Evaluating System Design Choices in Biomedical AI Agents

Ali Saadat [1] **Jacques Fellay** [1]

Abstract

Tool-augmented biomedical agents are increasingly being proposed as systems that can plan analyses, interact with computational resources, and return scientific answers from heterogeneous data. However, the design choices that make such agents reliable remain poorly characterized. We report a preliminary ablation study of the Biomni agent on the BixBench-Verified-50 dataset, focusing on two practical components of an agentic workflow: whether the agent uses a tool retriever, and whether a critic is inserted in the reasoning loop. We evaluate two LLM models, GPT-4.1 mini and GPT-4.1, across six critic–retriever settings. The strongest condition was GPT-4.1 with tool retrieval and an end-of-run critic, which achieved 60% accuracy. Overall, our results provide preliminary evidence that the placement and interaction of agentic components can substantially affect both reliability and efficiency in biomedical AI workflows.

1. Introduction

AI agents are increasingly being developed as systems that can move beyond single-step prediction or text generation toward goal-directed scientific work (Gottweis et al., 2025). Rather than producing only a static answer, an agent can decompose a task, inspect input data, select and invoke external tools, write and execute code, evaluate intermediate results, and revise its approach when the initial attempt fails (Ferrag et al., 2026). These capabilities are especially relevant in computational biology and bioinformatics, where many questions require coordinated reasoning over heterogeneous data formats, specialized software, public databases, and domain-specific assumptions. Recent surveys describe this transition as a shift from standalone predictive models toward autonomous systems capable of planning, tool use, reasoning, and iterative interaction with biological resources (Gao et al., 2024; Qi et al., 2026).

Agentic analysis raises design questions that are less central in conventional generative modeling. An agent must decide many factors such as when to rely on built-in tools, when to write custom analysis code, which data sources to inspect, and how to recover from incomplete or erroneous intermediate reasoning Abou Ali et al., 2025. Recent biological agents explore different points in this design space. GeneGPT showed that language models can be augmented with domain-specific APIs by using NCBI tools to answer genomics questions (Jin et al., 2024). BioAgents proposed a multi-agent framework for bioinformatics workflows built on smaller fine-tuned language models and retrieval-augmented generation (Mehandru et al., 2025). Kosmos developed a broader AI-scientist architecture for long-horizon, data-driven discovery, combining data-analysis agents, literature-search agents, and a structured world model (Mitchener et al., 2025b). Biomni integrates large-language-model reasoning with retrieval over biomedical tools, software, databases, and code-execution environments (Huang et al., 2025). These systems illustrate that agent performance depends not only on the underlying language model, but also on the surrounding workflow architecture.

A growing body of work has begun to evaluate language models and agents on scientific and biomedical tasks. Several recent benchmarks focus specifically on computational-biology and bioinformatics agents. CompBioBench introduces diverse, verifiable computational-biology tasks across genomics, transcriptomics, epigenomics, single-cell analysis, human genetics, and machine-learning workflows (Nair et al., 2026). BioAgent Bench evaluates robustness on end-to-end bioinformatics pipelines such as RNA-seq, variant calling, and metagenomics, including perturbation settings with corrupted inputs, decoy files, and prompt bloat (Fa et al., 2026). GenomeArena reflects the increasing emphasis on realistic genomic analysis tasks rather than isolated biological facts (Sokolova et al., 2025). BixBench evaluates LLM-based agents on open-ended computational-biology tasks that require agents to explore biological datasets, perform multi-step analyses, and interpret resulting outputs (Mitchener et al., 2025a). In this work, we use BixBench-Verified-50, a curated subset designed for more reliable evaluation, as a compact but general testbed for biomedical

[1]School of Life Sciences, Ecole Polytechnique Fédérale de Lausanne, Switzerland. Correspondence to: Jacques Fellay <jacques.fellay@epfl.ch>.

Accepted as a workshop paper at ICML 2026.

data-analysis questions (Phylo, 2026).

We report an ablation study of Biomni on BixBench-Verified-50, focusing on two practical components of an agentic workflow: the use of a tool retriever and the placement of a critic within the reasoning loop. A recent study benchmarked biological agents on single-cell tasks and analyzed critic and retriever components within Biomni (Panigrahi et al., 2026). We build on this direction in a broader bioinformatics setting by evaluating Biomni on BixBench-Verified-50 and studying how critic placement interacts with tool retrieval. We also report metrics stratified by the grading modes (i.e. verifier group). Beyond accuracy, we measure runtime, model-token cost, and LLM-call count, since reliability gains are most useful in practice when their computational and economic trade-offs are understood. Overall, our goal is to clarify how these architectural choices affect the reliability and efficiency of biomedical AI workflows.

