LSM-Copilot: A Skill-Flow Agent for Fluorescence Microscopy Analysis

Ruofan Liu
rfliu@uw.edu
University of Washington
Seattle, WA, USA

Pengcheng Chen
pengcc@uw.edu
University of Washington
Seattle, WA, USA

Eric J. Seibel
eseibel@uw.edu
University of Washington
Seattle, WA, USA

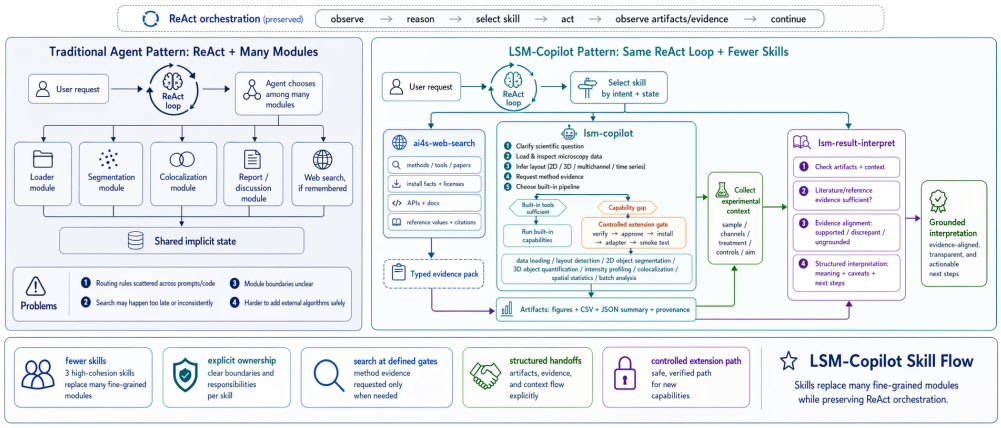

Figure 1. LSM-Copilot, a skill-Flow contract (right) replaces the many-module ReAct pattern (left): the same ReAct loop is preserved, but domain modules collapse into three skills (Search, Process, Interpret) with explicit ownership, evidence-gated search, typed handoffs, and a controlled extension path—the five properties summarized along the bottom.

Abstract

Fluorescence and confocal laser scanning microscopy produce large, heterogeneous image collections whose analysis often requires manual coordination across vendor readers, segmentation models, statistical scripts, and reporting tools. We present **LSM-Copilot**, an agent suite that turns raw microscopy files and a natural-language analysis goal into auditable masks, tables, figures, and reports for two evaluated microscopy workflows: spot localization and 3D LSM object quantification. The system is organized as a **Skill-Flow**: a thin host agent routes work through three self-contained skills, *Search*, *Process*, and *Interpret*, while the skills package prompts, tools, domain knowledge, and verification logic. On public fluorescence spot-detection benchmarks, the same LSM-Copilot skills run in Claude Code, Codex, Cursor, and OpenClaw [**?**] with Claude 4.7 Opus [1], GPT-5.5 xhigh [9], Composer 2 [2], and DeepSeek V4 Pro [4]. All four hosts independently recover evidence-backed method plans and obtain matching final F1 scores on Spotiflow, deepBlink, and 3D simulated spot-localization tasks. The prompt gives no benchmark-source hints, and the skills contain no hard-coded method names. On private 3D raw LSM crystallization data, the same skills work without web search: each host writes and runs a local 3D analysis pipeline and agrees with Imaris [10] reference measurements at about the five-percent level. These results support a narrow claim: skill-level packaging can make both search-guided method discovery and fully local microscopy analysis portable across host agents on the evaluated tasks.

CCS Concepts: • **Computing methodologies** → **Computer vision tasks**; *Natural language processing*; • **Applied computing** → **Bioinformatics**.

