Toward AI VIS Co-Scientists: An Agent Harness for End-to-End Design of Scientific Visualization Applications

Haichao Miao[1], Zhimin Li[2], Kuangshi Ai[3], Kaiyuan Tang[3], Chaoli Wang[3], Peer-Timo Bremer[1], and Shusen Liu[1]
[1]Lawrence Livermore National Laboratory, [2]Vanderbilt University, [3]University of Notre Dame
e-mail: miao1@llnl.gov, zhimin.li@vanderbilt.edu, kai@nd.edu, ktang2@nd.edu, chaoli.wang@nd.edu, bremer5@llnl.gov, liu42@llnl.gov

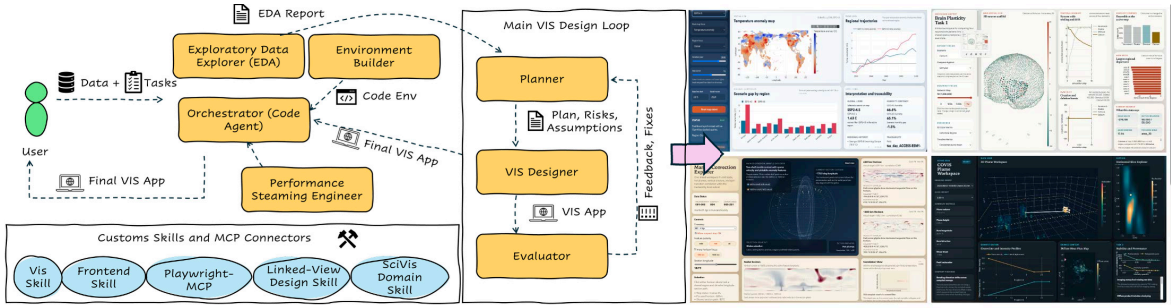


Figure 1: **The VIS co-scientist agent harness.** From a dataset and a one-sentence task description, an orchestrating code agent runs a multi-stage workflow with no further human input: an *Exploratory Data Analyzer* profiles the data, an *Environment Builder* configures the environment, a *Performance Streaming Engineer* handles large or remote data, and a design loop of *Planner*, *VIS Designer*, and *Evaluator* iterates until browser-based validation reports no blocking issues. Custom skills and MCP connectors (bottom) determine what the agents can inspect, verify, and generate. The output is a single-page interactive VIS App with coordinated views.

Abstract

Recent LLM-based visualization systems generate charts from natural language, drive visualization tools, prototype analytics interfaces, and score well on visualization benchmarks, but complex scientific visualization tasks still require repeated human steering before they yield a usable application. Much of the work done by visualization researchers calls for automating a larger task: an interactive application that starts from ambiguous domain materials, explores the data, derives task requirements, designs visual representations and coordinated interactions, implements the result, and checks whether it answers the original scientific questions. We argue that agentic visualization systems should automate this application-level design loop, and we study it in a zero-shot setting: one prompt with no human intervention after launch. The domain scientists who need such applications rarely have a visualization expert to consult, so the agent has to get as far as it can from an underspecified request. Each clarifying turn adds design intent that the system is supposed to supply, making the final result hard to attribute. We support the position with a reusable multi-agent harness that acts as a VIS co-scientist for domain experts. We test the harness on IEEE SciVis Contests spanning multiple science and engineering fields. These contests include the pressures we care about: ambiguous requirements, diverse data modalities, design trade-offs, and task-driven validation. We use the runs to map what the current harness can and cannot do, and to identify where visualization expertise should enter agent design.

1 Introduction

Across the natural sciences and machine learning, researchers have begun building AI co-scientists: systems intended to carry out autonomous scientific reasoning, experimentation, and discovery [29, 50, 20, 34, 12]. Such systems will not remove humans from science; if anything, they increase the need for oversight, interpretation, validation, and visualization [52]. Current co-scientist work often treats visualization as an output format rather than as scientific work in its own right, even though designing a visual analysis tool demands perceptual judgment, weighing design alternatives [26, 17], requirements elicitation, and engineering against real data. That makes an *AI VIS co-scientist* part of the scientific work itself, and it falls to the visualization community to define what it must be able to do.

Recent LLM-based visualization systems now handle natural-language chart generation, tool-mediated scientific visualization, visual analytics prototyping, and benchmarked SciVis workflows [11]. What they deliver, however, is usually an individual chart, script, report, or structured benchmark output (Table 1). Real visualization research tasks often require automating a larger task: an interactive visualization application (VIS App) that begins from ambiguous domain materials, profiles the data, derives task requirements, designs coordinated views and interactions, implements the application, and validates whether the result supports the original scientific questions. This application-level design loop remains underexplored, and it is rarely studied *zero-shot*, from a single prompt with no back-and-forth with the user.

