VERIFIER-GUIDED SKILL EVOLUTION FOR AUTONOMOUS SPATIAL OMICS ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatial omics platforms now pair histopathology images with spatially resolved transcriptomes at scale, yet turning such data into biological conclusions still depends on expert judgment at nearly every step. Recent large language model agents can invoke bioinformatics tools from natural language, but treat each dataset as an isolated episode and discard whatever they learn. We argue the missing ingredient is not a better planner but a usable reward: without a signal separating a biologically sound output from a merely runnable one, an agent has nothing to learn from. We present BIOSKILL-EVOLVER, an agent that stores each analysis capability as an executable skill annotated with explicit preconditions and postconditions, moving composability checking from the language model into a symbolic test, and scores every execution with a multi-granularity verifier combining statistical, marker, pathway and spatial checks with a language-model judge. Because this verifier turns an unlabeled analysis into a scalar, it can drive an evolution loop that rewrites failing skills from their own error traces and splits high-variance skills into platform-specific variants under a conservative acceptance rule. Across 38 end-to-end episodes on public Visium, Xenium and Stereo-seq sections, BIOSKILL-EVOLVER raises autonomous task success from 0.079 to 0.500 and the verification score from 0.311 to 0.463 relative to a stateless agent with an identical backbone, while the skill library grows from 19 to 22 skills through three platform specializations. Ablations localize the effect to the evolution loop and the verifier rather than to perception or planning, and the ranking is preserved for three of four alternative backbones. We report this as a single-run, local-benchmark study and state precisely which design mechanisms it exercised.

1 INTRODUCTION

Spatial omics has become a data-rich field with a human-throughput bottleneck. Platforms such as 10x Visium, Xenium, NanoString CosMx and BGI Stereo-seq routinely deliver paired histopathology images and spatially resolved expression for entire tissue sections, but converting a section into a biological conclusion still requires an expert to choose a normalization, pick a clustering resolution, decide which markers define a domain, and judge whether the resulting map is anatomically plausible. Each choice is platform- and tissue-dependent, and none is captured by the tools themselves.

Large language model agents are an attractive response, because they can read an instruction, select a tool and write the glue code. The agents proposed so far for this domain, however, are episodic: they plan, execute and then forget (Yiyao et al., 2025; Wei et al., 2025). An agent that has annotated a hundred Xenium breast sections begins the hundred-and-first with the knowledge it had at the start. The obvious fix, keep a library of reusable skills, as open-ended embodied agents do (Wang et al., 2023), runs into a difficulty specific to biology. In Minecraft, success is checkable: the item is in the inventory or it is not. In spatial omics the analysis almost always runs to completion, and the interesting failure is a pipeline that produces a clean-looking cluster map whose domains contradict their own marker genes. Execution success is therefore nearly uninformative, and without a usable reward an agent has nothing to learn from.

We take this as the central design problem, and it organizes the system. BIOSKILL-EVOLVER stores each analysis capability as an executable skill annotated with preconditions and postconditions, so that whether two skills compose is decided by a symbolic test rather than by the language model. It scores

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

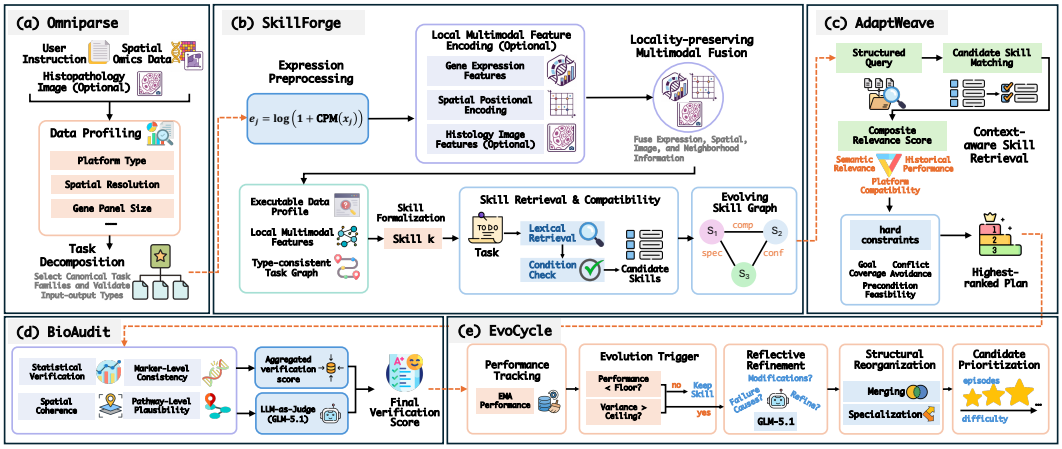


Figure 1: **BIOSKILL-EVOLVER.** OMNIPARSE emits a data profile and a typed task graph; ADAPTWEAVE retrieves skills from the SKILLFORGE graph and synthesizes a plan under precondition and conflict constraints; BIOAUDIT scores the executed outputs; EVOCYCLE writes those scores back as skill rewrites and platform specializations.

every execution with a multi-granularity verifier that asks four largely independent questions: does the output meet its quantitative target, do the predicted groups carry the marker genes they should, do they show coherent pathway activity, and do they respect tissue geometry, and folds in a language-model judgment of biological plausibility. The resulting scalar is what makes learning possible: it identifies which skills are failing and in which biological context, and a reflective loop then rewrites those skills from their own error traces or splits them into platform-specific variants, accepting a rewrite only when it does not regress on held-out episodes. Figure 1 shows the architecture.

Our empirical claims are deliberately narrow. On a local benchmark of 38 end-to-end episodes, BIOSKILL-EVOLVER raises autonomous task success from 0.079 to 0.500 over a stateless agent with the same backbone, and ablations attribute almost all of that gain to the evolution loop and the verifier rather than to perception or plan search. The library grows by three skills, all platform specializations, which says something more specific than "the agent learns": most of the improvement comes from *rewriting* existing skills, not acquiring new ones.

Contributions. We (i) formalize spatial omics analysis as lifelong skill learning over a typed skill graph in which composability is a symbolic precondition test rather than a language-model judgment; (ii) propose BIOAUDIT, a multi-granularity verifier that turns an unlabeled analysis into a scalar usable as a learning signal; (iii) close the loop with a conservative reflective rule that rewrites failing skills from their own error traces and specializes high-variance skills per platform; and (iv) report a controlled evaluation with a backbone-matched stateless baseline, module ablations and evolution dynamics, together with an explicit statement of which mechanisms were exercised (Appendix A.2).

2 RELATED WORK

Computational pipelines for spatial omics. Classical workflows compose normalization, reduction and clustering under expert control (Wolf et al., 2018; Palla et al., 2022; Hao et al., 2021; Zhao et al., 2021), and task-specific deep models improve individual stages: domain identification (Hu et al., 2021; Dong & Zhang, 2022; Long et al., 2023; Xu et al., 2024; Singhal et al., 2024; Gui et al., 2024; Zhu et al., 2025a), expression prediction from histology (He et al., 2020; Pang et al., 2021; Zeng et al., 2022; Xie et al., 2023; Chung et al., 2024; Zhu et al., 2025b; Fu et al., 2025) and annotation (Hao et al., 2021; Xu et al., 2021; Domínguez Conde et al., 2022; Biancalani et al., 2021). Each is a fixed capability a human must select, configure and validate; we do not replace them, but learn when to invoke capabilities of this kind and check the result.

LLM agents and self-improving skill libraries. OmicsNavigator orchestrates multiple agents for zero-shot spatial omics analysis (Yiyao et al., 2025) and LLMiniST probes whether language models can identify cellular niches (Wei et al., 2025); both are episodic by design, as are general scaffolds

such as ReAct (Yao et al., 2022) and AutoGen (Wu et al., 2023). We use all four as baselines, placing each domain agent only on the tasks it was designed for. In open-ended environments, Voyager maintains a growing library of executable skills (Wang et al., 2023) and Reflexion and Self-Refine convert verbal feedback into improved behaviour (Shinn et al., 2023; Madaan et al., 2023). These methods rely on an environment that returns a reliable success signal, and the obstacle in spatial omics is precisely that this signal is missing: a pipeline that runs without error is not thereby correct, and ground-truth labels are unavailable at analysis time. BIOSKILL-EVOLVER differs from prior skill-library agents mainly in supplying that missing signal and in making composability a typed constraint rather than a prompt.

3 METHOD

BIOSKILL-EVOLVER represents an analysis episode as a typed plan over a graph of executable skills, scores the resulting outputs with a verifier, and uses that score to revise the graph. This section states each module at the level of detail needed to interpret the experiments; full specifications, prompts and hyperparameters are in Appendix A.1. All planning, code generation, reflection and judging use a single language-model backbone with a fixed decoding configuration (Sec. 4.1).

3.1 OMNIPARSE: PROFILING AND TYPED TASK DECOMPOSITION

Input formats differ sharply across platforms: Visium gives spot-level counts over a genome-wide panel, Xenium and CosMx single-cell resolution over a few hundred genes, and Stereo-seq flexible binning, so the agent must characterize its data before any skill can apply. OMNIPARSE first computes a deterministic *data profile* $\mathbf{d}$ by inspecting the input object (platform, tissue, species, resolution, panel size, unit count, sparsity, and flags for modalities and annotations); the profile is executable, in that downstream modules consume it directly to gate feature extraction, skill applicability and task decomposition. Per-unit features are then assembled without learnable components: CPM+$\log(1+\cdot)$ expression reduced by truncated SVD (Wolf et al., 2018), random Fourier features of the coordinate (Tancik et al., 2020) instead of raw position, an optional morphology block from unit-aligned image patches (handcrafted by default, optionally a frozen pathology encoder (Chen et al., 2024; Lu et al., 2024)), and a spatial-neighbourhood average of the expression embedding, each block z-scored and concatenated. This *locality-preserving fusion* (Appendix A.1, Eq. equation 8) is cheap, reproducible and degrades gracefully when the image pathway is off.