2. Methods

2.1. Benchmark and Execution Protocol

We evaluated Biomni on BixBench-Verified-50. Each benchmark instance includes a natural-language question, a reference answer, an evaluation mode, and a data capsule with the input files needed to complete the analysis. The 50 questions are divided into three verifier groups: `llm_verifier`, which uses LLM-based judging for 20 questions; `str_verifier`, which uses exact-string matching for 17 questions; and `range_verifier`, which uses numeric range checks for 13 questions. We report aggregate results in the main text and provide verifier-stratified results in the appendix.

Runs were isolated at the item level. Each question was executed in a separate workspace, which preserved the corresponding prompts, transcripts, intermediate files, generated artifacts, and run metadata for later inspection. For every benchmark item, the agent was given the question together with the associated data capsule. The instruction prompt asked the agent to carry out the necessary analysis and place its final response inside a `<solution>...</solution>` tag. Once the run finished, we extracted the solution and submitted that extracted answer to the benchmark evaluator.

2.2. Agent Design Ablations

We studied two agent-design factors including critic placement and tool retrieval. These factors target separate parts of the agentic workflow. The critic controls whether and when an additional review step is introduced, while the retriever controls whether the agent is first given a task-specific subset of available resources before it begins planning.

Critic placement was evaluated under three settings including 1) no critic: In this setting, Biomni proceeded through the task without an explicit secondary review stage. 2) end-critic: A separate critic module examined the agent's completed work after the agent had decided to stop. The critic could flag issues such as incomplete analyses, weakly supported conclusions, or incorrectly formatted outputs, after which the agent could revise or continue its workflow. 3) plan critic: the critic instead operated earlier. It reviewed the agent's initial plan before execution and could suggest changes before any analysis steps were run.

Tool retrieval was evaluated under two settings: 1) With retrieval enabled, Biomni used an LLM-based retriever before planning to identify a task-relevant subset of tools, software packages, and databases from the broader Biomni environment. This step narrows the set of resources exposed to the planner, reducing the size of the action space before the agent generates its initial plan. 2) With retrieval disabled, the preliminary resource-selection step was omitted, so the agent planned and acted without the same task-specific filtering of available tools.

These settings yielded six ablation conditions for each LLM used by Biomni. We evaluated all six conditions with GPT-4.1 mini and GPT-4.1 as Biomni's underlying models.

2.3. Scoring and Accounting

We measured accuracy, runtime, model cost, and LLM-call count. Accuracy is computed as the number of correct answers divided by the total number of questions. For `llm_verifier` items, the benchmark evaluator uses GPT-5.4 as the judge model to compare each extracted answer against the corresponding ground-truth answer. For `str_verifier` and `range_verifier` items, correctness is determined by exact-string matching and numeric range checks, respectively.

Runtime is computed per question by measuring elapsed wall-clock time for each completed run. Model cost is computed from the recorded input and output token counts for each run using the corresponding model prices: \$0.40 per million input tokens and \$1.60 per million output tokens for GPT-4.1 mini, and \$2.00 per million input tokens and \$8.00 per million output tokens for GPT-4.1. These cost estimates include only model usage by the agent and exclude judge-model grading cost, local compute, storage, and any external costs. LLM-call count is computed as the number of LLM invocations captured during the agent run.

3. Results

3.1. Accuracy Across Critic and Retrieval Settings

Figure 1 summarizes accuracy, runtime, estimated cost, and LLM-call count across model, critic, and retrieval settings. Across all 50 manifest questions, the mean accuracy across the 12 conditions was 35.2%, ranging from 8.0% to 60.0%. For both GPT-4.1 mini and GPT-4.1, the best-performing configuration used an end critic with the tool retriever enabled. Under this setting, GPT-4.1 achieved the highest overall accuracy, reaching 60% accuracy.