Our position is that this loop is what agentic visualization systems should automate, and that zero-shot single-turn operation is the setting in which to measure it (Sec. 3.3). The domain scientists who most need a visualization application often have no visualization expert to consult. A system that converges only under expert steering then assumes the expertise it was meant to supply: the agent has to be the visualization expert, not a drafting assistant for one. Methodologically, every clarifying turn adds design intent that the system should contribute itself. A multi-turn result measures operator and agent together, whereas a zero-shot result shows what an agentic system can and cannot do when visualization researchers design the harness but do not operate the run.

To test this position, we built an agent harness around an LLM code agent (OpenAI's Codex using *gpt-5.4*). An orchestrator delegates bounded subtasks to specialized subagents and communicates with them through explicit artifacts (Fig. 1). We ran the harness on IEEE SciVis Contest problem packages from 2021–2026. They provide real data, underspecified goals, and externally authored tasks that neither we nor the model chose. A combined harness ablation, in

Table 1: Agentic and LLM-based visualization work organized by the artifact the system is responsible for delivering. The application-level deliverable, reached without human turns, is the gap this position paper targets.

Deliverable	Representative work
Chart or visualization spec	nvAgent [35], DeepVIS [40], PlotGen [19], DataWink [47], MultiVis-Agent [30]
Annotation, overlay, narrative	FinFlier [22], narrative generation [46]
State or script inside a VIS tool	ChatVis [37], ParaView agents [28], NLI4VolVis [5], SASAV [41], TopoPilot [18], VizGenie [10]
Analysis report or insight set	composable visual reporting [21], DataMinds [49], AgenticInsight [32]
VA prototype with a human in the loop	LightVA [53], LLM4VA [48], HiLSVA [1]
Critique, preference, or repair	Visualizationary [39], DracoGPT [44], MisVis-Fix [15]
Benchmark score	SciVisAgentBench [3], VisEval [13], ChartQA tuning [51]
Complete interactive VIS App, zero-shot	**this work**

which the same model receives the same prompt with the subagents, skills, and connectors disabled together, yields standalone plots rather than an integrated multi-view application.

Our contributions are:

- **A position.** We frame the complete interactive VIS App as what agentic visualization systems should automate, show why this target is underserved, and argue that zero-shot single-turn operation is the right test setting.
- **A reusable agent harness.** An arrangement of orchestrator, subagents, custom visualization skills, and browser-based validation that produces and verifies complete VIS Apps, described so that it can be directly applied to a new problem.
- **Case studies and a capability demonstration.** We test this approach on existing SciVis contests (2021–2025) and zero-shot entries to an ongoing externally judged contest, locating where current agents succeed (data understanding, coordinated views, task coverage) and where they do not (encoding quality, perception of dynamic behavior).

2 Related Work: Reading the Field by What It Delivers

We group prior work not by technique but by the artifact a system is ultimately responsible for producing, which makes the gap we target visible as a missing row (Table 1). A recent state-of-the-art report [11] surveys the space far more broadly; our purpose is only to locate the application-level endpoint within it.

AI co-scientists. Autonomous research systems generate hypotheses, run experiments, and draft manuscripts [29, 50, 42, 45, 38, 31, 34, 20]. They establish the multi-agent pattern we adopt, but what they deliver is a hypothesis or a paper; the visual evidence a human needs to trust either is out of scope.

Generation, tools, and prototypes. Chart-level systems map language to a specification and are judged on whether that specification is correct [35, 40, 19, 47, 30]. Tool-mediated SciVis agents drive an existing application, so the design space is fixed by the host tool and the agent's job is state selection [37, 28, 5, 41, 18, 10]; recent work packages this expertise as reusable agent skills [4] and studies which agent designs suit scientific visualization [43]. Reporting agents produce insight narratives with provenance [21, 49, 32]. Closest to us are systems that assemble visual analytics prototypes [53, 48], but these are built around iterative human steering: the analyst supplies task decomposition and design correction as the system runs. We argue the same target should also be attempted without those turns, precisely so the harness's contribution can be measured.

Critique, benchmarks, and evaluation. A parallel line asks what models know about visualization: extracting design preferences [44], giving design feedback [39], detecting misleading charts [15], and generating literacy items [14]. Literacy instruments [25, 36, 16] have been repurposed to probe LLM chart understanding [9, 23, 27, 51], while an evaluation-centric paradigm [2], SciVisAgentBench [3], and VisEval [13] argue that scalable evaluation gates progress. An interactive application has no single gradable answer, which is why we combine an ablation, a structured audit, and external contest judging instead.

