CHATSPATIAL: SCHEMA-ENFORCED AGENTIC ORCHESTRATION FOR REPRODUCIBLE SPATIAL TRANSCRIPTOMICS ANALYSIS

Chen Yang
Department of Statistics
Texas A&M University
College Station, TX 77843, USA
cafferychen777@tamu.edu

Xianyang Zhang*
Department of Statistics
Texas A&M University
College Station, TX 77843, USA
zhangxiany@stat.tamu.edu

Jun Chen*
Division of Computational Biology
Department of Quantitative Health Sciences
Mayo Clinic
Rochester, MN 55905, USA
Chen.Jun2@mayo.edu

ABSTRACT

Spatial transcriptomics has transformed our ability to study tissue architecture at molecular resolution. Yet analyzing these data demands navigating 60+ computational methods across incompatible Python and R ecosystems, creating a widening gap between experimental capability and analytical accessibility. LLM-based agents offer a path forward through natural language interaction, but code-generating approaches introduce hallucinated functions, incorrect parameters, and non-reproducible outputs—undermining the reliability that scientific workflows require. We introduce *schema-enforced agentic orchestration*, a paradigm in which the LLM selects from pre-validated tool schemas rather than generating free-form code. Built on the Model Context Protocol (MCP), our approach embeds domain expertise directly into schema descriptions for context-aware parameter inference and seamlessly bridges Python and R through automated data conversion. We instantiate this paradigm in ChatSpatial, a platform orchestrating 60+ methods across 15 analytical categories for spatial transcriptomics. Through replication of two published studies—recovering key biological findings including tumor microenvironment organization and subclonal heterogeneity—and systematic validation across 28 test scenarios spanning four spatial transcriptomics platforms, we demonstrate that schema-enforced orchestration enables reproducible, multi-step discovery through conversation, supporting iterative loops from reasoning to computational experimentation.

1 INTRODUCTION

The explosive growth of spatial transcriptomics—now spanning platforms from 10x Visium to MERFISH, Xenium, and beyond—has outpaced the field's ability to make its computational methods accessible (Arora et al., 2023; Denisenko et al., 2024). A typical study today requires preprocessing, spatial domain identification, cell type deconvolution, cell-cell communication analysis, and trajectory inference, each demanding specialized tools that often exist exclusively in either the Python or R ecosystem. This fragmentation imposes an "implementation tax": biologists with clear scientific questions find themselves debugging cross-platform data conversions rather than pursuing

*Co-corresponding authors.

biological insights, while even computational researchers struggle to reproduce analyses that span multiple tool ecosystems.

LLM-based agents offer a compelling vision for closing this gap: researchers could express complex analytical intents through natural language rather than code. However, current code-generating agents (Lin et al., 2025; Wang et al., 2025a) realize this vision at a steep cost. By dynamically writing executable code, they introduce hallucinated function names, incorrect API calls, and version-dependent syntax errors—failure modes that recent studies show are pervasive in LLM-generated code (Spracklen et al., 2025; Wang et al., 2025b). Interactive copilots (Huang, 2024; Nagarajan et al., 2025) offer more controlled assistance but typically operate within a single ecosystem, leaving researchers to manage cross-platform integration themselves. In scientific computing, a workflow that works "most of the time" is a workflow that cannot be trusted.

Scientific discovery is inherently iterative: researchers alternate between forming hypotheses, running computational analyses, and designing follow-up experiments. In genomics, the computational portion of this loop is increasingly the bottleneck because it is both complex and fragile. Our goal is to make this computational experimentation loop conversational *without* sacrificing trustworthiness, so that agentic systems can accelerate biological exploration while preserving rigorous reproducibility.

We observe that this reliability gap is fundamentally an *interface* problem, not a model problem. Code generation asks the LLM to perform an unconstrained creative act—writing arbitrary programs—where any syntactic or semantic error causes failure. We propose *schema-enforced agentic orchestration*, a paradigm that recasts this as a constrained selection task: the LLM chooses from pre-validated tool schemas where method names are enumerated literals, parameters have validated types, and outputs follow standardized formats. This architectural constraint decouples probabilistic language understanding from deterministic *tool invocation*: once the LLM selects a tool and its parameters, the resulting call is fully specified and schema-validated. Many statistical procedures are deterministic given fixed inputs, while methods with inherent stochasticity can be made practically reproducible by exposing random seeds and recording parameters for replay. The remaining source of variability—the LLM's interpretation of user intent—is bounded by schema constraints and made transparent through auditable parameter choices.