For GPT-4.1 mini, adding an end critic improved accuracy relative to the no-critic baseline regardless of whether retrieval was enabled: accuracy increased from 38% to 44% with retrieval, and from 31% to 40% without retrieval. For GPT-4.1, however, the effect of the end critic depended on retrieval. With retrieval enabled, accuracy increased from 40% to 60%; without retrieval, it decreased from 52% to 40%. By contrast, the plan-stage critic performed poorly for both models, generally underperforming both the no-critic and end-critic configurations. Verifier-stratified accuracy, runtime, cost, and LLM-call results are provided in Appendix A.

3.2. Runtime, Cost, and LLM Calls

End-critic conditions were generally slower because they add a post-analysis critique step (Figure 1 b). Agent using GPT-4.1 was overall faster in this result set, averaging 0.62 minutes per question across ablations compared with 1.03 minutes for GPT-4.1 mini.

Tool retrieval reduced model-token cost in most matched comparisons (Figure 1 c). For GPT-4.1 mini, disabling retrieval increased cost per question from \$0.044 to \$0.129 under no critic and from \$0.072 to \$0.228 under the end critic. For GPT-4.1, disabling retrieval increased cost from \$0.173 to \$0.538 under no critic and from \$0.265 to \$0.884 under the end critic. .

LLM-call counts provide a complementary measure of agent effort (Figure 1 d). Overall the agent used a mean of 10.02 LLM calls per question and a median of 9.00 calls. We observed that end-critic accuracy gains often came with more LLM calls, which increases runtime and cost.