The instruction t and profile $\mathbf{d}$ are mapped to a task graph by *selecting and parameterizing* canonical task families rather than by free-form generation, each subtask being $\tau_m = (\textit{family}_m, \textit{input_type}_m, \textit{output_type}_m, \textit{constraint}_m)$. Selection may be rule-based or language-model-assisted, but any proposal violating a family definition or the typed IO schema is discarded and replaced by a valid canonical configuration, with a deterministic fallback if planning fails; dependency edges follow from type matching, so the resulting DAG $\mathcal{T}_{\mathrm{DAG}}$ is executable by construction.

3.2 SKILLFORGE: TYPED SKILLS AND THE SKILL GRAPH

Representing tools as natural-language descriptions loses the information that governs composition: annotation cannot precede clustering, a spatial graph requires registered coordinates, spot deconvolution is meaningless for molecule-resolved data. SKILLFORGE therefore defines each skill as a tuple

$$k = \big(\textit{id},\ \textit{code},\ \mathbf{c}_{\mathrm{pre}},\ \mathbf{c}_{\mathrm{post}},\ \mathcal{P},\ \mathcal{H},\ v\big), \tag{1}$$

where *id* is a version-tracked identifier encoding role and lineage (e.g. `cluster_v2_xenium`), *code* is the executable implementation, $\mathbf{c}_{\mathrm{pre}}, \mathbf{c}_{\mathrm{post}} \subset \mathcal{C}$ are semantic conditions from a closed controlled vocabulary (`normalized_expr`, `clusters_assigned`, ...; Appendix A.1), $\mathcal{P}$ is a typed parameter schema, $\mathcal{H}$ a bounded history of (context, score) pairs and $v \in \mathbb{N}$ a version counter. Because $\mathcal{C}$ is closed, composability is a set test rather than a prompt, and $\mathcal{H}$ supplies the evidence used by EVOCYCLE.

Skills form a typed directed graph $\mathcal{G}_{\mathrm{skill}} = (\mathcal{V}, \mathcal{E})$ with composition, specialization and conflict edges (Appendix A.1). A composition edge $k_a \to k_b$ records that k_b may follow k_a and is weighted by a

convex combination of structural coverage and empirical reliability,

$$w_{\text{comp}}(k_a, k_b) = \underbrace{\frac{1}{2}\,\frac{|\mathbf{c}^a_{\text{post}} \cap \mathbf{c}^b_{\text{pre}}|}{|\mathbf{c}^b_{\text{pre}}|}}_{\text{condition coverage}} + \frac{1}{2}\underbrace{\bar{r}_{a\to b}}_{\text{historical success}}, \tag{2}$$

where $\bar{r}_{a\to b}$ is an exponential moving average of the fraction of past episodes in which the transition produced a passing score; with no history the weight falls back to coverage alone, so an unobserved but structurally valid transition is neither rewarded nor penalized. Specialization edges preserve a fallback to the general parent when no platform variant matches, and conflict edges forbid mutually exclusive skills (two incompatible normalizations, say) from appearing in one plan. The graph is initialized from a registry of 33 canonical spatial omics skill definitions spanning quality control through report generation; the subset applicable to the evaluated platforms is activated at initialization, and the first recorded checkpoint of the reported run contains 19 active skills (Sec. 4.5).

3.3 AdaptWeave: retrieval and constraint-guided plan synthesis

For each subtask τ_m, AdaptWeave ranks candidate skills with a lightweight symbolic model that avoids dependence on any embedding backbone,

$$\text{score}(k_i \mid \tau_m) = \lambda_1\,\text{rel}(k_i, \tau_m) + \lambda_2\,\text{compat}(k_i, \mathbf{d}) + \lambda_3\,\text{perf}(k_i), \tag{3}$$

where rel is the normalized overlap between the structured query (family, typed IO, constraint keywords) and the skill's metadata tags, compat filters or up-weights skills by the registry's platform annotations against $\mathbf{d}$, and perf is the EMA of past verification scores; the coefficients are fixed a priori at $(0.4, 0.3, 0.3)$ and used unchanged throughout. A plan $\pi = (k_{\pi_1}, \ldots, k_{\pi_L})$ must then satisfy three hard constraints: (C1) every skill's preconditions are covered by the initial conditions plus its predecessors' postconditions, $\mathbf{c}^{\pi_l}_{\text{pre}} \subseteq \mathbf{c}_{\text{init}} \cup \bigcup_{l'<l} \mathbf{c}^{\pi_{l'}}_{\text{post}}$; (C2) the final postconditions cover the goal conditions of every leaf subtask; and (C3) no two selected skills share a conflict edge. Among feasible plans we maximize

$$\pi^\star = \arg\max_{\pi} \sum_{l=1}^{L} \text{score}\big(k_{\pi_l} \mid \tau_{\text{map}(l)}\big) + \eta \sum_{l=1}^{L-1} w_{\text{comp}}\big(k_{\pi_l}, k_{\pi_{l+1}}\big), \tag{4}$$

trading skill quality against transition reliability, by beam search with constraint pruning (B=5, η=0.3); the top-B plans are reranked once by the language model for biological plausibility and the winner is executed.

3.4 BioAudit: multi-granularity verification

Verification is what makes the loop possible, so we describe the scores actually computed rather than an idealized form. V_{stat} is a weighted average of the *raw* task metrics (ARI, NMI, macro-F1 for grouping; PCC, Spearman and a normalized error term for expression) rather than of pass/fail indicators, which keeps the signal graded because the evolution trigger reads its variance. V_{pathway} averages $\mathbb{I}\{\text{FDR}_c < \alpha\} \cdot \text{NES}_c / \max_{c'} \text{NES}_{c'}$ over fGSEA results against MSigDB Hallmark, so a group with no enriched pathway contributes zero. V_{spatial} combines edge homophily, a binarized Moran's I and a cluster-continuity term on a k-nearest-neighbour spatial graph, which is more stable than a single global statistic for categorical outputs. V_{marker} compares each group's top differentially expressed genes against a curated reference marker table: with a the fraction of units receiving a marker-supported label and o the mean overlap with reference marker sets,

$$V_{\text{marker}} = \sqrt{a \cdot o}, \tag{5}$$

penalizing both a system that labels everything with weak evidence and one that labels almost nothing with strong evidence, which matters for reading our annotation results (Sec. 4.2). Weightings and reference sources are in Appendix A.1.

The applicable signals are combined with task-family weights α_j summing to one and mixed with a language-model judge $V_{\text{LLM}} \in [0, 1]$ of biological plausibility; the final score additionally multiplies by the fraction $\kappa \in [0, 1]$ of requested target outputs the episode actually produced,

$$V_{\text{agg}} = \sum\nolimits_j \alpha_j V_j, \qquad V_{\text{final}} = \kappa \cdot \big[\beta V_{\text{agg}} + (1-\beta) V_{\text{LLM}}\big], \quad \beta = 0.7. \tag{6}$$

The coverage factor κ is why V_{final} stays well below its components even when execution succeeds: an episode producing a valid clustering but no marker report is penalized for incompleteness rather than credited for what it did produce (Sec. 4.3).

3.5 EvoCycle: reflective skill evolution

After each episode t the verification score of every invoked skill is appended to its history and its running estimate updated by $\bar{V}_k^{(t)} = \mu \bar{V}_k^{(t-1)} + (1-\mu) V_k^{(t)}$ with μ=0.9. A skill is flagged when it is either persistently weak or unstable,

$$\text{Evolve}(k) = \mathbb{I}\{\}[\bar{V}_k^{(t)} < \theta_{\text{evo}}] \ \vee \ \mathbb{I}\{\}[\text{Var}(V_k^{(t-T:t)}) > \theta_{\text{var}}], \tag{7}$$

with θ_{evo}=0.6, θ_{var}=0.04, T=10. The variance branch is what distinguishes a skill that is simply bad from one that is good on some platforms and bad on others; the two cases call for different repairs.

Reflective refinement. A flagged skill is passed to the backbone with its source, its recent failure cases (inputs, outputs, error messages and sub-threshold scores) and their biological context, under a template requiring a diagnosis, concrete code-level changes and a complete replacement k'. The replacement inherits the lineage of k with v incremented and is tested on held-out episodes from the same distribution as the failures; if $\bar{V}_{k'}$ does not exceed $\bar{V}_k$ within three validation episodes it is discarded. This conservative rule keeps the library non-degrading in expectation and is why it can improve substantially while growing by only three skills. *Specialization.* When a single platform or tissue accounts for at least 80% of a general skill's recent weak cases, the backbone is asked for a context-specific variant from that skill plus those cases, attached to its parent by a specialization edge that preserves the fallback path; this produces the three specializations of Sec. 4.5.

Three further mechanisms belong to the design but were *not* exercised here: merging of near-duplicate skills, a pairwise-similarity diversity regularizer and a difficulty-annealed curriculum scheduler. We specify them in Appendix A.3 and mark them in the specification–implementation table of Appendix A.2, so that no reported number is attributed to a component that did not run.