This paper makes three contributions:

1. **A schema-enforced execution model** built on the Model Context Protocol (MCP) (Anthropic, 2024) that transforms LLM-driven analysis from probabilistic code generation to schema-constrained parameter selection, eliminating code-level hallucinations while preserving natural language flexibility.
2. **Knowledge injection via protocol**, embedding domain expertise directly into tool schema descriptions to enable context-aware parameter inference—adapting to different experimental platforms and data characteristics without model fine-tuning.
3. **Seamless cross-ecosystem orchestration** that automatically bridges Python and R environments through data object conversion, removing a persistent barrier in computational biology.

We instantiate these principles in ChatSpatial[1], a platform for spatial transcriptomics that orchestrates 60+ methods across 15 analytical categories. Through replication of two published studies and validation on 28 test scenarios spanning four spatial transcriptomics platforms, we demonstrate that researchers can conduct publication-quality analyses through iterative conversational interaction—posing questions, inspecting results, and refining parameters in a continuous loop between scientific reasoning and computational execution, without writing a single line of code.

2 Schema-Enforced Agentic Orchestration

Our approach rests on a single architectural principle: constrain what the LLM produces, rather than attempting to make unconstrained generation more reliable. We realize this through three mechanisms: schema-enforced parameter selection via the Model Context Protocol, knowledge injection

[1]This paper was submitted under the working title *AgenticST*.

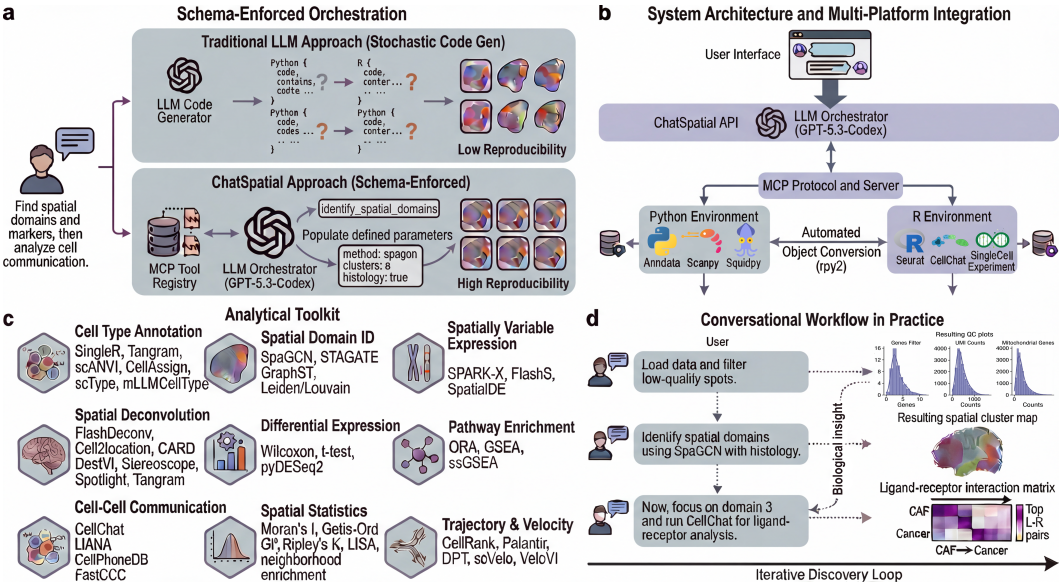

Figure 1: **Overview of ChatSpatial.** **(a)** Schema-enforced orchestration compared with traditional code-generation: the LLM populates pre-validated tool schemas instead of writing arbitrary code, yielding high reproducibility. **(b)** System architecture: user requests pass through the LLM orchestrator to the MCP server, which dispatches execution across Python and R environments with automated data conversion. **(c)** Analytical toolkit spanning 15 categories and 60+ methods. **(d)** Conversational workflow in practice: the researcher iteratively poses questions, inspects results, and refines the analysis through dialogue. The architecture is LLM-agnostic; any MCP-compatible model can serve as the orchestrator.

through schema descriptions, and transparent cross-ecosystem dispatch. Together, these transform the LLM from a code author into a reliable workflow orchestrator (Figure 1).