3 What a VIS Co-Scientist Must Do

Discussions of AI co-scientists concentrate on hypothesis generation [20]. Visualization adds a different set of demands. It serves another domain, so the system must understand the scientific context well enough to choose useful representations. Its targets are often less quantitative than those used to score prediction or retrieval systems, since visualization quality has no single accuracy metric. It is also human-centered: the artifact is only useful if a person can act on it. We define a VIS co-scientist as a research partner that can independently (Fig. 2): (1) acquire enough domain understanding to turn a stated goal into concrete requirements; (2) conduct self-directed exploratory data analysis and reasoning; (3) design and implement a complete interactive visualization tool; and (4) assess the quality and robustness of its own analysis and communicate uncertainty well enough to support a human decision. Each capability is difficult for current agentic systems. This paper concerns the first three; the fourth remains the deepest open problem and is discussed in Sec. 5.

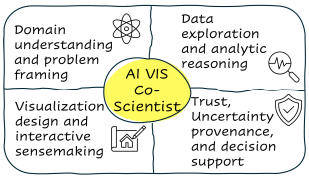


Figure 2: Key capabilities for an AI VIS co-scientist.

3.1 A Multi-Agent Harness

We realize the VIS co-scientist as a *harness*: a structured execution environment around a main LLM code agent that fixes agent roles, tool access, artifact handoffs, and evaluation loops. Given data and a task description, it runs the stages in Fig. 1 to a final VIS App. Each stage emits an artifact that either constrains the next stage or supplies material for later verification. This artifact-based design follows current guidance on long-running agent design [8]. We describe it as roles and contracts so it can be reimplemented in a different session, independent of the LLM's context window size.

Orchestrator. The Orchestrator owns the run from end to end: it reads the task materials, launches subagents, integrates their artifacts, and writes most of the application code, targeting a single-page VIS App with coordinated views, shared interaction state, and a desktop-style layout. It fixes routine problems itself and delegates bounded ones, both for specialized attention and to keep its own context focused.

Exploratory Data Analyzer. Before any design happens, the EDA agent profiles the dataset: schema, scale, missing values, formats, access constraints, and task-relevant variables. Its report is the empirical basis for planning, so the design loop reasons about the data as it is rather than as the metadata describes it.

Environment Builder and Performance Streaming Engineer. The Environment Builder selects minimal dependencies for data access, visualization, and browser validation and records installation details. When a run breaks because of data access, dependencies, rendering, or interaction logic, this subagent can repair the code environment directly. For large or remote data, a Performance Streaming Engineer handles progressive, resolution-adaptive access. We added this role because general coding ability was not enough when correctness depended on indexing, resolution, streaming, and provenance.

Main VIS design loop. Three roles iterate. The Planner turns the task and the EDA report into a specification covering encodings, coordinated panels, interaction state, risks, and validation checks. The VIS Designer implements or repairs demanding views: volume rendering, progressive streaming, custom brushing, linked selection. The Evaluator exercises the live application against the task goals and returns *Feedback and Fixes*, which the Orchestrator folds in, repeating until no blocking issue remains. This iteration is internal; it is not user interaction.

Skills and connectors. The harness extends the base agent with custom skills [7] and MCP [6] connectors that can be accessed by all subagents during execution: a *VIS Skill* and *SciVis Domain Skill* for design conventions and domain data handling, a *Linked-View Design Skill* for coordination patterns, a *Frontend Skill* for library and interaction choices, and *Playwright-MCP* [33] for browser inspection, screenshots, console checks, and scripted interaction. These resources address the often non-quantitative aspects of visualization research because they determine what the agents can *inspect and verify*, not only what they can generate. They are also where we put visualization expertise the base model lacks [4].

3.2 Artifacts and Layered Validation

Agents communicate through explicit artifacts rather than shared conversation history: the EDA report, the design and implementation report, the code environment, the VIS App, and the Evaluator's feedback and fixes. These artifacts act as contracts between stages and leave a trail a human can audit afterward, which is the main support for reproducibility in the design.

Validation is layered. Mechanistically, Playwright-MCP loads the live application, exercises controls, checks linked-view synchronization, captures screenshots, and reports console errors, catching blank panels, wrong color mappings, broken brushing, and layout overflow. Beyond that, the Evaluator judges whether a *sequence* of interactions addresses the analysis tasks, and scores the system on task achievability. This is harder than validating an ordinary web application, where correctness is largely visible in layout, text, and state, and it is where our design is weakest: progressive and asynchronous views introduce time-dependent state that final-state inspection cannot see.