4 Experiments

We ask whether BioSkill-Evolver matches or surpasses classical pipelines, task-specific deep models and prior LLM agents (Sec. 4.2–4.3), which module is responsible (Sec. 4.4), whether capability accumulates across episodes rather than being memorized (Sec. 4.5), what the failures look like at the level of individual sections (Sec. 4.6), and whether the learned skills transfer and the verifier is usable as a reward (Sec. 4.7).

4.1 Setup

Data and evaluated subsets. All data are public: 10x Visium (including DLPFC (Maynard et al., 2021)), 10x Xenium human breast (Janesick et al., 2023), the legacy ST HER2$^+$ breast cohort (Andersson et al., 2021), cSCC (Ji et al., 2020), two public 10x breast sections, and BGI Stereo-seq mouse organogenesis (Chen et al., 2022). Not every dataset is wired into every table and we report only what was measured: domain identification on Xenium Breast, HER2 Breast, MOSTA Embryo and MOSTA Heart; expression prediction on 12 cSCC sections and 2 10x Breast sections (the HER2 H&E archives were not retrievable locally); annotation on Xenium Breast, HER2 Breast and MOSTA Embryo; and the end-to-end setting on 38 episodes spanning Visium, Xenium and the sequencing-based track. The harness identifier `stomics:*` denotes that sequencing-based track (HER2 ST, MOSTA, cSCC) and is a group label, not a platform claim. Entries marked "–" are not wired locally rather than zero.

Evaluation protocol. We do not use a three-way split, and state the protocol per task. Domain identification and annotation are *unsupervised or zero-shot at inference*: the method processes a full section and its output is scored against the released labels, with predicted clusters aligned to expert domains by Hungarian matching. Expression prediction uses *within-section 80/20 spot splits*: in every section, 80% of spots are available for training and the remaining 20% are scored, with the same split shared by all methods. There is no validation partition, so no baseline's hyperparameters are tuned on held-out data; our expression skill was, however, developed on the same cohorts, a

Table 1: **Spatial domain identification.** ARI/NMI against released domain labels; Mean is the unweighted average of the four columns. – denotes a method not wired locally (BayesSpace and BANKSY are omitted, being unwired on every column). The three LLM/agentic rows are collapsed because their outputs are numerically identical (Sec. 4.2). Best results are **bold**; second best are underlined.

Method	Xenium Breast		HER2 Breast (ST)		Stereo-seq MOSTA Embryo		Stereo-seq MOSTA Heart		Mean	
	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
Classical pipelines										
Scanpy (Leiden)	0.0087	0.0444	0.1394	0.1958	0.3204	0.4786	0.0872	0.0890	0.1389	0.2020
Seurat	-0.0020	0.0168	0.1576	0.2088	0.4143	0.5898	0.0320	0.0357	0.1505	0.2128
Task-specific deep learning										
SpaGCN	-0.0004	0.0134	**0.1802**	0.2455	0.2170	0.3591	0.0361	0.0416	0.1082	0.1649
STAGATE	0.0042	0.0275	0.1677	0.2659	0.3756	0.5430	0.3489	0.3559	0.2241	0.2981
GraphST	0.0117	0.0516	0.1692	0.2611	0.3308	0.5187	0.3260	0.3384	0.2094	0.2925
SEDR	0.0003	0.0149	0.1168	0.2070	0.4212	0.5905	0.0805	0.0993	0.1547	0.2279
GRAS4T	-0.0019	0.0174	0.1725	0.2638	0.3864	0.5480	0.0000	0.0000	0.1393	0.2073
MAEST	0.0131	0.0530	0.1511	0.2259	0.2198	0.3952	0.0706	0.3192	0.1137	0.2483
LLM / agentic systems										
ReAct / LLMiniST / GLM-5.1 (no skills)	0.0087	0.0444	0.1394	0.1958	0.3204	0.4786	0.0872	0.0890	0.1389	0.2020
BIOSKILL-EVOLVER	**0.0144**	**0.0623**	0.1649	**0.2927**	**0.4470**	**0.5930**	**0.4412**	**0.4099**	**0.2669**	**0.3395**

caveat we return to in Sec. 4.2. The end-to-end setting gives the agent only a free-text instruction and a raw dataset and requires it to perceive, plan, generate code, execute, self-verify and report unaided.

Baselines. We compare classical pipelines (Wolf et al., 2018; Palla et al., 2022; Hao et al., 2021; Zhao et al., 2021), the task-specific deep models of Sec. 2, and LLM/agentic systems: ReAct (Yao et al., 2022) and AutoGen (Wu et al., 2023) on the same backbone, plus OmicsNavigator (Yiyao et al., 2025) and LLMiniST (Wei et al., 2025), each placed only on the tasks it was designed for. The decisive comparison is **GLM-5.1 (ReAct, no skills)**, sharing our backbone and prompts but without SKILLFORGE, ADAPTWEAVE and EVOCYCLE.

Metrics and implementation. Domain identification uses ARI and NMI; expression prediction gene-wise PCC, Spearman, MSE and SSIM; annotation macro-F1 and accuracy; the end-to-end setting task success rate (SR), plan validity, execution success and V_{final}. We also report #Interv., but the harness charges exactly one intervention to each failed episode, so #Interv. $= 1 - \text{SR}$ by construction and the column carries no independent information. All planning, code generation, reflection and judging use GLM-5.1 (Zeng et al., 2026) under one configuration throughout: temperature 0.6 for planning and reflection, 0.2 for code generation. All remaining hyperparameters are the values fixed in Sec. 3 and are listed together in Appendix A.6. **Every number comes from a single recorded run per configuration**; the backbone is sampled at non-zero temperature, so runs are not bit-reproducible and we report no seed variance (App. A.4).

4.2 TASK-LEVEL COMPARISONS

Table 1 shows that BIOSKILL-EVOLVER attains the best mean ARI (0.2669 vs. 0.2241 for STAGATE) and mean NMI (0.3395 vs. 0.2981), leading on seven of eight per-dataset cells. Two observations matter more than the averages. First, the three stateless agent rows are numerically identical to each other *and* to Scanpy (Leiden): given the same instruction, all of them synthesize the same default Leiden pipeline. This is the cleanest evidence in the paper that agentic scaffolding alone does not improve analysis quality; what changes the outcome is having a platform-appropriate skill to retrieve. Second, BIOSKILL-EVOLVER does *not* win on HER2 ARI, where SpaGCN (0.1802) and GRAS4T (0.1725) lead our 0.1649, while our NMI on the same section is highest by a clear margin. Higher NMI with lower ARI is the signature of a partition that identifies domains correctly but fragments them, which Sec. 4.6 confirms on saved outputs.

Gene expression prediction from histology. Appendix A.5, Table 6 reports prediction of 100 target genes from paired H&E on 12 cSCC and 2 10x Breast sections, using within-section 80/20 spot splits shared by all methods. BIOSKILL-EVOLVER attains the best mean PCC (0.5910 vs.

Table 2: **Cell-type annotation.** Macro-F1 and accuracy against expert or reference-mapped labels; Mean is the unweighted average of the three columns. The stateless GLM-5.1 row is a partial run ($10/13$ episodes produced valid metrics; its means use measured episodes only). Best results are **bold**; second best are underlined.

Method	Xenium Breast		HER2 Breast (ST)		MOSTA Embryo		Mean	
	F1↑	Acc.↑	F1↑	Acc.↑	F1↑	Acc.↑	F1↑	Acc.↑
Classical / reference-transfer baselines								
Seurat label transfer	0.0159	0.0227	0.2638	0.5389	0.1986	0.4215	0.1594	0.3277
LLM / agentic systems								
OmicsNavigator	0.0161	0.0377	0.2792	0.4999	**0.2092**	0.4240	0.1682	0.3205
LLMiniST	0.0217	0.0386	0.2778	0.4959	0.1924	0.4107	0.1640	0.3151
GLM-5.1 (ReAct, no skills)	0.0180	0.0484	0.2192	0.6171	0.1330	0.5320	0.1234	0.3992
BIOSKILL-EVOLVER	**0.0277**	**0.1409**	**0.3423**	**0.7325**	0.1616	**0.5714**	**0.1772**	**0.4816**

0.5542 for ST-Net), Spearman (0.5668 vs. 0.5648) and SSIM (0.6994 vs. 0.5831 for BLEEP), and the second-best mean MSE (0.2865 vs. 0.2796 for BLEEP). We read this with three caveats. First, the gain is carried by cSCC, where we lead on PCC, MSE and SSIM; on 10x Breast, BLEEP is best on all four metrics, we rank third on PCC and Spearman, and our MSE (0.1358) is the highest of all six methods, roughly twice that of most baselines. Because the Mean column weights the 14 sections equally, cSCC contributes 12 of 14 sections and dominates it. Second, on 10x Breast our PCC exceeds ST-Net's (0.5520 vs. 0.5278) while our MSE is nearly double (0.1358 vs. 0.0722); correlation and squared error disagree this way when a predictor recovers spatial-expression *shape*, which spots are high or low, without calibrating absolute magnitude. Third, the comparison is neither compute- nor development-matched: baselines run under released default configurations, whereas our expression skill was developed on these cohorts and the target genes were selected by full-section variance, so the table measures within-section interpolation rather than generalization to unseen sections or cohorts. SSIM for every row, including ours, is computed by the shared external evaluator; our own verifier does not compute it, so SSIM never enters the learning signal.

Cell-type annotation.