2.1 From Code Generation to Parameter Selection

To make this principle concrete, consider how different architectures handle the same user request. Code-generating agents output executable strings—a process where any syntactic or semantic error causes failure. Schema-enforced orchestration instead presents the LLM with a structured form: pre-defined method choices, typed parameter fields, and validated defaults. The shift is from writing a "free-form essay" to completing a "fill-in-the-blank" form.

Consider a user request: *"Deconvolve the cell types using the reference dataset."*

Code generation approach:

```
# LLM generates arbitrary code
cell2location.run_deconvolution(
    adata=spatial_data,
    reference=sc_reference,
    n_epochs=30000  # May hallucinate parameters
)
```

Schema-enforced approach:

```
{
  "tool": "deconvolve_data",
  "parameters": {
    "data_id": "1",
    "method": "cell2location",  // Enum: must be valid
```

```
    "reference_data_id": "2",
    "n_epochs": 30000           // Type-checked integer
  }
}
```

In the schema-enforced model, the LLM cannot invent a method name—it must select from a pre-validated enumeration (e.g., `Literal["rctd", "card", "cell2location", ...]`). This structurally eliminates code-level errors—hallucinated functions, type mismatches, invalid method names—while preserving the LLM's ability to interpret user intent and select appropriate methods.

2.2 Model Context Protocol Architecture

We implement schema-enforced orchestration using the Model Context Protocol (MCP) (Anthropic, 2024), an open standard for LLM-tool integration that provides three key mechanisms:

Tool Registration. Each analytical capability is exposed as an MCP tool with a strict JSON schema defining required and optional parameters, enumerated choices, and type constraints. Tools are registered via decorators that automatically generate schema documentation visible to the LLM.

Schema Validation. Before any tool execution, MCP validates that parameters conform to the schema. Type mismatches, invalid enum values, and missing required parameters are caught at the protocol level, not at runtime.

Capability Negotiation. At session initialization, the server declares its available tools to the client, enabling the LLM to reason about what operations are possible and their requirements.

ChatSpatial exposes 20 high-level MCP tools spanning 15 analytical categories (e.g., `deconvolve_data`, `identify_spatial_domains`) with a `method` parameter that routes to specific algorithm implementations. This provides a manageable interface while supporting 60+ underlying methods. Because MCP is an open protocol supported by multiple LLM clients, ChatSpatial's analytical capabilities are accessible from any MCP-compatible interface without modification, decoupling the scientific tooling from any particular LLM platform.

2.3 Knowledge Injection via Protocol

We embed domain expertise directly into MCP tool schema descriptions, transforming the interface from a passive validation layer into an active cognitive scaffold for the LLM. This addresses a key challenge in scientific tool orchestration: different experimental platforms, data characteristics, and analysis goals demand different parameter configurations, yet traditional approaches rely on either rigid rule engines (which lack flexibility) or model fine-tuning (which requires retraining).

For example, the `n_neighbors` parameter for spatial graph construction includes:

> *"For 10x Visium (hexagonal grid), recommended k=6 for immediate neighbors; for single-cell resolution platforms like MERFISH, recommended k=10–15 based on cell density; for Xenium, consider Delaunay triangulation."*

The LLM performs *in-context inference*: upon detecting MERFISH data characteristics, it selects `n_neighbors=10` based on the schema documentation, without hallucinating arbitrary values. This decouples domain knowledge from model weights, enabling updates through schema modification rather than retraining, while keeping reasoning transparent and auditable.

2.4 Cross-Ecosystem Orchestration

ChatSpatial transparently bridges Python and R execution environments, enabling researchers to invoke methods from either ecosystem through a unified conversational interface. This addresses a persistent barrier: critical methods such as SpaGCN (Python) for spatial domains and CellChat (R) for cell communication exist in only one ecosystem, forcing researchers to write custom conversion scripts. Our dispatch layer detects when a requested algorithm requires R, automatically converts Python `AnnData` objects to R-compatible formats via `rpy2`, executes the R function, and translates

results back to Python within the unified session state. A request like *"Analyze cell communication using CellChat"* triggers R execution invisibly, eliminating the "integration tax" of mastering both ecosystems.

3 EXPERIMENTAL VALIDATION

We validate ChatSpatial through two complementary strategies: replication of published studies to demonstrate that schema-enforced orchestration yields genuine biological insights, and systematic testing across diverse scenarios to confirm robustness. In both cases, the researcher interacts entirely through conversation—posing questions, inspecting intermediate results, and refining the analysis through dialogue. All experiments used Claude Sonnet 4.5 (`claude-sonnet-4-5-20250929`) as the orchestrating LLM with default client sampling settings. Computational analyses were executed on a single workstation (Apple M-series, 36 GB RAM, macOS); R-based methods used `rpy2` 3.5 with R 4.3.

