepSeek`: deepseek-v4-flash
- `Mistral (proprietary)`: mistral-large-2512

For open-weight models used in fine-tuning experiments:

- `LLaMA`: Llama-3.1-8B
- `Mistral (open)`: Mistral-7B-Instruct-v0.3
- `Gemma`: gemma-4-E4B-it
- `Qwen`: Qwen2.5-7B-instruct
- `GPT-oss`: gpt-oss-20B

D Results

E Hyperparameter Details

F Schema Prompt

All five models were fine-tuned using LoRA (Hu et al., 2022) on a single NVIDIA RTX A6000 (48 GB).

We apply LoRA with rank $r = 16$, $\alpha = 32$, and dropout $p = 0.05$. For Llama-3.1-8B, Mistral-7B, Qwen3.5-9B, and Gemma-4-4B, adapters target all linear layers (query, key, value, output, gate, up, and down projections). For GPT-oss-20B, whose weights use a proprietary MX floating-point format, adapters are restricted to the four attention projections (Q/K/V/O) under bfloat16 dequantization, yielding ≈8M trainable parameters (0.04% of total weights) versus ≈42M for the other models.

All models are trained for 3 epochs using the TRL `SFTTrainer` with a cosine learning-rate schedule, peak learning rate 2×10^{-4}, 20 warmup steps, effective batch size 16, and a maximum sequence length of 2,048 tokens. All models used `bfloat16` precision and fused AdamW. Per-device batch size and gradient accumulation steps were adjusted per model to maintain the effective batch size of 16 within the 48 GB memory budget. GPT-oss-20B required MxFP4 weight dequantization to fit on a single GPU; Qwen2.5-7B was trained with thinking mode disabled.

G Computing Environment

All experiments (API calls and data processing) are done using Google Colab Pro+ accounts.

Additional open model experiments are done in a research cluster consisting of 4* NVIDIA RTX A6000 (48GB) GPU nodes. These took us around 500 A6000 hours to complete. We used popular libraries including *PyTorch, HuggingFace Transformers, and accelerate*.

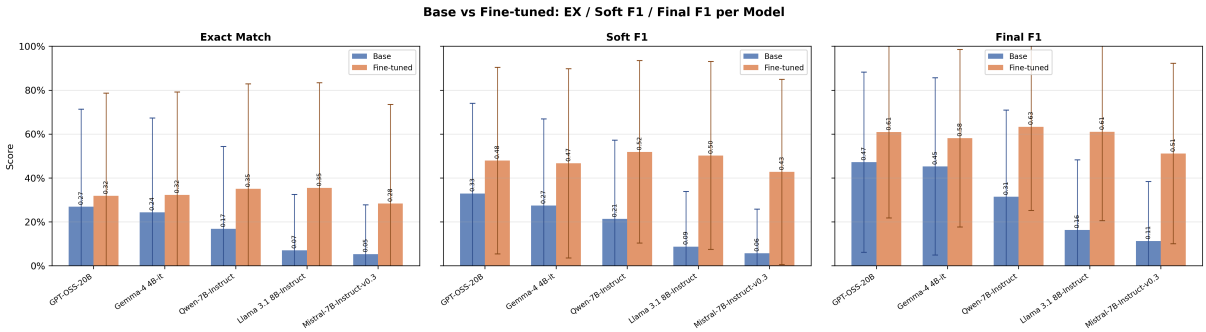

Figure 5: Comparison of base and fine-tuned open-weight language models on the SPaRTA benchmark across Exact Match (EA), Soft-F1, and Final-F1 metrics.

Model	Failure (%)	EA	Soft-F1	Final-F1
GPT-4o	0.6	0.3399	0.4437	**0.6028**
Gemini 2.5 Flash	6.9	0.3212	0.4035	0.5804
Claude Sonnet 4.6	2.8	0.3005	**0.4692**	0.5746
DeepSeek-V4 Flash	3.3	0.3222	0.4075	0.5696
Mistral Large 25.12	5.7	0.2897	0.3637	0.5358
GPT-OSS-20B	12.6	0.2690	0.3294	0.4716
Grok-4.3	22.2	0.2453	0.3285	0.4538
Gemma-4 4B-it	9.6	0.2433	0.2743	0.4523
Qwen-7B-Instruct	27.9	0.1685	0.2136	0.3145
Gemma-2 9B-Instruct	37.1	0.1892	0.2109	0.3139
Llama-3.1 8B-Instruct	61.4	0.0700	0.0872	0.1627
Mistral-7B-Instruct-v0.3	69.0	0.0532	0.0572	0.1123

Table 5: Benchmark performance of proprietary and open-weight NL2SQL models on SPaRTA-Bench . **EX** denotes exact-match execution accuracy, while **Soft-F1** and **Final-F1** measure partial overlap between predicted and ground-truth query outputs. Failure (%) reports the fraction of generated queries that failed execution or validation.

```
You are a PostgreSQL/PostGIS SQL expert for biospatial
transcriptomics data.

DATABASE SCHEMA:
- cells: cell_id (VARCHAR, PK), cluster (VARCHAR),
  geom (GEOMETRY Point), cell_area (FLOAT),
  nucleus_area (FLOAT), transcript_counts (BIGINT),
  total_counts (BIGINT), spatial_domain (VARCHAR)
- expression: cell_id (VARCHAR), gene_name (VARCHAR),
  raw_count (FLOAT)
- regions: region_id (VARCHAR), region_type (VARCHAR),
  geom (GEOMETRY Polygon), area_um2 (FLOAT)
- genes: gene_name (VARCHAR), gene_id (VARCHAR),
  feature_type (VARCHAR), n_cells (INT)

cluster values (20): B_Cells, CD4+_T_Cells, CD8+_T_Cells,
  DCIS_1, DCIS_2, Endothelial, Invasive_Tumor,
  Macrophages_1, Macrophages_2, Mast_Cells, Stromal, . . .

Note: All geometry coordinates, distances, and areas are
in micrometers (raw, unscaled).

Return ONLY the SQL query. Do not use markdown or
code fences. Return the SQL query only.
```

Figure 6: System prompt used for all zero-shot and fine-tuning experiments. The schema grounds the model in concrete table definitions; the instruction at the end suppresses markdown formatting and natural language explanation.