Table 2 contains the most instructive disagreement in the paper. BIOSKILL-EVOLVER has the best accuracy on all three datasets and the best mean accuracy by a wide margin (0.4816 vs. 0.3992), yet on MOSTA Embryo its macro-F1 (0.1616) falls below three baselines. The mechanism is abstention: the marker-consensus skill emits `unassigned` when no reference marker set reaches consensus for a cluster, which removes confident errors, hence the accuracy gain, but contributes a predicted class that can never match a gold label and leaves rare classes unrecovered, which macro-F1 punishes because it weights classes equally (Sec. 4.6 quantifies this). We read it as a system property rather than a defect of the metric: a verifier-driven agent trained to avoid unsupported claims trades rare-class recall for precision, and macro-F1 makes that trade visible.

4.3 END-TO-END AUTONOMOUS ANALYSIS

Table 3 isolates the contribution of accumulated skills. Against the backbone-matched stateless agent, success rises from 0.0789 to 0.5000 ($3/38$ to $19/38$ episodes) and V_{final} from 0.3108 to 0.4633, while *plan validity and execution success are identical at* 0.9474. That equality is the point of the table: every system here writes code that parses and runs on at least 35 of 38 episodes ($36/38$ for all systems except AutoGen, which reaches $35/38$), and the stateless agents still succeed on at most 5 of 38 episodes. The bottleneck is not code generation but producing an output that survives biological checking, and only the skill library moves that number. The table also explains why V_{final} stays below 0.5 for our system: multiplied by the coverage factor κ of Eq. equation 6, an episode that produces a defensible clustering but omits a requested marker table is scored as incomplete. Execution success near 0.95 with V_{final} near 0.46 therefore measures partial fulfilment of multi-part instructions, not broken code.

Table 3: **End-to-end autonomous analysis** over the same 38 episodes. #Interv. equals $1 - \mathrm{SR}$ by harness construction (Sec. 4.1). Best results are **bold**; second best are underlined.

Method	SR↑	Plan Valid.↑	Exec. Succ.↑	V_{final} ↑	#Interv.↓
ReAct agent (GLM-5.1)	0.0789	**0.9474**	**0.9474**	0.3108	0.9211
AutoGen (multi-agent)	0.1316	0.9211	0.9211	0.3375	0.8684
OmicsNavigator	0.0526	**0.9474**	**0.9474**	0.3058	0.9474
LLMiniST	0.0789	**0.9474**	**0.9474**	0.3108	0.9211
GLM-5.1 (ReAct, no skills)	0.0789	**0.9474**	**0.9474**	0.3108	0.9211
BIOSKILL-EVOLVER	**0.5000**	**0.9474**	**0.9474**	**0.4633**	**0.5000**

Table 4: **Module ablation** on the same episode set. The last row is a proxy: it evaluates an early, low-diversity library snapshot and therefore measures library composition, *not* the unimplemented regularizer of Appendix A.3. An expression-PCC column is omitted because the local episode set contains no expression-prediction episodes. Best results are **bold**; second best are underlined.

Variant	Domain ARI↑	Annot. F1↑	Agentic SR↑	V_{final} ↑	$\Delta V_{\mathrm{final}}$
BIOSKILL-EVOLVER (full)	–	–	**0.5000**	**0.4633**	**0.0000**
w/o locality-preserving fusion	0.1911	0.2008	**0.5000**	0.4587	−0.0046
w/o skill graph (SKILLFORGE)	0.1911	0.2008	**0.5000**	0.4588	−0.0045
w/o constraint synthesis (ADAPTWEAVE)	0.1911	0.2008	**0.5000**	0.4596	−0.0037
w/o verification (BIOAUDIT)	0.1911	0.1883	0.4737	0.4196	−0.0437
w/o evolution (EVOCYCLE)	0.1911	0.1460	0.1316	0.3153	−0.1480
early low-diversity library (proxy)	0.1911	0.1767	**0.5000**	0.4252	−0.0381

4.4 MODULE ABLATIONS

Table 4 is unambiguous about where the gain lives. Freezing the library costs 0.1480 of V_{final} and drops success from 0.5000 to 0.1316, back to the stateless regime of Table 3, and removing verifier-guided acceptance costs a further 0.0437. The three perception and planning ablations cost between 0.0037 and 0.0046, differences we do not consider meaningful for a single run. Those three rows also carry identical Domain ARI and Annotation F1, which deserves an explanation rather than a footnote: on this episode set the evolved library already contains a single dominant candidate per subtask, so flattening the graph, disabling the fusion block and replacing beam search with greedy selection all converge on the same executed plan, and only the reranking score changes. This is a property of the current episode set rather than evidence that SKILLFORGE and ADAPTWEAVE are inert: their function is to keep the plan space searchable as the library grows, and 22 skills do not yet stress that machinery. We state it plainly because a reader should be able to rule out the alternative reading from the text alone.

4.5 DOES THE AGENT ACCUMULATE CAPABILITY?

Appendix A.7, Table 7 reports the frozen-graph evaluation: the skill graph is snapshotted at 16 checkpoints and each snapshot re-evaluated on the same 22 held-out Visium and Xenium instructions. Average V_{final} rises from 0.1547 at episode 22 to 0.5663 and success from 0.2273 to 0.8636 ($5/22$ to $19/22$), which is the behaviour expected from accumulation rather than episode memorization. The trajectory is not monotone, however: checkpoint 352 falls to 0.5041 from 0.5634 at checkpoint 286, and success drops to $17/22$. Our acceptance rule guarantees that an individual skill is not replaced by a worse version on its own validation episodes; it does not guarantee that the library as a whole improves monotonically on a different instruction set, because a specialization accepted for one platform changes which skill is retrieved for neighbouring tasks. The honest statement is a strong upward trend with late-stage fluctuation of roughly 0.06.

The gains are also not where the growth is. The library adds only three skills across 352 episodes, all platform specializations, and the largest single jump, $0.2008 \rightarrow 0.5123$ between checkpoints 66 and 88, with success going from $7/22$ to $19/22$, occurs with library size unchanged at 20 and

specialization count unchanged at 1. Whatever produced it was a *rewrite* of existing skills by reflective refinement, not an addition. With Table 4, where freezing evolution is the most damaging ablation by an order of magnitude, this suggests the operative mechanism is error-trace-driven repair of a small library, and that "how many skills" is the wrong quantity to track. One caveat: the evaluation is held out over instructions but not over cohorts, since both sets are drawn from the same pool of sections.

4.6 Source-level failure modes

Saved per-unit outputs let us name the two dominant failure modes instead of inferring them from aggregates (Appendix A.11, Figure 2). On the MOSTA embryo sections almost every predicted cluster is dominated by a single gold domain (8 of 12 clusters exceed 70% purity in E2S3, 8 of 11 in E2S2), so the agent splits large domains rather than confusing anatomically distinct ones, which depresses ARI far more than NMI and reproduces the pattern of Table 1; the implied repair is resolution selection conditioned on an expected domain count. On HER2 the errors are dominated by abstention, which accounts for 23% of all predictions in C1 and 33% in D1 and is the direct cost of the marker-consensus rule behind the accuracy/macro-F1 split of Table 2. Part of the remaining gap is a benchmark property rather than model error: the released label set contains an `undetermined` class that the agent's controlled vocabulary cannot express, so those units are wrong by construction.

4.7 Transfer, backbone and verifier reliability

Three further studies appear in full in the appendix. *Transfer* (App. A.8): a Xenium-specialized library scores $V_{\mathrm{final}} = 0$ on Visium zero-shot, because no retained skill passes both the platform filter and the precondition test and hence no executable plan exists, yet it reaches 0.6031 with 0.9333 success after one round of target-side specialization, while the reverse direction transfers almost for free ($0.4802 \rightarrow 0.4875$). Specialization therefore buys within-platform quality at the cost of declared generality, and the fallback along specialization edges matters more than the specializations themselves; transfer onto the sequencing-based subdomains lifts V_{final} by 0.01–0.03 but never produces a passing episode, a negative result we report as such. *Backbone* (App. A.9): with the framework fixed, GPT-5.5, Claude-Opus-4.8 and GLM-5.1 all reach $\mathrm{SR} = 0.5000$ and lie within 0.017 on V_{final}, and DeepSeek-v4-flash is close (0.4722 over 36 scorable episodes); the Gemini run is incomplete (12 of 38 episodes) and we draw no conclusion from it, so the supported claim is narrower than backbone-independence. *Verifier reliability* (App. A.10): we ran no blinded expert study and report a leave-one-out internal-consistency check instead, whose main finding is negative: V_{pathway} is *anti*-correlated with the consensus of the other signals ($r = -0.70$, agreement 18.7%) while V_{marker} and V_{spatial} are only weakly positive ($r = 0.40, 0.39$), and the aggregate's $r = 0.97$ is near-circular by construction. The defensible conclusion is that the verifier is useful enough to drive learning, as Table 4 shows, while at least one of its five granularities is miscalibrated.

5 Conclusion and Outlook

Within the bounds recorded in Appendix A.4, this study supports a focused claim: the bottleneck for self-improving spatial omics agents is a usable reward rather than a better planner. The evidence is threefold. First, the library grows by only three skills while performance improves substantially, indicating that error-trace-driven repair matters more than acquisition. Second, every system executes at least 92% of episodes without error, yet stateless agents pass biological checking on at most one in seven, showing that runnable code is not enough. Third, one verifier granularity is anti-correlated with the others, so the reward itself needs calibration. Together, these findings suggest that future progress will depend less on adding increasingly complex agent scaffolds and more on building verifiers that expose biologically meaningful failure modes across tissues, platforms, and analysis objectives. The next steps are therefore concrete: replace the Hallmark pathway collection with tissue-matched gene sets, reweight or ablate the pathway term, report the coverage factor κ separately from quality in V_{final}, add a resolution-selection skill conditioned on an expected domain count, extend the annotation label vocabulary, and exercise merging, diversity and curriculum mechanisms on a larger library. Multi-seed replication and blinded expert ratings remain the most important measurements for sharpening future claims. The released per-episode artifacts are intended to make both straightforward.