Keywords: bioimage analysis, microscopy, agent skills, scientific agents

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Agent Skills '26, San Jose, CA, USA
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Ruofan Liu, Pengcheng Chen, and Eric J. Seibel. 2026. LSM-Copilot: A Skill-Flow Agent for Fluorescence Microscopy Analysis. In *Proceedings of The First Workshop on Agent Skills: Design, Evaluation, and Optimization of Procedural Knowledge for LLM Agents (Agent Skills '26)*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Fluorescence and confocal laser scanning microscopy (LSM) are central to cell biology, spatial transcriptomics, chemistry, and materials characterization. In practice, however, the analysis path from pixels to conclusions remains fragmented. A scientist may need to open vendor formats such as `.lsm`, `.czi`, and `.lif`; infer channel, Z, time, and voxel metadata; choose between classical image processing and learned methods such as Cellpose [12] or StarDist [11]; run quantification; and finally assemble figures and a report. General LLM agents can help write scripts, but they do not by default know when to search for method evidence, how to audit a microscopy pipeline, or how to prevent benchmark leakage during calibration. These details matter because small decisions about channel layout, voxel calibration, thresholding, and benchmark splits can change scientific conclusions while leaving little trace in an ad hoc script.

We build **LSM-Copilot**, a microscopy analysis agent suite whose main design choice is to move domain complexity out of the host loop and into reusable skills. The host keeps the usual observe–reason–act pattern of ReAct-style agents [14], but it selects among only three high-level skill contracts: *Search* for method and evidence discovery, *Process* for raw data analysis, and *Interpret* for result explanation. Each skill is a filesystem package containing instructions, tools, knowledge files, and handoff contracts. The resulting Skill-Flow is less a fixed workflow than a portable unit of scientific capability: the same analysis policy can be loaded by different coding agents without embedding microscopy logic in the host itself.

This paper makes three contributions. First, we describe a complete skill package for fluorescence and confocal microscopy, including file intake, method discovery, processing, and report generation. The implementation includes additional modules such as colocalization, spatial statistics, tracking, and batch processing, but this paper evaluates only spot localization and 3D LSM quantification. Second, we introduce the Skill-Flow organization, which separates host orchestration from domain skill ownership and makes tool extension reviewable. Third, we report a portability evaluation on public spot-localization suites and a private organic-material crystallization benchmark against Imaris [10] reference segmentation.

2 Skill-Flow Design

Figure 1 contrasts the two patterns: a conventional ReAct host coordinates many ad hoc modules over shared implicit state, whereas LSM-Copilot keeps the same ReAct loop but routes through three skills with typed handoffs. The host owns routing, context passing, and tool execution; skills own the domain policy. This division is important because microscopy tasks are rarely a single tool call: method choice depends on modality, dimensionality, density, physical calibration, available benchmark splits, and the scientific question.

Skill package format. Each skill is a directory containing `SKILL.md` (instructions and contracts), `tools/` (Python modules the host invokes), `knowledge/` (curated reference tables), and an optional `extensions/` directory for runtime adapters. Skills communicate through typed JSON handoff schemas: Search emits an *evidence pack*; Process consumes it and emits an *artifact bundle*; Interpret consumes that to write the report. This schema is the only inter-skill coupling. In the evaluated workflows, the concrete handoff artifacts are an evidence JSON from Search, masks/CSVs/figures from Process, and a report plus summary JSON from Interpret. This structure makes review practical: a new method enters through an adapter and evidence record rather than through hidden prompt text.

Search performs grounded method discovery before expensive analysis begins. It searches for algorithms, APIs, licenses, installation notes, benchmark claims, and reference values, then returns the evidence pack. The skill specifies what evidence to collect and how to check it; it does not contain a hard-coded list of benchmark-specific methods, papers, or expected scores. This makes search a required gate for method selection and third-party extensions rather than an optional background behavior. Thus a host must recover the method from external evidence, not from an answer key embedded in the skill.