3.3 Why Zero-Shot and an End-to-End Agentic Loop

A run receives one user prompt naming the data and the task, e.g., "solve the tasks in this SciVis contest and provide a visualization solution." There is no human intervention after launch, no worked example, and no task-specific design guidance. The agents iterate internally; the user does not.

The user has no visualization expert. The domain scientists who most need a VIS App are often the least equipped to specify one. They can state a scientific question and supply data, but not the encodings, view coordination, and interaction design that turn those inputs into a usable tool. In many labs there is no visualization researcher to ask. That scarcity sets the requirement: the agent must act as the visualization expert and get as far as it can from an underspecified request, rather than eliciting design intent from someone who does not have it to give. Autonomy also changes the economics. An hour-long run is practical as a background job launched in parallel across tasks, but not as a supervised session.

Attribution. The same property is what makes the result measurable. In a multi-turn session, the analyst's corrections ("use a diverging scale," "link the scatterplot to the map," "that panel is unreadable") are themselves visualization design decisions, and they are precisely the expertise the system is supposed to contribute. A result produced that way measures operator and agent jointly with no way to separate them. Zero-shot operation is the only setting in which an outcome is attributable to the harness.

Reproducibility and comparison. A fixed prompt can be restated in a paper and replayed against another model or harness; a multi-turn transcript cannot, since it varies with the operator's skill. Where agentic visualization lacks agreed evaluation protocols [2], fixing the human contribution at zero is a useful common denominator.

Diagnostic failures. Under this constraint, a failure cannot be waved away as poor prompting. When our experiments find missing legends and ambiguous color semantics (Sec. 4), the result tells us what the model and harness do *by default*. That is the actionable quantity for whoever builds the next harness.

What we are not claiming. Zero-shot is a capability requirement and a measurement regime, not a claim that the human should be shut out. We do not claim autonomous agents should replace visualization researchers or domain scientists; the audit below is partly an argument that they should not. Where a visualization expert is available, human-agent collaboration remains the right interaction model [24, 1, 52], and a system that clears the zero-shot bar is a better collaborator, because the human's turns are then spent on judgment rather than on supplying a design the system should already have proposed. The requirement is that the system must not *depend* on those turns, because for most of its intended users they will never come.

4 Capability Study

IEEE SciVis Contests are a good test case: real scientific data, underspecified goals, and tasks authored by someone other than the system's builders. Every run below used the zero-shot protocol of Sec. 3.3.

Setup and ablation. The 2021–2025 runs use OpenAI Codex with GPT-5.4 as the coding model. The *baseline* is a combined ablation of the harness, not a different system: the same model, the same reasoning-effort setting, and the same prompt, with the custom skills, MCP connectors, and subagents disabled together. The comparison measures the contribution of the harness as a whole rather than any single mechanism.

4.1 Case Study: 2025 SciVis Contest on Materials Discovery

The 2025 contest asks participants to support discovery of recycled aluminum alloy candidates in a high-dimensional simulation dataset. The task requires a design-space overview, explanations of composition–phase–property relationships, and interactive steering for multi-objective optimization. Under ablation, the base agent produced standalone plots and correlation analysis that partially addressed the tasks: useful analysis artifacts, but not an integrated interactive design. In several cases the agent silently failed to follow stated instructions because nothing in the loop checked the running result (supplemental material). The ablation separates partial analysis generation from an integrated, testable visualization system.

Using our system, the full run took roughly two hours and 5.6M tokens (369K input, 5.18M cached, 58K output). The resulting VIS App (Fig. 3) combines a candidate overview, a PCA embedding mode, a scope-aware relationship matrix, a composition–phase–property pathway panel, and a ranking area, linked through a shared filter with brushing.

The EDA stage did substantive work. It found that the authoritative source was not the stale CSV named in the metadata but a tab-delimited file with 324,632 candidate alloys and 70 variables once padded blank columns were removed; it dropped fully empty variables, created a stable candidate identifier, and selected `YS(MPa)` against `El.conductivity(S/m)` as the primary trade-off view. The application organized the design space around a strength–conductivity Pareto front, and the matrix and pathway views suggested strength rises with Si, Ni, and `Vf_DIAMOND_A4` while conductivity rises with Al and `Vf_FCC_A1`. The final Evaluator pass con-



Figure 3: Zero-shot output for the 2025 materials-discovery case study: candidate overview, embedding view, relationship matrix, pathway panel, and ranking view linked in one workspace.