AI USE STATEMENT

LLMs were used in two ways in this work. First, LLMs accessed through their APIs are part of the proposed method. In BIOSKILL-EVOLVER, GLM-5.1 (Zeng et al., 2026) handles planning, code generation, reflective skill refinement and the plausibility judgment V_{LLM} (Sec. 3), and the backbone study swaps in other models with the framework held fixed (App. A.9). Second, LLMs were used to polish the language of the manuscript.

ETHICS STATEMENT

All authors have read and follow the ICLR Code of Ethics. This work uses only publicly available spatial omics datasets, all cited in Sec. 4.1, and involves no new data collection from human subjects. We did not run a blinded expert evaluation, so the verifier score does not establish biological correctness (App. A.10). Biological conclusions produced by the system should therefore be checked by domain experts. We report the limitations and negative results openly (App. A.4).

REPRODUCIBILITY STATEMENT

Sec. 3 and App. A.1 give the module specifications. App. A.6 gives the evaluation protocol, how episodes are counted and all hyperparameters. App. A.2 lists which mechanisms ran in the reported experiments. Domain identification and annotation scores can be recomputed from the released per-unit assignment files without any model call. The deterministic components give exactly the same results when re-run. The LLM components are sampled at non-zero temperature, so re-running them will not match exactly. For this reason we release the per-episode artifacts (plans, generated code, outputs and verifier traces).

REFERENCES

Alma Andersson, Ludvig Larsson, Linnea Stenbeck, Fredrik Salmén, Anna Ehinger, Sunny Z Wu, Ghamdan Al-Eryani, Daniel Roden, Alex Swarbrick, Åke Borg, et al. Spatial deconvolution of her2-positive breast cancer delineates tumor-associated cell type interactions. *Nature communications*, 12(1):6012, 2021.

Tommaso Biancalani, Gabriele Scalia, Lorenzo Buffoni, Raghav Avasthi, Ziqing Lu, Aman Sanger, Neriman Tokcan, Charles R Vanderburg, Åsa Segerstolpe, Meng Zhang, et al. Deep learning and alignment of spatially resolved single-cell transcriptomes with tangram. *Nature methods*, 18(11): 1352–1362, 2021.

Ao Chen, Sha Liao, Mengnan Cheng, Kailong Ma, Liang Wu, Yiwei Lai, Xiaojie Qiu, Jin Yang, Jiangshan Xu, Shijie Hao, et al. Spatiotemporal transcriptomic atlas of mouse organogenesis using dna nanoball-patterned arrays. *Cell*, 185(10):1777–1792, 2022.

Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature medicine*, 30(3):850–862, 2024.

Youngmin Chung, Ji Hun Ha, Kyeong Chan Im, and Joo Sang Lee. Accurate spatial gene expression prediction by integrating multi-resolution features. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11591–11600. IEEE, 2024.

César Domínguez Conde, Chao Xu, Louie B Jarvis, Daniel B Rainbow, Sara B Wells, Tamir Gomes, Sarah K Howlett, Ondrej Suchanek, Krzysztof Polanski, Hamish W King, et al. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*, 376(6594):eabl5197, 2022.

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

Xiaohang Fu, Yue Cao, Beilei Bian, Chuhan Wang, Dinny Graham, Nirmala Pathmanathan, Ellis Patrick, Jinman Kim, and Jean Yee Hwa Yang. Spatial gene expression at single-cell resolution from histology using deep learning with ghist. *Nature Methods*, 22(9):1900–1910, 2025.

Yang Gui, Chao Li, and Yan Xu. Spatial domains identification in spatial transcriptomics using modality-aware and subspace-enhanced graph contrastive learning. *Computational and Structural Biotechnology Journal*, 23:3703–3713, 2024.

Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

Bryan He, Ludvig Bergenstråhle, Linnea Stenbeck, Abubakar Abid, Alma Andersson, Åke Borg, Jonas Maaskola, Joakim Lundeberg, and James Zou. Integrating spatial gene expression and breast tumour morphology via deep learning. *Nature Biomedical Engineering*, 4(8):827–834, 2020.

Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J Irwin, Edward B Lee, Russell T Shinohara, and Mingyao Li. Spagcn: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature Methods*, 18(11):1342–1351, 2021.

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, et al. High resolution mapping of the tumor microenvironment using integrated single-cell, spatial and in situ analysis. *Nature communications*, 14(1):8353, 2023.

Andrew L Ji, Adam J Rubin, Kim Thrane, Sizun Jiang, David L Reynolds, Robin M Meyers, Margaret G Guo, Benson M George, Annelie Mollbrink, Joseph Bergenstråhle, et al. Multimodal analysis of composition and spatial architecture in human squamous cell carcinoma. *cell*, 182(2): 497–514, 2020.

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

Ming Y Lu, Bowen Chen, Drew FK Williamson, Richard J Chen, Ivy Liang, Tong Ding, Guillaume Jaume, Igor Odintsov, Long Phi Le, Georg Gerber, et al. A visual-language foundation model for computational pathology. *Nature medicine*, 30(3):863–874, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

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

Giovanni Palla, Hannah Spitzer, Michal Klein, David Fischer, Anna Christina Schaar, Louis Benedikt Kuemmerle, Sergei Rybakov, Ignacio L Ibarra, Olle Holmberg, Isaac Virshup, et al. Squidpy: a scalable framework for spatial omics analysis. *Nature methods*, 19(2):171–178, 2022.

Minxing Pang, Kenong Su, and Mingyao Li. Leveraging information in spatial transcriptomics to predict super-resolution gene expression from histology images in tumors. *bioRxiv*, pp. 2021–11, 2021.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Vipul Singhal, Nigel Chou, Joseph Lee, Yifei Yue, Jinyue Liu, Wan Kee Chock, Li Lin, Yun-Ching Chang, Erica Mei Ling Teo, Jonathan Aow, et al. Banksy unifies cell typing and tissue domain segmentation for scalable spatial omics data analysis. *Nature genetics*, 56(3):431–441, 2024.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

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

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):15, 2018.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Ronald Xie, Kuan Pang, Sai Chung, Catia Perciani, Sonya MacParland, Bo Wang, and Gary Bader. Spatially resolved gene expression prediction from histology images via bi-modal contrastive learning. *Advances in Neural Information Processing Systems*, 36:70626–70637, 2023.

Chenling Xu, Romain Lopez, Edouard Mehlman, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular systems biology*, 17(1):MSB20209620, 2021.

Hang Xu, Huazhu Fu, Yahui Long, Kok Siong Ang, Raman Sethi, Kelvin Chong, Mengwei Li, Rom Uddamvathanak, Hong Kai Lee, Jingjing Ling, et al. Unsupervised spatially embedded deep representation of spatial transcriptomics. *Genome medicine*, 16(1):12, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

L Yiyao, N Vakharia, W Liang, AT Mayer, R Luo, AE Trevino, and Z Wu. Omicsnavigator: an llm-driven multi-agent system for autonomous zero-shot biological analysis in spatial omics. 2025.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Yuansong Zeng, Zhuoyi Wei, Weijiang Yu, Rui Yin, Yuchen Yuan, Bingling Li, Zhonghui Tang, Yutong Lu, and Yuedong Yang. Spatial transcriptomics prediction from histology jointly through transformer and graph neural networks. *Briefings in Bioinformatics*, 23(5):bbac297, 2022.

Edward Zhao, Matthew R Stone, Xing Ren, Jamie Guenthoer, Kimberly S Smythe, Thomas Pulliam, Stephen R Williams, Cedric R Uytingco, Sarah EB Taylor, Paul Nghiem, et al. Spatial transcriptomics at subspot resolution with bayesspace. *Nature biotechnology*, 39(11):1375–1384, 2021.

Pengfei Zhu, Han Shu, Yongtian Wang, Xiaofeng Wang, Yuan Zhao, Jialu Hu, Jiajie Peng, Xuequn Shang, Zhen Tian, Jing Chen, et al. Maest: accurately spatial domain detection in spatial transcriptomics with graph masked autoencoder. *Briefings in Bioinformatics*, 26(2):bbaf086, 2025a.

Sichen Zhu, Yuchen Zhu, Molei Tao, and Peng Qiu. Diffusion generative modeling for spatially resolved gene expression inference from histology images. In *International Conference on Learning Representations*, volume 2025, pp. 19666–19690, 2025b.

A APPENDIX

A.1 MODULE SPECIFICATIONS

Data profile. The profile $\mathbf{d}$ records platform type, tissue origin and species, spatial resolution, gene-panel size, median genes per unit, total unit count, expression-matrix sparsity, and boolean flags for paired histology, cell-type labels, region annotations and niche labels. It is computed by deterministic inspection of the input object; no component is learned. Downstream, $\mathbf{d}$ gates which optional features are extracted, which skills pass the platform filter of Eq. equation 3, and which canonical task families are admissible.