3.1 REPLICATION OF PUBLISHED STUDIES

To demonstrate that conversational interaction alone can yield publication-quality biological insights, we replicated two recent spatial transcriptomics studies without any manual code intervention.

OSCC Tumor Architecture (Arora et al., 2023). We replicated the spatial analysis of oral squamous cell carcinoma across 12 patient samples. Through natural language commands, ChatSpatial executed a multi-step workflow: multi-sample integration (Harmony (Korsunsky et al., 2019)), spatial domain identification (Louvain clustering (Blondel et al., 2008)), cell type deconvolution (FlashDeconv (Yang et al., 2025); estimating cell type proportions per spatial spot using a single-cell reference), and cell-cell communication analysis (CellChat (Jin et al., 2021); inferring intercellular signaling from ligand-receptor co-expression). The agent recovered the study's key biological findings: three spatially distinct tumor states—Tumor Core (TC), Leading Edge (LE), and a Transitory state—with characteristic marker genes (Figure 2a,b), robust enrichment of macrophages (11/12 samples, 92%) and fibroblasts/CAFs (10/12, 83%) at the tumor leading edge (Figure 2d,e), and extracellular matrix interaction pathways (COL1A1-SDC1, FN1-SDC1) mediating tumor-stroma communication at the invasive front (Figure 2f,g).

HGSOC Subclone Analysis (Denisenko et al., 2024). We replicated the subclonal heterogeneity analysis of high-grade serous ovarian carcinoma across 8 patients spanning different chemotherapy response categories. The workflow comprised RCTD-based deconvolution (Cable et al., 2022) (a reference-informed method for estimating cell type mixtures), inferCNV-based copy number inference (Patel et al., 2014) (detecting chromosomal gains and losses indicative of tumor subclones), and cross-patient comparison of microenvironment composition. RCTD deconvolution revealed heterogeneous tumor fractions across patients (Figure 3a), and inferCNV confirmed that 5 of 8 patients harbored multiple spatially distinct subclones (Figure 3b). Cell type composition analysis showed variable immune infiltration patterns across chemotherapy response groups (Figure 3c), while tumor spot percentage and mean CNV score exhibited strong concordance (Figure 3d), cross-validating deconvolution and CNV-based estimates. Spatial autocorrelation analysis revealed that cell type markers ranked among the top spatially variable genes, while CNV markers were dispersed across the distribution (Figure 3e), with no significant enrichment relative to background (Mann–Whitney $p = 0.07$; Figure 3f)—suggesting that spatial organization in this sample is driven more by microenvironment composition than by subclonal identity, an exploratory observation consistent with the original study's findings.

Both replications were completed through multi-turn conversational interaction, with the researcher guiding the analysis through iterative dialogue—inspecting intermediate outputs, selecting methods, and refining parameters at each step. This demonstrates that schema-enforced orchestration closes the loop between scientific reasoning and computational execution, enabling publication-quality discovery without manual coding.