Table 2: Structured expert self-audit of the 2025 case study, 1–5 Likert (1 = unacceptable, 5 = excellent), five raters.

Question	Mean	Range
EDA report makes correct assumptions about the data and tasks	4.6	4–5
EDA identifies key implementation insights and constraints	4.6	4–5
Proposed design follows established visualization principles	4.0	3–5
Evaluation documents give logical next-iteration guidance	4.4	4–5
Visual encodings are appropriate for the data and intent	3.8	2–5
Interface is visible and responsive for exploration	4.0	3–5
Coordinated views support sensemaking	4.6	4–5
Dashboard enables stated contest tasks	4.2	3–5
Removing all-NaN fields was correct	5.0	5–5
`KS1295[%]` improves strength but hurts thermal/electrical performance	3.4	2–5
Strength and thermal/electrical performance form a fundamental trade-off	5.0	5–5
Hardness and yield strength are strongly correlated	5.0	5–5
`6082[%]` strongly influences Cu, Ti, and Zn	2.4	1–4

firmed that brushing the trade-off view re-scoped the matrix, pathway, and ranking, and that embedding mode cleared that scope.

4.2 Structured Expert Self-Audit

As a structured self-audit beyond the agent's own Evaluator, five visualization researchers (included in the author team) rated the run on 13 items (Table 2) covering the artifacts, mechanical usability, whether the generated views supported the stated tasks, and whether domain insights curated from the winning entries' reports could be recovered from the tool [3]. Raters saw all artifacts and the live VIS App; the instrument is in the supplemental material.

The audit separates task completion from design quality, and the split is consistent. EDA assumptions, implementation insights, coordinated views, and task coverage all score at or above 4.2, and open comments named brushing, filter propagation, and candidate ranking as strengths. Encoding and interface polish score lower and vary more: missing legends, ambiguous color semantics, dense layout, limited variable control, and reduced responsiveness after brushing. The output also lacked the encoding novelty that characterizes contest winners; we read this as a difference in optimization target rather than a defect, since the harness aims to complete the stated tasks with established encodings.

The domain-insight items show a similar pattern. Raters unanimously endorsed removing all-NaN fields, the strength–performance trade-off, and the hardness–yield relationship, but agreement was weak on the `KS1295[%]` claim and low on the `6082[%]` claim. Automatically surfaced correlations can be stated with more confidence than their provenance supports. This is the failure mode most relevant to the fourth capability in Sec. 3. We use the audit to identify failure modes, not as independent validation: the five raters are members of the author team, which is its principal limitation.

4.3 Transfer and External Judging

We applied the same harness and protocol to the 2021, 2023, 2024, and 2026 contests, spanning climate science, sonar imaging, neuroscience, and mantle convection (supplemental material). The runs all produced mechanically valid VIS Apps across different formats, questions, and interaction demands, so the harness is not tuned to the materials case; results were strongest where the task decomposes into data profiling, established encodings, coordinated views, and task-specific interaction.

Because a self-administered audit can only go so far, we also submitted a zero-shot entry to the *ongoing* 2026 IEEE SciVis Contest covering all four tasks with live remote streaming of the simulation data; those runs used GPT-5.5 and took roughly 1.5 hours and 43M tokens each. External judging will provide independent assessment, but the outcome is not yet known by this deadline. We treat these runs as transfer demonstrations rather than validation. Their qualitative pattern matches the audit: the agents need explicit pressure to include legends, units, and value ranges, and they gravitate toward familiar dashboard encodings over more ambitious scientific visualization designs. The VIS co-scientist and harness code is available here: *(under internal institutional review)*.

5 Discussion and Outlook

The study above maps current capability; it is not a claim that autonomous agents can replace visualization experts. Given a dataset and one sentence, a harnessed code agent can profile data, derive requirements, implement coordinated views, validate them in a browser, and deliver a functional VIS App. The same model without the harness falls back to much simpler outputs.

Two failures recurred in every run. First, encoding quality: agents assemble safe, established designs and omit the legends, units, and value ranges that make a scientific claim readable, as seen in the audit's 3.8 on encoding appropriateness and in the 2026 runs. Second, perception of dynamic behavior: screenshots and DOM inspection confirm that an interface rendered, but they weakly test sequences of camera movement, filtering, and linked-state change. Progressive queries also add asynchronous state that final-state inspection cannot see. An agent cannot yet critique the temporal behavior of its own interface, which is central to visualization design. These are visualization research problems as much as they are agent problems.