Process converts microscopy data into auditable artifacts. It reads Zeiss LSM/CZI, Leica LIF, OME-TIFF, plain TIFF, and MRC-like files; infers array layout and physical metadata; and routes to the appropriate processing path. Built-in modules cover classical 2D and 3D segmentation, intensity profiling, colocalization, spatial statistics, tracking, enhancement, and batch processing; the present evaluation exercises spot localization and 3D object quantification. When a task requires a stronger external method, Process records the capability gap, verifies candidate tools through Search, adds a thin adapter, smoke-tests it, and logs provenance.

Interpret consumes the artifact bundle emitted by Process. It asks for experimental context not present in file metadata, such as sample identity, channel-to-marker mapping, treatment groups, and comparators. It then aligns numerical results with the evidence pack and writes a report whose claims trace back to generated CSV and JSON artifacts. In this paper we evaluate the reproducibility of generated measurements and provenance, not open-ended biological interpretation.

3 Evaluation

We evaluate the same skill suite under four host/model settings: Claude Code with Claude 4.7 Opus [1], Codex with GPT-5.5 xhigh [9], Cursor with Composer 2 [2], and OpenClaw with DeepSeek V4 Pro [4]. The host and underlying model change, but the Search, Process, and Interpret skill files are held fixed. This setup does not measure which model is strongest; it asks whether the same skill package can support two operating modes: web-search-guided method discovery on public benchmarks, and fully local code generation and analysis on private raw microscopy data. Each run starts from the same natural-language request and data path. Skill-enabled rows are compared with no-skill raw-model rows to isolate the effect of procedural guidance. The comparison is therefore about procedural stability, not model ranking: each host may write different intermediate code, but should arrive at compatible evidence, artifacts, and measurements.

For the public benchmarks, we test discovery rather than memorized execution: the run prompts do not reveal the source papers, benchmark tables, SOTA method names, or expected scores for the data. The skills also contain **no benchmark-answer key**. They encode the search, extension, execution, and audit procedure, so each host must use the skills to discover and justify the method choice step by step.

3.1 Spot Localization Benchmarks

Table 1 reports final F1 on three spot-detection datasets: the public Spotiflow benchmark [5], the deepBlink test splits [6], and a 3D simulated spot-localization benchmark. In all four skill-enabled hosts, Search found the same method evidence: U-FISH [13] reports F1 0.984 on its own 3D simulation set, RS-FISH [3] native is reported at F1 0.952 in the U-FISH paper, Piscis [8] 3D reports mean F1 0.895, Spotiflow [5] reports F1 0.882 on its own 3D synthetic benchmark, and Big-FISH [7] appears at F1 0.683 in the Spotiflow 3D benchmark table. We use these numbers as method-selection evidence rather than as a single merged leaderboard, because the reported 3D benchmarks are not all the same dataset. The hosts then ran compatible approved pipelines and produced matching final F1 scores. Claude Code and Cursor selected a Spotiflow-only path for the 2D data, whereas Codex and OpenClaw selected the best available method per split; the final aggregate scores remain closely aligned.

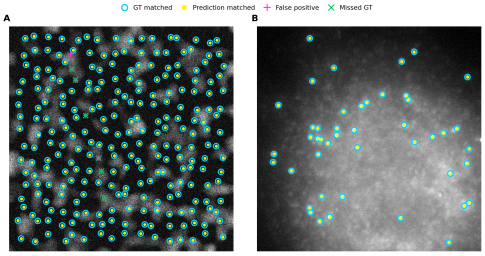

Figure 2. Representative 2D spot-localization examples; aggregate benchmark scores are reported separately.

The hosts do not share identical traces or intermediate code. The useful property is that Skill-Flow forces the same evidence gate, extension gate, and artifact schema, stabilizing the scientific decision boundary rather than replaying a fixed script. Relative to raw-model runs, the skill-enabled setting yields substantial improvements in average F1: from 0.580 to 0.893 on Spotiflow, from 0.772 to 0.968 on deepBlink, and from 0.374 to 0.858 on the 3D simulated benchmark. The largest gap appears in 3D, where unguided hosts often choose incompatible detection code or thresholds. Once a protocol is selected, deterministic tools produce the final numbers; the agent role being evaluated is evidence-grounded orchestration and provenance capture. For the 3D simulated benchmark, the shared run also reports 1.58 px localization error with default thresholds.