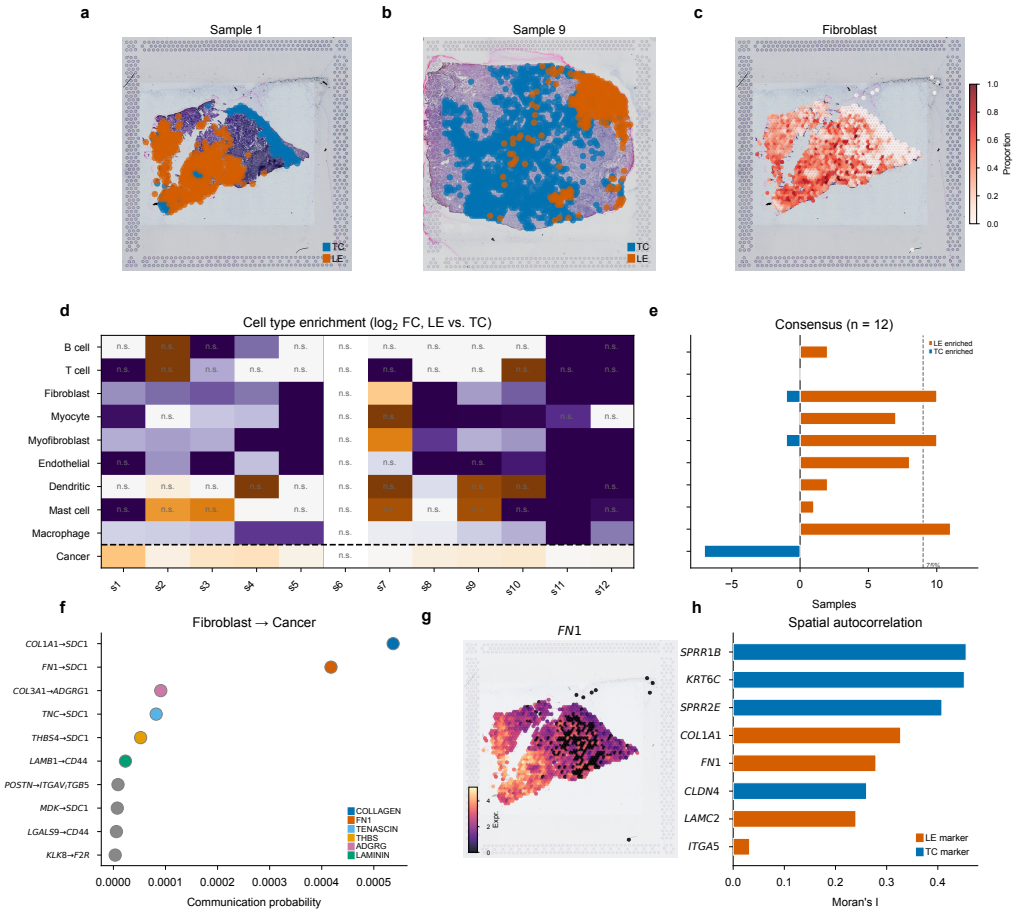

Figure 2: **Replication of OSCC tumor architecture (Arora et al., 2023).** **(a,b)** Spatial domain maps for two representative samples showing tumor core (TC, orange) and leading edge (LE, blue) regions identified via Louvain clustering. **(c)** FlashDeconv deconvolution showing fibroblast proportion across spatial spots, with enrichment at the tumor–stroma interface. **(d)** Per-sample cell type enrichment ($\log_2$ fold change, LE vs. TC) across 12 samples. "n.s." indicates non-significant enrichment. **(e)** Consensus enrichment across all 12 samples; macrophages (11/12, 92%) and fibroblasts (10/12, 83%) show consistent LE enrichment (orange), while cancer cells are TC-enriched (blue). **(f)** CellChat-inferred fibroblast→cancer communication probabilities, highlighting collagen (COL1A1-SDC1) and fibronectin (FN1-SDC1) pathways. **(g)** Spatial expression of *FN1* at the invasive front. **(h)** Spatial autocorrelation (Moran's I) of LE and TC marker genes, confirming spatially coherent expression patterns.

3.2 Systematic Validation

Beyond published study replication, we evaluated ChatSpatial across 28 comprehensive test scenarios covering four dimensions:

- **Data handling**: Standard Visium, high-resolution Xenium (150K cells), low-quality filtering, multi-sample integration, missing coordinate handling
- **Core analyses**: Spatial domain identification, variable gene detection, deconvolution, cell annotation, trajectory analysis, RNA velocity
- **Conversational workflows**: Iterative refinement, ambiguous command handling, parameter override, cross-category pipelines, error recovery
- **Scalability**: Large datasets (100K+ spots), sparse data (>99.9% sparsity), noisy references, long conversation contexts