This points to where visualization expertise should enter a harness. Visualization researchers can author skills, validation hooks, and evaluation protocols, and these components determine what an agent can verify rather than only generate. For VIS research, the shift is from designing visualization artifacts alone toward designing and validating the process that produces them, and toward formalizing design knowledge.

References

[1] K. Ai, P. P. Do, and C. Wang. HiLSVA: Design and evaluation of a human-in-the-loop agentic system for scientific visualization. *arXiv preprint arXiv:2606.26614*, 2026. 2, 3

[2] K. Ai, H. Miao, Z. Li, C. Wang, and S. Liu. An evaluation-centric paradigm for scientific visualization agents. In *IEEE VIS Workshop on Visualization and Generative AI (VIS x GenAI)*, 2025. arXiv:2509.15160. 2, 3

[3] K. Ai, H. Miao, K. Tang, N. Gorski, J. Sun, G. Liu, H. I. Ingolfsson, D. Lenz, H. Guo, H. Yu, et al. SciVisAgentBench: A benchmark for evaluating scientific data analysis and visualization agents. *arXiv preprint arXiv:2603.29139*, 2026. 2, 4

[4] K. Ai, H. Miao, K. Tang, S. Liu, and C. Wang. Scivisagentskills: Design and evaluation of agent skills for scientific data analysis and visualization. *arXiv preprint arXiv:2606.05525*, 2026. 2, 3

[5] K. Ai, K. Tang, and C. Wang. NLI4VolVis: Natural language interaction for volume visualization via multi-LLM agents and editable 3D Gaussian splatting. *IEEE Transactions on Visualization and Computer Graphics*, 32(1):46–56, 2026. 2

[6] Anthropic. Announcements: Introducing the model context protocol. https://www.anthropic.com/news/model-context-protocol. 3

[7] Anthropic. Equipping agents for the real world with agent skills. https://claude.com/blog/equipping-agents-for-the-real-world-with-agent-skills, 2025. 3

[8] Anthropic. Harness design for long-running application development. https://www.anthropic.com/engineering/harness-design-long-running-apps, 2026. 2

[9] A. Bendeck and J. Stasko. An empirical evaluation of the GPT-4 multimodal language model on visualization literacy tasks. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1105–1115, 2025. 2

[10] A. Biswas, T. L. Turton, N. R. Ranasinghe, S. Jones, B. Love, W. Jones, A. Hagberg, H.-W. Shen, N. DeBardeleben, and E. Lawrence. VizGenie: Toward self-refining, domain-aware workflows for next-generation scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 32(1):1021–1031, 2026. doi: 10.1109/TVCG.2025.3634655 2

[11] M. Brossier, T. Isenberg, K. J. Schönborn, J. Unger, M. Romero, J. Björklund, A. Ynnerman, and L. Besançon. State of the art of llm-enabled interaction with visualization. *ArXiv*, abs/2601.14943, 2026. 1, 2

[12] D. Castelvecchi. Researchers built an 'AI scientist'—what can it do. *Nature*, 633(8029):266–266, 2024. 1

[13] N. Chen, Y. Zhang, J. Xu, K. Ren, and Y. Yang. VisEval: A benchmark for data visualization in the era of large language models. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1301–1311, 2025. 2

[14] Y. Cui, L. W. Ge, Y. Ding, L. Harrison, F. Yang, and M. Kay. Promises and pitfalls: Using large language models to generate visualization items. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 2

[15] A. K. Das and K. Mueller. MisVisFix: An interactive dashboard for detecting, explaining, and correcting misleading visualizations using large language models. *arXiv preprint arXiv:2508.04679*, 2025. 2

[16] P. P. Do, K. Tang, K. Ai, and C. Wang. SVLAT: Scientific visualization literacy assessment test. *arXiv preprint arXiv:2603.19000*, 2026. 2

[17] S. Esmaeili, S. Kabir, A. M. Colas, R. P. Linder, and E. D. Ragan. Evaluating graphical perception of visual motion for quantitative data encoding. *IEEE Transactions on Visualization and Computer Graphics*, 29(12):4845–4857, 2022. 1

[18] N. Gorski, S. Liu, and B. Wang. TopoPilot: Reliable conversational workflow automation for topological data analysis and visualization. *arXiv preprint arXiv:2603.25063*, 2026. 2

[19] K. Goswami, P. Mathur, R. Rossi, and F. Dernoncourt. PlotGen: Multi-agent LLM-based scientific data visualization via multimodal retrieval feedback. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 1672–1676, 2025. doi: 10.1145/3701716.3716888 2