3.2 Raw 3D LSM Quantification

The second benchmark uses our private dataset of 24 Zeiss LSM 3D confocal stacks of organic material crystallization samples (small molecules crystallized from solution; 20×, two fluorescence channels, voxel $0.415 \times 0.415 \times 1.0\,\mu$m). Reference segmentations were produced by a domain expert using Imaris [10] (Oxford Instruments) surface rendering and manual curation, yielding per-crystal diameter, volume, and centroid as CSVs. Web search is disabled: each host sees only raw images, writes local 3D processing code, and is compared with Imaris only post hoc (Table 2). Imaris is the standard commercial tool used by the contributing laboratory, so agreement with it is a meaningful external check rather than a training target for the agent.

Figure 3 visualizes one case; Table 2 reports the aggregate private-benchmark comparison.

The four skill-enabled hosts produce closely aligned, but not identical, results. Their generated code chooses slightly different thresholds and object filters, which shifts the metrics within a narrow band: count MAE

Table 1. Spot-localization portability. Entries are final F1 scores. Identical entries reflect convergence on the same final tool invocation; intermediate code differs across hosts (Sec. 3.1).

Host / model	Mode	Spotiflow data ↑	deepBlink data ↑	3D simulated data ↑
Claude Code / Claude 4.7 Opus	LSM-Copilot skills	0.893	0.968	0.858
Codex / GPT-5.5 xhigh	LSM-Copilot skills	0.893	0.968	0.858
Cursor / Composer 2	LSM-Copilot skills	0.893	0.968	0.858
OpenClaw / DeepSeek V4 Pro	LSM-Copilot skills	0.893	0.968	0.858
Claude Code / Claude 4.7 Opus	raw model	0.588	0.883	0.424
Codex / GPT-5.5 xhigh	raw model	0.654	0.756	0.295
Cursor / Composer 2	raw model	0.528	0.663	0.299
OpenClaw / DeepSeek V4 Pro	raw model	0.549	0.785	0.476

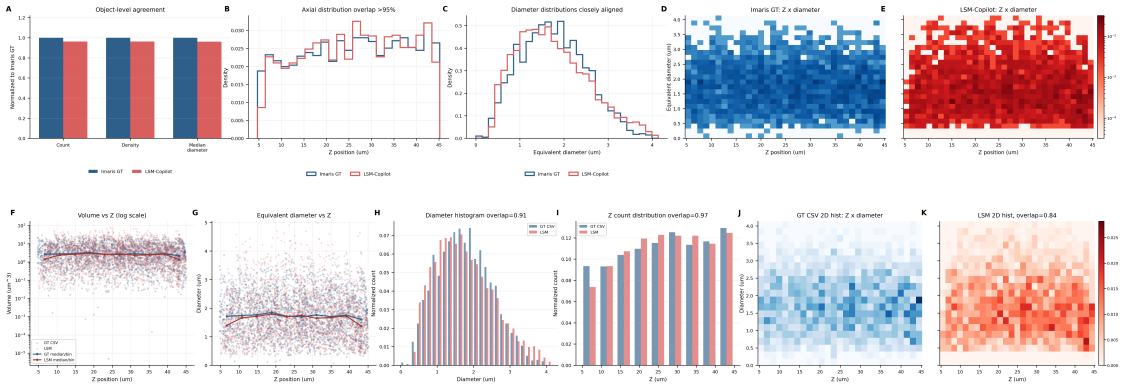

Figure 3. Representative private 3D raw-LSM case comparing LSM-Copilot quantification with Imaris commercial-software output. (A–E) Summary views; (F–K) diagnostic distributions. Aggregate benchmark results are reported separately.

Table 2. Private 3D LSM quantification against Imaris reference segmentation.