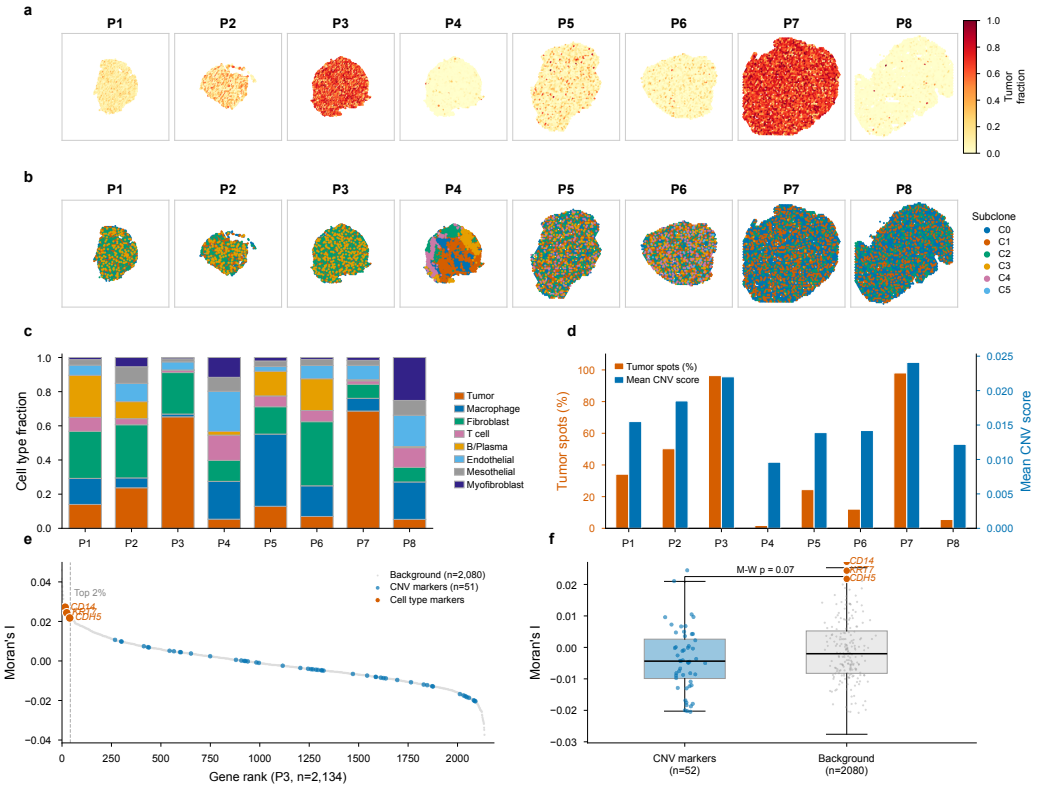

Figure 3: **Replication of HGSOC subclonal heterogeneity (Denisenko et al., 2024).** **(a)** RCTD-deconvolved tumor fraction across 8 patients (P1–P8). **(b)** InferCNV-derived spatial maps of copy number subclones, with 5 of 8 patients harboring multiple spatially distinct subclones. **(c)** Cell type composition per patient from RCTD deconvolution. **(d)** Tumor spot percentage (orange) and mean CNV score (blue) per patient, showing concordance between deconvolution and CNV-based estimates. **(e)** Gene ranking by Moran's I for Patient 3; cell type markers (orange dots) rank among the top 2% of spatially variable genes, while CNV markers (blue dots) are dispersed across the distribution. **(f)** Comparison of Moran's I between CNV marker genes and background genes (Mann–Whitney $p = 0.07$; not significant), suggesting that spatial organization in this patient is shaped more by microenvironment composition than by subclonal identity.

All 28 scenarios executed successfully through conversation alone, validated across four spatial transcriptomics platforms—Visium, Xenium, MERFISH, and STARmap—covering both sequencing-based and imaging-based modalities (up to 930K+ cells). We considered a scenario successful if the end-to-end workflow completed without schema violations or tool runtime failures and produced the expected analysis artifacts (e.g., clustering labels, marker tables, communication networks, or visualizations), which are recorded via exported results and provenance metadata.

4 Related Work

Table 1 situates schema-enforced orchestration relative to existing paradigms. STAgent (Lin et al., 2025) and SpatialAgent (Wang et al., 2025a) apply the code-generation paradigm specifically to spatial transcriptomics, producing Python scripts from natural language queries. While both systems achieve substantial analytical coverage, execution success depends on the LLM generating syntactically and semantically correct programs for every invocation—a guarantee our schema-enforced approach provides by construction. QuST-LLM (Huang, 2024) and scAssist (Nagarajan et al., 2025) take a more controlled approach within individual programming environments, but neither supports

Table 1: Comparison of LLM-driven paradigms for scientific computing. Schema-enforced orchestration trades unconstrained flexibility for schema-validated execution and auditable replay—a favorable exchange in domains where reliability is paramount.

Property	Code Generation	Interactive Copilot	Schema-Enforced (Ours)
Reproducibility	Variable	High	**Auditable & replayable**
Hallucination risk	High	Medium	**Code-level eliminated**
Cross-ecosystem	Manual	Limited	**Automatic**
Domain knowledge	In model weights	Rule-based	**In schema (updatable)**
User interaction	Post-hoc debugging	Task-specific	**Conversational loop**
Flexibility	Unconstrained	Moderate	Bounded by schema

the cross-ecosystem workflows or iterative multi-step analyses that comprehensive spatial transcriptomics studies demand.