Locality-preserving fusion. The per-unit representation of Sec. 3.1 is

$$\mathbf{h}_j = \mathrm{zscore}\big[\,\mathbf{u}_j\,;\,\gamma(\mathbf{s}_j)\,;\,\mathbf{m}_j\,;\,\bar{\mathbf{u}}_{\mathcal{N}(j)}\,\big], \qquad \bar{\mathbf{u}}_{\mathcal{N}(j)} = \tfrac{1}{|\mathcal{N}(j)|}\textstyle\sum_{i\in\mathcal{N}(j)}\mathbf{u}_i, \tag{8}$$

with $\mathbf{u}_j$ a truncated-SVD embedding of the preprocessed expression $\mathbf{e}_j = \log(\mathbf{1} + \mathrm{CPM}(\mathbf{x}_j)) \in \mathbb{R}^G$, $\mathbf{m}_j$ the optional morphology block (dropped when the image pathway is off) and $\mathcal{N}(j)$ the spatial nearest neighbours of unit j.

Spatial encoding. Random Fourier features are $\gamma(\mathbf{s}_j) = d_s^{-1/2}[\cos(2\pi\mathbf{B}\mathbf{s}_j); \sin(2\pi\mathbf{B}\mathbf{s}_j)] \in \mathbb{R}^{2d_s}$ with $\mathbf{B} \in \mathbb{R}^{d_s\times 2}$ drawn once from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ and held fixed. Larger σ makes the encoding sensitive to fine positional differences, which suits single-cell-resolution platforms; smaller σ suits spot-level data. In the reported runs n_{spatial}=4 frequencies are used, and the expression block keeps n_{expr}=8 truncated-SVD components.

Skill-graph edge types and registry. Composition edges are weighted by Eq. equation 2. Specialization edges $k_a \to k_b$ mark k_b as a platform- or tissue-specific variant of k_a and encode the hierarchy that lets the planner fall back to the general skill when no variant matches the current profile. Conflict edges are undirected and mark skills that perform the same function under incompatible assumptions; constraint (C3) of Sec. 3.3 forbids both endpoints from entering one plan. The seed registry covers quality-control filtering, library-size normalization, highly variable gene selection, dimensionality reduction, spatial nearest-neighbour graph construction, Leiden clustering, differential expression testing, marker-overlap cell-type annotation, spatial domain detection, ligand–receptor interaction analysis, gene module scoring, niche composition summarization, pathway enrichment, visualization and report generation.

Verifier weighting and reference sources. The task-family weights α_j of Eq. equation 6 place greater weight on expression-fidelity terms for prediction tasks and on annotation-correctness and marker terms for annotation tasks; signals whose evaluation module is unavailable for a task are dropped and the remaining weights renormalized. Marker references are held as a local curated table; pathway testing uses the MSigDB Hallmark collection at α=0.05. The spatial graph used by V_{spatial} is the generic k-nearest-neighbour construction provided by Squidpy (Palla et al., 2022).

Controlled condition vocabulary. Preconditions and postconditions are drawn from a fixed vocabulary $\mathcal{C}$ covering data state (`has_expression`, `has_coordinates`, `has_histology`), processing state (`qc_filtered`, `normalized_expr`, `hvg_selected`, `embedding_built`, `spatial_graph_built`) and analysis state (`clusters_assigned`, `de_computed`, `cell_typed`, `domains_called`, `pathways_scored`, `report_written`). Because the vocabulary is closed, constraint (C1) of Sec. 3.3 is a set-containment test and never invokes the language model.

Reflective refinement prompt. The refinement call receives four blocks: the skill source; up to ten recent failure cases, each with the input profile, the produced output, any error message and the verification score; the biological context of those cases (tissue, subtype, platform); and a template that requires a diagnosis, a list of concrete code-level modifications, and a complete replacement implementation. The diagnoses observed in our logs are of the kind "clustering resolution too coarse for the bin size of this section" or "normalization assumes a genome-wide panel but the data carries a targeted panel", i.e. statements that name a parameter and a data property. The acceptance rule of Sec. 3.5 is applied before the replacement is written back to the graph.

A.2 SPECIFICATION–IMPLEMENTATION CORRESPONDENCE

Table 5 maps every mechanism described in this paper to its status in the runs that produced our numbers. We provide it so that no reported result can be mistakenly attributed to a component that did not execute, and so that a reader checking the released code against the paper knows in advance which differences are intentional.

Table 5: Status of each specified mechanism in the reported runs. "Exercised" means the component ran in every evaluated episode; "specified only" means it is part of the design and is described here but did not run and therefore contributes to no reported number.

Mechanism	**Reference**	**Status in the reported runs**
Deterministic data profiling	Sec. 3.1	Exercised.
Locality-preserving fusion	Eq. equation 8	Exercised; the optional image pathway uses handcrafted morphology features by default.
Template-guided task decomposition	Sec. 3.1	Exercised, with the deterministic rule-based fallback available.
Typed skill tuple and condition sets	Eq. equation 1	Exercised.
Composition edge weight	Eq. equation 2	Exercised in the convex form shown, with fallback to the coverage term when no transition history exists.
Skill retrieval score	Eq. equation 3	Exercised with lexical/structural overlap for rel; no neural embedding model is used.
Constrained beam search	Eq. equation 4	Exercised, followed by one language-model reranking pass.
V_{stat}	Sec. 3.4	Exercised as a weighted average of raw metrics; no per-metric thresholds and no top-k recovery term.
V_{marker}	Eq. equation 5	Exercised against a local curated marker table.
V_{pathway}	Sec. 3.4	Exercised; see the calibration caveat in Sec. 4.7.
V_{spatial}	Sec. 3.4	Exercised as the three-term composite on a kNN spatial graph, not as a single Delaunay Moran's I.
V_{final} with coverage factor	Eq. equation 6	Exercised, including the multiplicative target-coverage term κ.
EMA tracking and evolution trigger	Eq. equation 7	Exercised.
Reflective refinement and acceptance	Sec. 3.5	Exercised.
Platform specialization	Sec. 3.5	Exercised using the 80% failure-concentration rule; a formal per-platform ANOVA test is not used.
Skill merging	App. A.3	Specified only; no merge was triggered.
Diversity regularizer	App. A.3	Specified only; the ablation row in Table 4 is a library-composition proxy.
Curriculum scheduling	App. A.3	Specified only; episodes were presented in recorded order.
Blinded expert rating	App. A.4	Not conducted.
Multi-seed replication	App. A.4	Not conducted; one run per configuration.
SSIM inside the verifier	Sec. 4.2	Not computed inside the verifier; SSIM for all rows of Table 6, including BIOSKILL-EVOLVER, comes from the external evaluator.

A.3 SPECIFIED BUT NOT EXERCISED

We record these three mechanisms because they are part of the intended design and because naming them precisely makes the boundary in Table 5 checkable.

Skill merging. Two skills k_a, k_b are merge candidates when they are semantically close and neither dominates the other in measured performance, $\cos(\mathbf{k}_a, \mathbf{k}_b) > \theta_{\mathrm{merge}}$ and $|\bar{V}_{k_a} - \bar{V}_{k_b}| < \epsilon$, with $\theta_{\mathrm{merge}}{=}0.85$ and $\epsilon{=}0.05$. The backbone would then be asked to emit a unified implementation inheriting the union of the two condition sets and the concatenation of their histories. With a library

of 22 skills and a registry designed to avoid near-duplicates, no pair reached the trigger in our runs, so merging never fired.

Diversity regularization. The intended regularizer penalizes pairwise similarity across the library,

$$\mathcal{L}_{\text{div}} = -\frac{2}{|\mathcal{V}|(|\mathcal{V}|-1)} \sum_{i<j} \log\bigl(1 - \cos^2(\mathbf{k}_i, \mathbf{k}_j)\bigr),$$

rejecting a candidate skill that would raise $\mathcal{L}_{\text{div}}$ past a threshold unless it beats the best existing skill for the target subtask by at least $\delta{=}0.05$. It is not implemented in the evaluated code path, and the corresponding row of Table 4 substitutes an early low-diversity library snapshot, which measures the effect of library composition rather than of the rule.

Curriculum scheduling. The intended scheduler defines task difficulty relative to the current library as

$$\text{diff}(\tau) = 1 - \max_{k \in \mathcal{V}} \text{score}(k \mid \tau) + \lambda_{\text{novel}}\bigl(1 - \max_{k \in \mathcal{V}} \cos(\mathbf{k}, \boldsymbol{\tau})\bigr),$$

and samples episodes with probability proportional to $\exp(\text{diff}(\tau)/T_c^{(t)})$ with T_c annealed from 1.0 to 0.1. Evaluating it requires a separate evolution run under randomized ordering, which we did not perform; episodes in all reported runs were presented in their recorded order. No conclusion in this paper depends on curriculum effects.

A.4 Extended limitations

(i) Every value comes from a *single recorded run* per configuration; the backbone is sampled at non-zero temperature, so runs are not bit-reproducible, we report no seed variance, and the spread of between 0.0037 and 0.0046 among the perception and planning ablations should not be interpreted. **(ii)** No blinded pathologist rating was conducted, so Table 10 is an internal-consistency proxy and does not establish that V_{final} tracks biological correctness. **(iii)** There is no three-way split: domain identification and annotation are scored zero-shot on full sections, and expression prediction uses within-section 80/20 spot splits without a validation partition, with target genes chosen by full-section variance and our expression skill developed on the same cohorts, so it measures within-section interpolation rather than cross-section generalization. **(iv)** The benchmark is 38 end-to-end episodes on a locally available subset of platforms; CosMx and Visium HD are described in the design but not evaluated. **(v)** Skill merging, the diversity regularizer and curriculum scheduling are specified but did not run; Appendix A.2 maps every equation to its status. **(vi)** The pathway granularity is miscalibrated on this data, and the coverage factor in V_{final} conflates partial completion with poor quality.