Host / model	Mode	Cnt. MAE (%) ↓	Pool err. (%) ↓	Diam. MAE (%) ↓	Diam. ovlp. ↑	Z ovlp. ↑
Claude Code / Claude 4.7 Opus	LSM-Copilot skills	5.06	-4.33	3.76	0.861	0.959
Codex / GPT-5.5 xhigh	LSM-Copilot skills	3.23	-1.25	1.82	0.902	0.981
Cursor / Composer 2	LSM-Copilot skills	5.08	-4.12	3.91	0.857	0.948
OpenClaw / DeepSeek V4 Pro	LSM-Copilot skills	4.37	-3.75	3.72	0.891	0.975
Claude Code / Claude 4.7 Opus	raw model	28.3	-20.4	11.57	0.754	0.916
Codex / GPT-5.5 xhigh	raw model	8.15	-31.94	14.31	0.637	0.870
Cursor / Composer 2	raw model	92.4	-97.4	11.55	0.593	0.278
OpenClaw / DeepSeek V4 Pro	raw model	10.5	-25.8	11.76	0.682	0.825

ranges from 3.23% to 5.08%, pooled count error from −1.25% to −4.33%, and median-diameter MAE from 1.82% to 3.91%. Raw-model runs are far less stable; the skill-enabled setting yields substantial improvements across all aggregate metrics. Average count MAE rises from 4.44% to 34.84%, average absolute pooled error from 3.36% to 43.89%, and average diameter MAE from 3.30% to 12.30%. Distributional agreement also drops, from 0.878 to 0.667 for diameter overlap and from 0.966 to 0.722 for Z-count overlap. This variation is desirable in the skill-enabled condition: the skills constrain scientific procedure without forcing all hosts to emit identical code.

4 Discussion

The main result is that the same skill package supports two useful forms of portability across four host/model settings, despite different reasoning traces and intermediate code. On public spot-localization benchmarks, LSM-Copilot uses web search to find current method evidence, justify an extension, install and wrap it, run a smoke test, and produce benchmark artifacts. On private raw 3D LSM stacks, it works without web search: the skills guide each host to write local processing code, choose thresholds and filters, and compare only after processing is complete. In both settings, the run leaves an auditable trail: evidence or pipeline decision, parameters, CSV tables, figures, and JSON summaries.

These results support the central Skill-Flow design choice (Figure 1, right). The host model can vary, but

the scientific procedure remains anchored in the skills: Search constrains method choice when evidence is online, Process constrains local execution and provenance, and Interpret consumes generated artifacts rather than inventing results from the prompt. The private crystallization benchmark adds an external check through Imaris [10], a commercial tool used by the contributing laboratory. The completed raw-model ablations show why those gates matter: skill-enabled runs produce substantial improvements over raw models across both public and private benchmarks.As one extreme case, raw Cursor/Composer 2 produced 92.4% count MAE on this dataset, far outside the 3.23–5.08% band achieved by all four skill-enabled runs

The scope of the evidence is intentionally narrow. We evaluate two analysis families in one shared environment, so portability across different operating systems, institutional restrictions, and operators remains future work. Likewise, modules such as colocalization, tracking, enhancement, and expert-scored free-form interpretation are part of the suite but are not claimed as validated here. This keeps the current claim focused: Skill-Flow packaging makes the evaluated microscopy procedures portable and auditable; broader modality coverage is the next step.

5 Conclusion

We presented Skill-Flow, an agent design that moves domain procedure into portable filesystem skills with typed handoffs. On microscopy spot detection and private 3D LSM quantification, the same skills run on four host/model combinations and produce stable, auditable results, supporting Skill-Flow as a route to reusable scientific agents.

Acknowledgments

This work was supported by the National Science Foundation under award NSF PFI 2234356, the M. J. Murdock Charitable Trust under award CI-202324074, and the University of Washington CoMotion under award 202324074.