More broadly, our work builds on the growing trend toward structured tool use in LLM systems, where models invoke pre-defined APIs rather than generating arbitrary code. The Model Context Protocol (MCP) (Anthropic, 2024) provides the open infrastructure for such integration; our contribution is demonstrating that this infrastructure, when combined with domain knowledge injection and cross-ecosystem dispatch, enables reliable agentic orchestration for a complex scientific domain.

5 Limitations and Future Work

ChatSpatial's capabilities are bounded by its pre-integrated tool inventory: analyses requiring methods outside the current schema cannot be performed through conversation. While schema enforcement eliminates code-level hallucinations, the LLM may still misinterpret ambiguous user intents or select suboptimal methods for specific data characteristics. Our current validation is functional (pass/fail across scenarios) rather than statistical; quantifying tool-invocation consistency across repeated runs and head-to-head comparison with code-generating agents on identical tasks are important next steps. The current implementation focuses on spatial transcriptomics; extending to other omics domains requires integrating additional tool ecosystems. Finally, our validation uses a single LLM (Claude Sonnet 4.5); evaluating generalization across models remains future work.

We envision several extensions: automated schema generation from API documentation to accelerate tool integration, multi-modal interaction incorporating experimental images, and formal verification of workflow reproducibility. More broadly, the schema-enforced paradigm is domain-agnostic—any field with established computational tools and reproducibility requirements could benefit from this approach.

6 Conclusion

We have demonstrated that constraining LLM outputs to validated tool schemas—rather than allowing free-form code generation—enables reproducible, multi-step scientific workflows through conversational interaction. Through ChatSpatial, researchers replicated published spatial transcriptomics findings and executed complex cross-ecosystem analyses without writing a single line of code.

The broader implication is that reliable AI-driven genomics discovery does not require maximizing agent autonomy—it requires carefully designing the interface between probabilistic language understanding and deterministic computational execution. Schema-enforced orchestration provides this interface: by keeping the researcher in the conversational loop while enforcing schema-valid execution, it closes the gap between scientific reasoning and computational experimentation, establishing a foundation for trustworthy AI systems that accelerate discovery without compromising reproducibility.

ACKNOWLEDGMENTS

We thank Dr. Jade Wang (Texas A&M University) for valuable feedback on the case study analyses and Jiahao Mai (Southern Medical University) for assistance with figure design. This work was supported by NIH R01GM144351 (J.C. & X.Z.), NSF DMS1830392, DMS2113359, DMS1811747 (X.Z.), NSF DMS2113360, and the Mayo Clinic Center for Individualized Medicine (J.C.).

REPRODUCIBILITY STATEMENT

ChatSpatial's code is publicly available at `https://github.com/cafferychen777/ChatSpatial` under an open-source license. All validation datasets are publicly available (GEO accessions: GSE208253, GSE211956, GSE103322). For each analysis step, ChatSpatial stores scientifically relevant provenance in the dataset object (selected method, parameterization, and result keys) and exports result tables with a machine-readable index to a local results directory. Datasets can be snapshotted to a persistent cache for replay and external inspection. Some MCP clients additionally persist local conversation history and tool state, enabling practical resumption of multi-turn workflows; ChatSpatial itself remains client- and model-agnostic, and any remaining variability arises from client-side sampling and inherently stochastic algorithms.

REFERENCES

Anthropic. Model context protocol, 2024. URL `https://modelcontextprotocol.io/`. Accessed: 2025-10-30.

Rohit Arora, Christian Cao, Mehul Kumar, Sarthak Sinha, Ayan Chanda, Reid McNeil, Divya Samuel, Rahul K. Arora, T. Wayne Matthews, Shamir Chandarana, Robert Hart, Joseph C. Dort, Jeff Biernaskie, Paola Neri, Martin D. Hyrcza, and Pinaki Bose. Spatial transcriptomics reveals distinct and conserved tumor core and edge architectures that predict survival and targeted therapy response. *Nat. Commun.*, 14:5029, 2023. doi: 10.1038/s41467-023-40271-4.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008(10):P10008, 2008. doi: 10.1088/1742-5468/2008/10/P10008.