A.5 Gene expression prediction from histology

All metrics in Table 6, including SSIM, are computed by the external evaluator shared by all rows; the agent's own verifier computes neither SSIM nor a top-k recovery term, so these values are evaluation-only and never enter V_{final}. The Mean column averages the 14 sections equally rather than by spot count, so cSCC carries a 12:2 weight. The per-dataset columns should therefore be read alongside it: on 10x Breast, BioSkill-Evolver trails BLEEP and TRIPLEX on PCC and Spearman and has the highest MSE of all methods.

A.6 Evaluation protocol, harness conventions and reproducibility

Episode accounting. The end-to-end set contains 38 episodes, so every rate in Table 3 is a multiple of $1/38$: $0.0526{=}2/38$, $0.0789{=}3/38$, $0.1316{=}5/38$, $0.5000{=}19/38$, $0.9211{=}35/38$, $0.9474{=}36/38$. The frozen-checkpoint evaluation of Table 7 uses the 22 Visium and Xenium instructions, so its rates are multiples of $1/22$: $0.2273{=}5/22$, $0.3182{=}7/22$, $0.7727{=}17/22$, $0.8182{=}18/22$, $0.8636{=}19/22$. Stereo-seq rounds are excluded from the checkpoint series because that branch had not completed enough evolution episodes at the time of the snapshot.

Intervention counting. The harness charges one manual intervention to any episode that does not pass the automatic success criteria and zero otherwise, so #Interv. $= 1 - \text{SR}$ identically. We retain the column because prior work reports it, but it should not be read as an independent measurement of

Table 6: Gene expression prediction from H&E using within-section 80/20 spot splits shared by all methods. Results cover 12 cSCC sections and two 10x Breast sections. Best results are bold; second-best results are underlined.

Method	cSCC				10x Breast				Mean			
	PCC↑	Spear.↑	MSE↓	SSIM↑	PCC↑	Spear.↑	MSE↓	SSIM↑	PCC↑	Spear.↑	MSE↓	SSIM↑
ST-Net	<u>0.5586</u>	**0.5668**	0.3434	0.4671	0.5278	0.5530	0.0722	0.5800	<u>0.5542</u>	<u>0.5648</u>	0.3047	0.4832
HisToGene	0.2977	0.3224	0.3432	0.5152	0.3534	0.4280	0.0724	0.5797	0.3057	0.3375	0.3045	0.5244
Hist2ST	0.1866	0.1901	0.3433	0.4712	0.2918	0.3194	0.0720	0.5732	0.2017	0.2085	0.3046	0.4858
BLEEP	0.4482	0.4727	<u>0.3202</u>	<u>0.5394</u>	**0.7639**	**0.8143**	**0.0366**	**0.8452**	0.4933	0.5215	**0.2796**	<u>0.5831</u>
TRIPLEX	0.2472	0.4624	0.3451	0.5148	<u>0.6690</u>	<u>0.7095</u>	<u>0.0620</u>	0.7291	0.3075	0.4977	0.3047	0.5454
BioSkill-Evolver (ours)	**0.5975**	<u>0.5564</u>	**0.3116**	**0.6944**	0.5520	0.6296	0.1358	<u>0.7296</u>	**0.5910**	**0.5668**	<u>0.2865</u>	**0.6994**

PCC, Spearman, and SSIM use raw-count predictions. MSE uses $\log_{10}(1 + \mathrm{CP10k})$: prediction and truth are independently normalized to a row total of 10,000 over the same 100 target genes. Negative predictions are clipped to zero for MSE only. This transformation is applied after prediction; it does not imply training on log-normalized targets. Dataset scores are arithmetic means over sections. Mean averages all 14 sections equally, giving cSCC and Breast weights of 12:2; it is not weighted by spot count. The archived target selection uses full-section expression variance, and the ours skill was developed on these cohorts. These results measure within-section interpolation rather than unseen-section or independent-cohort generalization.

human effort; a genuine measurement would require logging the number and type of edits a human actually made, which we did not collect.

Scoring. Domain identification and annotation scores are computed by the evaluation harness from released per-unit assignment files, so they can be recomputed from the artifacts without any model call; the case-study metrics quoted in Figure 2 were obtained this way. Expression-prediction metrics, including SSIM, come from the external evaluator shared by all methods in Table 6; the agent's own verifier does not compute SSIM or a top-k recovery term, so the SSIM values reported for BIOSKILL-EVOLVER are evaluation-only and do not enter the learning signal.

Reproducibility. Deterministic components, profiling, preprocessing, fusion, condition checking, beam search and all rule-based scorers, reproduce exactly. Language-model components do not: planning and reflection run at temperature 0.6 and code generation at 0.2, so a re-run will not reproduce the released artifacts bit-for-bit. We therefore release the per-episode artifacts (plans, generated code, outputs, verifier traces) rather than claiming re-runnable determinism, and we report no variance estimates.

Hyperparameters. Fusion uses $n_{\mathrm{expr}}{=}8$ truncated-SVD components and $n_{\mathrm{spatial}}{=}4$ Fourier frequencies, with the optional UNI2-h pathway (Chen et al., 2024) contributing $n_{\mathrm{foundation}}{=}16$ components when enabled; retrieval uses $\lambda{=}(0.4, 0.3, 0.3)$ and top-$K{=}5$; plan synthesis uses $B{=}5$ and $\eta{=}0.3$; evolution uses $\mu{=}0.9$, $\theta_{\mathrm{evo}}{=}0.6$, $\theta_{\mathrm{var}}{=}0.04$ and $T{=}10$; verification uses $\beta{=}0.7$ and $\alpha{=}0.05$. No value was tuned on held-out data.

A.7 COMPLETE EVOLUTION HISTORY

The full history makes two features of Sec. 4.5 easier to check. The series is flat from checkpoint 22 to 66 ($0.1547 \rightarrow 0.2008$), then steps sharply at checkpoint 88 without any change in library size or specialization count, then improves slowly from 0.5123 to a maximum of 0.5634 at checkpoint 286 while the three specializations are added, and finally falls back to 0.5041 at checkpoint 352. The intermediate dip at checkpoint 132 (0.4906, $18/22$) has the same character as the late dip. Both are consistent with retrieval churn: accepting a specialization changes which skill wins Eq. equation 3 for neighbouring subtasks, and that reshuffling is not covered by the per-skill acceptance guarantee.

Table 7: Complete EVOCYCLE checkpoint history on the Visium and Xenium episodes; Checkpoint identifiers span `20260609T021452Z`–`20260610T181518Z` in chronological order. The final row is the Table 3 evaluation restricted to Visium+Xenium episodes. A Domain ARI column is omitted because these outputs do not store gold domain labels. Best results are **bold**; second best are underlined.

Checkpoint	#Skills	#Spec.	Avg V_{final} ↑	Agentic SR↑
Ep. 22	19	0	0.1547	0.2273
Ep. 44	19	0	0.1555	0.2273
Ep. 66	20	1	0.2008	0.3182
Ep. 88	20	1	0.5123	0.8636
Ep. 110	20	1	0.5165	0.8636
Ep. 132	20	1	0.4906	0.8182
Ep. 154	21	2	0.5176	0.8636
Ep. 176	21	2	0.5400	0.8636
Ep. 198	21	2	0.5375	0.8636
Ep. 220	22	3	0.5404	0.8636
Ep. 242	22	3	0.5621	0.8636
Ep. 264	22	3	0.5597	0.8636
Ep. 286	22	3	0.5634	0.8636
Ep. 308	22	3	0.5632	0.8636
Ep. 330	22	3	0.5621	0.8636
Ep. 352	22	3	0.5041	0.7727
Final	**22**	**3**	**0.5663**	**0.8636**

A.8 CROSS-PLATFORM AND CROSS-DOMAIN TRANSFER

Table 8: Cross-platform and cross-domain transfer. "Zero-shot" evaluates a source-specialized library directly on the target; "+EVOCYCLE" allows target-side specialization; Δ is the absolute gain. The identifier `stomics:*` is the harness label for the sequencing-based track (HER2 ST, MOSTA, cSCC). V_{final} and SR replace the ARI-based columns of a conventional transfer table because these outputs do not store gold spatial-domain labels. Best results are **bold**; second best are underlined.

Source	Target	Zero-shot V_{final} ↑	+EVOCYCLE V_{final} ↑	ΔV_{final}	Zero-shot SR↑	+EVOCYCLE SR↑	ΔSR
xenium-specialized	visium	0.0000	**0.6031**	+0.6031	0.0000	**0.9333**	+0.9333
visium-specialized	xenium	0.4802	0.4875	+0.0073	0.7143	0.7143	0.0000
stomics	visium+xenium	**0.5634**	0.5663	+0.0029	**0.8636**	0.8636	0.0000
visium+xenium	stomics:her2	0.2426	0.2701	+0.0275	0.0000	0.0000	0.0000
visium+xenium	stomics:mosta	0.3540	0.3641	+0.0101	0.0000	0.0000	0.0000
visium+xenium	stomics:cscc	0.4275	0.4487	+0.0212	0.0000	0.0000	0.0000

Two rows deserve comment. The exact 0.0000 in the first row is a structural failure, not a numerical one: a library whose retained skills all declare Xenium compatibility leaves no candidate that passes the platform filter of Eq. equation 3 and the precondition test (C1) for a Visium section, so ADAPTWEAVE returns no executable plan and every episode is scored zero. One round of target-side specialization repairs this completely (0.9333 success). The reverse direction transfers almost for free, and the asymmetry is the practical lesson of the table: specialization buys within-platform quality at the cost of declared generality, so the fallback along specialization edges is what keeps a specialized library usable elsewhere. The last three rows are a negative result we prefer to state directly, specializing onto the sequencing-based subdomains lifts V_{final} by 0.01–0.03 but never produces a passing episode, so the improvement is real on the verifier scale and irrelevant at the task level.