[20] J. Gottweis, W.-H. Weng, A. Daryin, T. Tu, P. Sirkovic, A. Myaskovsky, G. Glowaty, F. Weissenberger, A. Orlandi, D. Popovici, A. Palepu, K. Rong, R. Tanno, K. Saab, F. Zhang, J. Blum, A. Carroll, K. Kulkarni, N. Tomašev, D. Zverinski, I. Rendulic, E. Vedadi, F. Hasler, L. Rimanic, M. Boia, I. Budiselic, B. Feinstein, M. Bellaiche, T. Sheffer, J. Freyberg, J. Ratcliff, O. Bertolli, K. Chou, A. Hassidim, B. Gokturk, A. Vahdat, Y. Guan, V. Dhillon, E. D. Vaishnav, B. Lee, T. R. D. Costa, J. R. Penadés, G. Peltz, Y. Matias, J. Manyika, D. Hassabis, Y. Xu, P. Kohli, A. Pawlosky, A. Karthikesalingam, and V. Natarajan. Accelerating scientific discovery with co-scientist. *Nature*, 655(8122):487–496, May 2026. doi: 10.1038/s41586-026-10644-y 1, 2

[21] P. F. Gyarmati, D. Moritz, T. Möller, and L. Koesten. A composable agentic system for automated visual data reporting. In *IEEE VIS Workshop on Visualization and Generative AI (VIS x GenAI)*, 2025. 2

[22] J. Hao, M. Yang, Q. Shi, Y. Jiang, G. Zhang, and W. Zeng. FinFlier: Automating graphical overlays for financial visualizations with knowledge-grounding large language model. *IEEE Transactions on Visualization and Computer Graphics*, 2025. arXiv:2412.06821. 2

[23] J. Hong, C. Seto, A. Fan, and R. Maciejewski. Do LLMs have visualization literacy? an evaluation on modified visualizations to test generalization in data interpretation. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):7004–7018, 2025. 2

[24] R. Koonchanok, A. Kale, and K. Reda. Assist, don't analyze: Designing a lightweight GenAI interface for visual data analysis. In *IEEE VIS Workshop on Visualization and Generative AI (VIS x GenAI)*, 2025. 3

[25] S. Lee, S.-H. Kim, and B. C. Kwon. VLAT: Development of a visualization literacy assessment test. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):551–560, 2016. 2

[26] Y. Li, E. D. Berger, M. Kahng, and C. X. Bearfield. From perception to decision: Assessing the role of chart type affordances in high-level decision tasks. In *Proceedings of IEEE VIS Conference (Short Papers)*, pp. 366–370, 2025. 1

[27] Z. Li, H. Miao, X. Yan, V. Pascucci, M. Berger, and S. Liu. See or recall: A sanity check for the role of vision in solving visualization question answer tasks with multimodal LLMs. *arXiv preprint arXiv:2504.09809*, 2025. 2

[28] S. Liu, H. Miao, and P.-T. Bremer. ParaView-MCP: An autonomous visualization agent with direct tool use. In *Proceedings of IEEE VIS Conference (Short Papers)*, pp. 61–65, 2025. 2

[29] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024. 1, 2

[30] J. Lu, Y. Song, C. Zhang, and R. C.-W. Wong. MultiVis-Agent: A multi-agent framework with logic rules for reliable and comprehensive cross-modal data visualization. *arXiv preprint arXiv:2601.18320*, 2026. 2

[31] Y. Lyu, X. Zhang, X. Yi, Y. Zhao, S. Guo, W. Hu, J. Piotrowski, J. Kaliski, J. Urbani, Z. Meng, et al. EvoScientist: Towards multi-agent evolving AI scientists for end-to-end scientific discovery. *arXiv preprint arXiv:2603.08127*, 2026. 2

[32] W. Meng, Y. Guo, Z. Xie, and K. Xu. What happened to automated visualization? an agentic analysis. In *IEEE VIS Workshop on Visualization and Generative AI (VIS x GenAI)*, 2025. 2

[33] Microsoft. Playwright MCP documentation. https://playwright.dev/docs/getting-started-mcp, 2024. Accessed: 2026-04-29. 3

[34] L. Mitchener, A. Yiu, B. Chang, M. Bourdenx, T. Nadolski, A. Sulovari, E. C. Landsness, D. L. Barabasi, S. Narayanan, N. Evans, et al. Kosmos: An AI scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025. 1, 2

[35] G. Ouyang, J. Chen, Z. Nie, Y. Gui, Y. Wan, H. Zhang, and D. Chen. nvAgent: Automated data visualization from natural language via collaborative agent workflow. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pp. 19534–19567, 2025. doi: 10.18653/v1/2025.acl-long.960 2