Dylan M Cable, Evan Murray, Luli S Zou, and et al. Robust decomposition of cell type mixtures in spatial transcriptomics. *Nat. Biotechnol.*, 40(4):517–526, 2022. doi: 10.1038/s41587-021-00830-w.

Elena Denisenko, Leanne de Kock, Adeline Tan, et al. Spatial transcriptomics reveals discrete tumour microenvironments and autocrine loops within ovarian cancer subclones. *Nat. Commun.*, 15:2860, 2024. doi: 10.1038/s41467-024-47271-y.

Chao Hui Huang. Qust-llm: Integrating large language models for comprehensive spatial transcriptomics analysis. *arXiv preprint arXiv:2406.14307*, 2024. doi: 10.48550/arXiv.2406.14307.

Suoqin Jin, Christian F Guerrero-Juarez, Lihua Zhang, and et al. Inference and analysis of cell-cell communication using cellchat. *Nat. Commun.*, 12(1):1088, 2021. doi: 10.1038/s41467-021-21246-9.

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nat. Methods*, 16(12):1289–1296, 2019. doi: 10.1038/s41592-019-0619-0.

Zuwan Lin, Wenbo Wang, Arnau Marin-Llobet, Qiang Li, Samuel D Pollock, Xin Sui, Almir Aljovic, Jaeyong Lee, Jongmin Baek, Ningyue Liang, Xinhe Zhang, Connie Kangni Wang, Jiahao Huang, Mai Liu, Zihan Gao, Hao Sheng, Jin Du, Stephen J Lee, Brandon Wang, Yichun He, Jie Ding, Xiao Wang, Juan R Alvarez-Dominguez, and Jia Liu. Spatial transcriptomics ai agent charts hpsc-pancreas maturation in vivo. *bioRxiv preprint 2025.04.01.646731*, 2025. doi: 10.1101/2025.04.01.646731.

Vijayaraj Nagarajan, Guangpu Shi, Samyuktha Arunkumar, Chunhong Liu, Jaanam Gopalakrishnan, Pulak R Nath, Junseok Jang, and Rachel R Caspi. SCassist: An AI Based Workflow Assistant for Single-Cell Analysis. *Bioinformatics*, 2025. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaf402.

Anoop P Patel, Itay Tirosh, John J Trombetta, Alex K Shalek, Shawn M Gillespie, Hiroaki Wakimoto, Daniel P Cahill, Brian V Nahed, William T Curry, Robert L Martuza, David N Louis, Orit Rozenblatt-Rosen, Mario L Suvà, Aviv Regev, and Bradley E Bernstein. Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401, 2014. doi: 10.1126/science.1254257.

Joseph Spracklen, Raveen Wijewickrama, A H M Nazmus Sakib, Anindya Maiti, Bimal Viswanath, and Murtuza Jadliwala. We have a package for you! a comprehensive analysis of package hallucinations by code generating llms. In *Proc. USENIX Security Symposium*, 2025.

Hanchen Wang, Yichun He, Paula P. Coelho, Matthew Bucci, Abbas Nazir, Bob Chen, Linh Trinh, Serena Zhang, Kexin Huang, Vineethkrishna Chandrasekar, Douglas C. Chung, Minsheng Hao, Ana Carolina Leote, Yongju Lee, Bo Li, Tianyu Liu, Jin Liu, Romain Lopez, Tawaun Lucas, Mingyu Ma, Nikita Makarov, Lisa McGinnis, Linna Peng, Stephen Ra, Gabriele Scalia, Avtar Singh, Liming Tao, Masatoshi Uehara, Chenyu Wang, Runmin Wei, Ryan Copping, Orit Rozenblatt-Rosen, Jure Leskovec, and Aviv Regev. Spatialagent: An autonomous ai agent for spatial biology. *bioRxiv*, 2025a. doi: 10.1101/2025.04.03.646459.

Zhijie Wang, Zijie Zhou, Da Song, Yuheng Huang, Shengmai Chen, Lei Ma, and Tianyi Zhang. Towards understanding the characteristics of code generation errors made by large language models. In *Proc. IEEE/ACM International Conference on Software Engineering (ICSE)*, pp. 2587–2599, 2025b. doi: 10.1109/ICSE55347.2025.00180.

Chen Yang, Xianyang Zhang, and Jun Chen. Flashdeconv enables atlas-scale, multi-resolution spatial deconvolution via structure-preserving sketching. *bioRxiv*, 2025. doi: 10.64898/2025.12.22.696108. Preprint.