References

[1] Anthropic. 2025. Claude: A Family of Highly Capable Foundation Models. https://www.anthropic.com/claude Model: Claude 4.7 Opus.

[2] Anysphere. 2025. Cursor: The AI Code Editor. https://cursor.com Composer 2 AI model.

[3] Ella Bahry, Laura Breimann, Marwan Zouinkhi, Leo Epstein, Klim Kolyvanov, Nicholas Mamrak, Benjamin King, Xi Long, Kyle I. S. Harrington, Timothée Lionnet, and Stephan Preibisch. 2022. RS-FISH: Precise, interactive, fast, and scalable FISH spot detection. *Nature Methods* 19 (2022), 1563–1567. doi:10.1038/s41592-022-01669-y

[4] DeepSeek-AI. 2024. DeepSeek-V3 Technical Report. arXiv:2412.19437

[5] Albert Dominguez Mantes, Antonio Herrera, Irina Khven, Anjalie Schlaeppi, Eftychia Kyriacou, Georgios Tsissios, Evangelia Skoufa, Luca Santangeli, Elena Buglakova, Emine Berna Durmus, Suliana Manley, Anna Kreshuk, Detlev Arendt, Can Aztekin, Joachim Lingner, Gioele La Manno, and Martin Weigert. 2025. Spotiflow: Accurate and Efficient Spot Detection for Fluorescence Microscopy with Deep Stereographic Flow Regression. *Nature Methods* 22 (2025), 1495–1504. doi:10.1038/s41592-025-02662-x

[6] Bastian T. Eichenberger, YinXiu Zhan, Markus Rempfler, Luca Giorgetti, and Jeffrey A. Chao. 2021. deepBlink: Threshold-Independent Detection and Localization of Diffraction-Limited Spots. *Nucleic Acids Research* 49, 13 (2021), 7292–7297. doi:10.1093/nar/gkab546

[7] Arthur Imbert, Wei Ouyang, Adham Safieddine, Emeline Coleno, Christophe Zimmer, Edouard Bertrand, Thomas Walter, and Florian Mueller. 2022. FISH-quant v2: a scalable and modular tool for smFISH image analysis. *RNA* 28, 6 (2022), 786–795. doi:10.1261/rna.079073.121

[8] Zijian Niu, Aoife O'Farrell, Jingxin Li, Sam Reffsin, Naveen Jain, Ian Dardani, Yogesh Goyal, and Arjun Raj. 2024. Piscis: a novel loss estimator of the F1 score enables accurate spot detection in fluorescence microscopy images via deep learning. bioRxiv preprint. doi:10.1101/2024.01.31.578123

[9] OpenAI. 2024. *GPT-4 Technical Report*. Technical Report. OpenAI. arXiv:2303.08774

[10] Oxford Instruments. 2024. Imaris: 3D and 4D Image Visualization, Analysis, and Interpretation Software. https://imaris.oxinst.com/

[11] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. 2018. Cell Detection with Star-Convex Polygons. In *Medical Image Computing and Computer Assisted Intervention*. Springer, Cham, 265–273. doi:10.1007/978-3-030-00934-2_30

[12] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. 2021. Cellpose: A Generalist Algorithm for Cellular Segmentation. *Nature Methods* 18, 1 (2021), 100–106. doi:10.1038/s41592-020-01018-x

[13] Weize Xu, Huaiyuan Cai, Qian Zhang, Zhengze Wang, Jiajun Yang, Xiaofeng Wu, Chengwen Li, Chenghua Cui, Changzhi Liu, Jin He, Florian Mueller, Jinxia Dai, Chen Hao, Wei Ouyang, and Gang Cao. 2025. U-FISH: a fluorescent spot detector for imaging-based spatial-omics analysis and AI-assisted FISH diagnosis. *Genome Biology* 26 (2025), 261. doi:10.1186/s13059-025-03736-x

[14] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. International Conference on Learning Representations. https://openreview.net/forum?id=WE_vluYUL-X