[36] S. Pandey and A. Ottley. Mini-VLAT: A short and effective measure of visualization literacy. *Computer graphics forum*, 42(3):1–11, 2023. 2

[37] T. Peterka, T. Mallick, O. Yildiz, D. Lenz, C. Quammen, and B. Geveci. ChatVis: Large language model agent for generating scientific visualizations. In *Proceedings of IEEE Workshop on Large Data Analysis and Visualization*, pp. 22–32, 2025. 2

[38] C. Shao, D. Huang, Y. Li, K. Zhao, W. Lin, Y. Zhang, Q. Zeng, Z. Chen, T. Li, Y. Huang, et al. OmniScientist: Toward a co-evolving ecosystem of human and AI scientists. *arXiv preprint arXiv:2511.16931*, 2025. 2

[39] S. Shin, S. Hong, and N. Elmqvist. Visualizationary: Automating design feedback for visualization designers using LLMs. *IEEE Transactions on Visualization and Computer Graphics*, 2025. arXiv:2409.13109. 2

[40] Z. Shuai, B. Li, S. Yan, Y. Luo, and W. Yang. DeepVIS: Bridging natural language and data visualization through step-wise reasoning. *arXiv preprint arXiv:2508.01700*, 2025. 2

[41] J. Sun, D. Lenz, T. Peterka, and H. Yu. SASAV: Self-directed agent for scientific analysis and visualization. *arXiv preprint arXiv:2604.03406*, 2026. 2

[42] J. Tang, L. Xia, Z. Li, and C. Huang. AI-researcher: Autonomous scientific innovation. *arXiv preprint arXiv:2505.18705*, 2025. 2

[43] J. E. Vonderhorst, K. Ai, H. Miao, S. Liu, and C. Wang. Exploring LLM agent designs and interaction modalities for scientific visualization. *arXiv preprint arXiv:2604.27996*, 2026. 2

[44] H. W. Wang, M. Gordon, L. Battle, and J. Heer. DracoGPT: Extracting visualization design preferences from large language models. *IEEE Transactions on Visualization and Computer Graphics*, 2025. arXiv:2408.06845. 2

[45] Y. Weng, M. Zhu, Q. Xie, Q. Sun, Z. Lin, S. Liu, and Y. Zhang. Deep-

Scientist: Advancing frontier-pushing scientific findings progressively. *arXiv preprint arXiv:2509.26603*, 2025. 2

[46] A. Wolter, G. Vidalakis, M. Yu, A. Grover, and V. Dhanoa. Multi-agent data visualization and narrative generation. In *IEEE VIS Workshop on Visualization and Generative AI (VIS x GenAI)*, 2025. 2

[47] L. Xie, Y. Lin, C. Liu, H. Qu, and X. Shu. DataWink: Reusing and adapting SVG-based visualization examples with large multimodal models. *IEEE Transactions on Visualization and Computer Graphics*, 2025. arXiv:2507.17734. 2

[48] J. Xiong, L. Chen, X. Dou, S. Wang, K. Lin, S. Yang, and Q. Li. LLM4VA: A multi-agent LLM framework for automated prototyping of visual analytics systems. In *IEEE VIS Workshop on Visualization and Generative AI (VIS x GenAI)*, 2025. 2

[49] J. Xiong, X. Xiao, H. Zhang, and Q. Li. DataMinds: A generalizable role-based multi-agent framework for automated data-to-report generation. In *IEEE VIS Workshop on Visualization and Generative AI (VIS x GenAI)*, 2025. 2

[50] Y. Yamada, R. T. Lange, C. Lu, S. Hu, C. Lu, J. Foerster, J. Clune, and D. Ha. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025. 1, 2

[51] X. Zeng, H. Lin, Y. Ye, and W. Zeng. Advancing multimodal large language models in chart question answering with visualization-referenced instruction tuning. *IEEE Transactions on Visualization and Computer Graphics*, 2025. arXiv:2407.20174. 2

[52] Y. Zhang, Y. Zhao, Y. Guo, G. Li, S. Chen, and X. Yuan. Position paper: Validatory visualization for human-GenAI co-creation. In *IEEE VIS Workshop on Visualization and Generative AI (VIS x GenAI)*, 2025. 1, 3

[53] Y. Zhao, J. Wang, L. Xiang, X. Zhang, Z. Guo, C. Turkay, Y. Zhang, and S. Chen. LightVA: Lightweight Visual Analytics With LLM Agent-Based Task Planning and Execution . *IEEE Transactions on Visualization & Computer Graphics*, 31(09):6162–6177, Sept. 2025. doi: 10.1109/TVCG.2024.3496112 2