Table 9: Backbone study with the BIOSKILL-EVOLVER framework held fixed. "Open?" indicates open-weight availability; n is the number of scorable episodes. The GLM-5.1 row reuses the Table 3 result. Two DeepSeek episodes returned no scorable output and are excluded from its denominator. The Gemini run is incomplete and charges its 26 missing episodes as zero, so it is not comparable with the complete rows and is shown for transparency only. Best results are **bold**; second best are underlined.

Backbone	Open?	Agentic SR↑	V_{final} ↑	#Interv.↓	n
gpt-5.5	No	**0.5000**	**0.4720**	**0.5000**	38
deepseek-v4-flash	Yes	0.4722	0.4380	0.5278	36
Gemini-3.1-Pro-preview	No	0.1842	0.1580	0.8158	12/38
claude-opus-4-8	No	**0.5000**	0.4554	**0.5000**	38
GLM-5.1 (ours)	Yes	**0.5000**	0.4633	**0.5000**	38

A.9 BACKBONE STUDY

The three complete alternative backbones and ours agree on success rate to within a single episode and on V_{final} within 0.017; GPT-5.5 is marginally ahead of our open-weight default, which we report as measured. Since #Interv. is $1 - \text{SR}$ by construction (App. A.6), it adds nothing here either. The narrow reading supported by this table is that the framework, not the specific backbone, accounts for the difference from the stateless baseline in Table 3; the table cannot support a claim that every frontier model works equally well, because one of the five runs is incomplete.

A.10 VERIFIER RELIABILITY

Table 10: Leave-one-out internal-consistency check on 510 local episode records. For each component signal, Pearson r and Spearman ρ are computed against the equal-weight consensus of the *other* available signals; for V_{final} the target is the equal-weight consensus of the five components, which makes that row near-circular. Agreement thresholds both the signal and the target at 0.5. This is not a validation against expert judgment, which we did not collect. Best results are **bold**; second best are underlined.

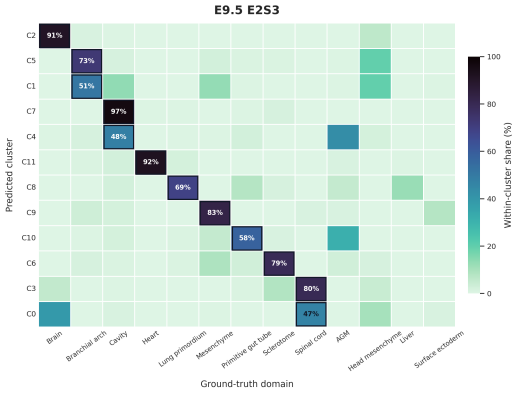
Verification signal	Pearson r	Spearman ρ	Agreement (%)	n
V_{stat} (statistical)	0.7220	0.7050	**95.38**	65
V_{marker} (marker overlap)	0.4048	0.3703	48.30	412
V_{pathway} (fGSEA)	-0.6985	-0.7242	18.69	412
V_{spatial} (composite)	0.3885	0.2866	36.65	412
V_{LLM} (LLM-as-judge)	0.8824	0.8309	69.83	411
V_{final} (aggregate)	**0.9723**	**0.9772**	77.67	412

The statistical signal is available for only 65 records because most episodes have no quantitative target, and its high agreement should be read against that small denominator. The pathway signal is the clear outlier: it is anti-correlated with the consensus of the other granularities on every measure. We do not think this shows that pathway coherence is uninformative about biology; the more likely explanation is that testing every predicted group against the Hallmark collection rewards groups with a single dominant enriched program, which on these tissues tends to select large, heterogeneous clusters rather than well-resolved domains. Either reweighting α_{pathway} toward zero or replacing the Hallmark collection with a tissue-matched gene-set library is a concrete next step, and until one of them is done the pathway term should be regarded as a source of noise in V_{final} rather than as evidence.

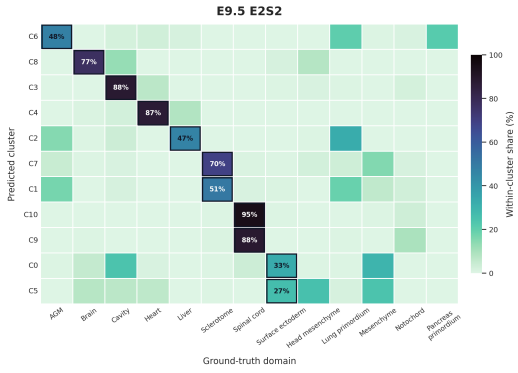
A.11 ADDITIONAL CASE-STUDY DETAIL

The four panels of Figure 2 are `mosta_e2s3` and `mosta_e2s2` for domain identification and `her2_c1` and `her2_d1` for annotation; all metrics quoted below are recomputed from the released per-unit assignment files.

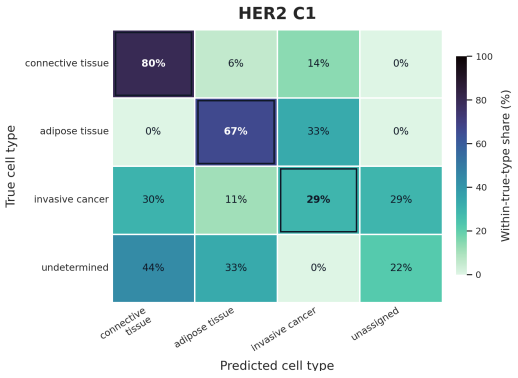
Domain identification. E2S2 contains 4,333 scored units, 11 predicted clusters and 13 gold domains, giving ARI 0.4494 and NMI 0.5745; E2S3 contains 5,059 units, 12 predicted clusters and 13 gold domains, giving ARI 0.5569 and NMI 0.6594. In both sections the predicted-cluster count is below the gold-domain count, yet the dominant error is over-segmentation of the largest domains rather than merging of distinct ones: brain, spinal cord and surface ectoderm each attract two or three clusters while several small domains (notochord, pancreas primordium, liver in E2S2) attract none. This combination, fewer clusters than domains, but the clusters unevenly distributed across them, is what produces the NMI-above-ARI signature discussed in Sec. 4.6. Concretely, three clusters all map to brain in E2S3, and in E2S2 two map to spinal cord (88% and 95%) and two to surface ectoderm. It suggests that a resolution-selection skill conditioned on an expected domain count, rather than a better embedding, is the appropriate repair.



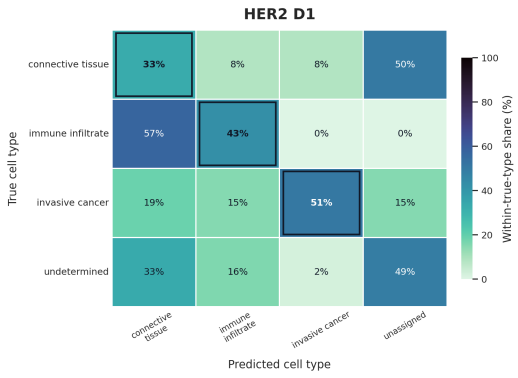
(a) Domain identification, MOSTA E9.5 E2S3 (ARI 0.557, NMI 0.659)



(b) Domain identification, MOSTA E9.5 E2S2 (ARI 0.449, NMI 0.575)



(c) Annotation, HER2 C1 (macro-F1 0.228, accuracy 0.387)



(d) Annotation, HER2 D1 (macro-F1 0.229, accuracy 0.360)

Figure 2: **Saved outputs of BIOSKILL-EVOLVER.** (a, b) Cluster-by-domain contingency, rows normalized to the predicted cluster. (c, d) Cell-type confusion, rows normalized to the gold class. All metrics are recomputed from the released per-unit assignment files.

Annotation. HER2 C1 contains 173 scored units over four predicted and four gold classes (macro-F1 0.2276, accuracy 0.3873); D1 contains 306 units (macro-F1 0.2291, accuracy 0.3595). In C1 the agent recovers connective tissue (80%) and adipose tissue (67%) well but invasive cancer poorly (29%), with 30% of cancer units called connective tissue and 29% left `unassigned`. The predicted label distributions are {connective tissue 70, invasive cancer 43, unassigned 39, adipose 21} for C1 and {unassigned 101, connective tissue 83, invasive cancer 81, immune infiltrate 41} for D1, against

gold distributions dominated by invasive cancer (126 and 139). Abstention is thus 23% and 33% of all predictions. The gold sets also contain 9 and 51 units labelled `undetermined`, a class with no counterpart in the agent's controlled label vocabulary; since macro-F1 averages over classes, these units alone bound the attainable macro-F1 well below one regardless of model quality. Separating this benchmark artifact from genuine confusion, most visibly the tumour/stroma boundary, where 30% of C1's invasive-cancer units are called connective tissue, would require either extending the label vocabulary or excluding unlabelable units, and we did neither.