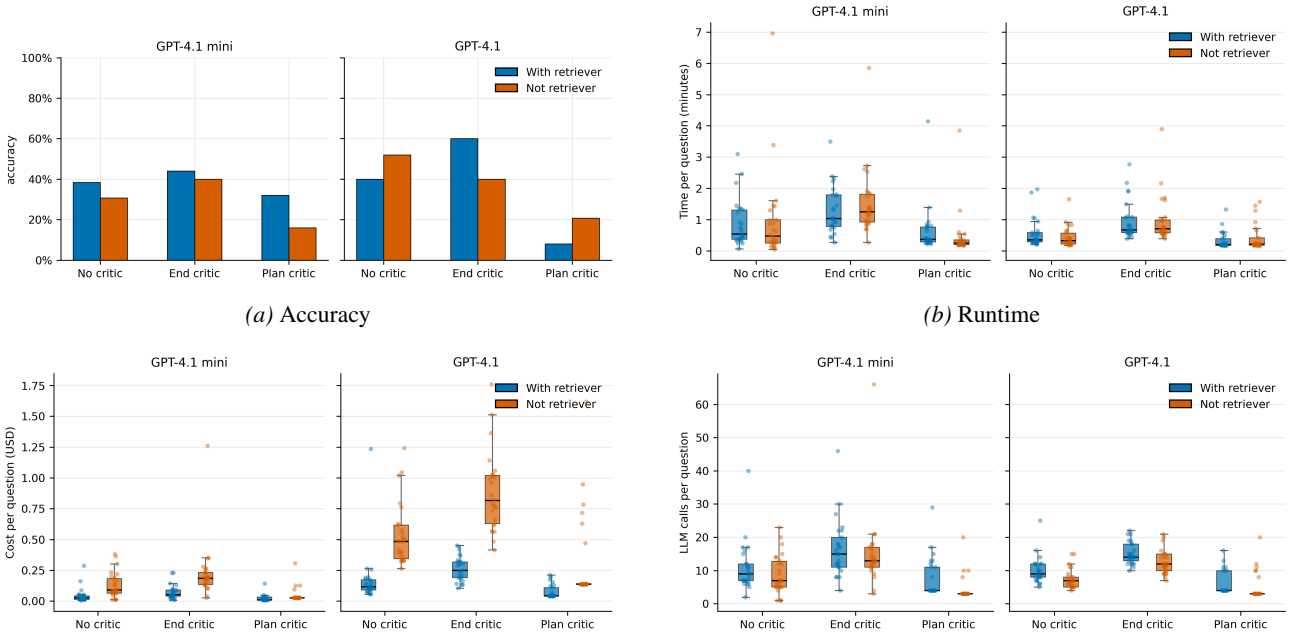
4. Discussion

This preliminary study adds to the growing literature on evaluating biomedical and scientific agents (Nair et al., 2026; Fa et al., 2026; Sokolova et al., 2025; Mitchener et al., 2025a). Prior benchmarks have shown that static biomedical knowledge evaluation is insufficient for measuring research usefulness, and that realistic agent benchmarks should assess planning, data exploration, tool use, code execution, and interpretation (Laurent et al., 2024; Wang et al., 2025; Mitchener et al., 2025a). Our results reinforce this view. Holding the benchmark and base agent fixed, we observed considerable differences in accuracy, runtime, cost, and LLM-call count depending on critic placement and tool retrieval. These findings suggest that biomedical-agent evaluation should treat the agent loop itself as an object of study, rather than as a secondary implementation detail around an LLM (Xu, 2026). The best setting reached 60% accuracy, which remains too low for autonomous biomedical analysis but is still meaningful for an open-ended benchmark with heterogeneous data-analysis tasks. This suggests that the surrounding agentic workflow can be an important source of capability. A simpler non-reasoning model equipped with tool use and critique can approach the performance of stronger reasoning models evaluated without comparable agentic scaffolding (Phylo, 2026).

Critic placement is a first-order design choice. Across our experiments, end-of-run criticism was more reliable than plan-stage criticism. This pattern is consistent with the structure of many biomedical data-analysis tasks: important errors often become visible only after the agent has inspected files, inferred schemas, run computations, and generated intermediate outputs. Before execution, a plan critic can only evaluate an abstract proposal. It may not yet know dataset-specific details such as column names, missing values, file encodings, sample identifiers, or the aggregation level required by the question. By contrast, an end critic can evaluate the final answer in light of the original question and the evidence produced during execution. This difference in available information may help explain why plan-stage criticism performed poorly, whereas GPT-4.1 with retrieval and an end critic achieved the strongest overall result.

This finding is also consistent with a recent study (Panigrahi et al., 2026), which studied critic and retriever components in Biomni on single-cell transcriptomic tasks and found that critic effects depend on both placement and the underlying model. Our results extend that observation to a broader bioinformatics setting. In both studies, criticism is not automatically beneficial. A critic appears useful only when it is placed at a point in the workflow where it has enough evidence to identify correctable errors without prematurely constraining the analysis. For biomedical agents, this suggests that "more reflection" is not necessarily better; the timing and information available to the critic matter.

Retrieval changes both efficiency and behavior. Tool retrieval reduced model-token cost in most matched comparisons. This is expected because retrieval narrows the tool and documentation context exposed to the planner, reducing the amount of irrelevant information carried through the interaction. However, retrieval had mixed effects on accu-



(a) Accuracy *(b)* Runtime

(c) Cost *(d)* LLM calls

Figure 1. Overall accuracy, runtime, estimated cost, and LLM-call count across all questions.

racy. For GPT-4.1 mini, retrieval improved accuracy in all three critic settings. For GPT-4.1, retrieval improved the end-critic condition but reduced accuracy in the no-critic and plan-critic settings. These results argue against treating retrieval as a uniformly beneficial add-on. Instead, retrieval should be understood as an intervention that changes the agent's action space and, consequently, its reasoning trajectory.

One possible explanation is that broad bioinformatics questions are not always best solved by invoking specialized tools. Some tasks may require direct data inspection, simple custom code, or careful interpretation of file contents. In these cases, retrieved tools or documentation can become overly prescriptive, nudging the agent toward a narrower computation than the question requires. Retrieval quality should therefore be evaluated not only by whether the selected tools are relevant, but also by whether the retrieved context preserves the agent's ability to choose a simpler or more general analysis strategy when appropriate.

Our manual inspection provides preliminary support for this concern. In several questions asking for median values, GPT-4.1 without retrieval produced the correct aggregate answer, whereas GPT-4.1 with retrieval produced an incorrect value that appeared consistent with a per-row, per-gene, or otherwise narrower computation. These examples are not sufficient to establish a general mechanism, but they illustrate a concrete failure mode: retrieval can make an agent more focused while also making it less flexible. Future work should characterize when retrieval improves task grounding and when it introduces misleading constraints.

Evaluations should include both accuracy and cost. The strongest accuracy gains were not free. End-critic configurations often required more LLM calls and higher model-token cost, because the agent performed an additional review step and sometimes continued execution after critique. This trade-off is important for biomedical workflows, where agents may be applied to large batches of analyses or interactive settings with limited compute budgets. A configuration that improves accuracy on difficult questions may be worthwhile for high-stakes analyses, but less attractive for routine screening tasks if the marginal gain is small relative to the added cost. These results therefore suggest that agent evaluation should report efficiency metrics alongside correctness, especially when comparing workflow-level interventions such as critics, retrievers, and self-revision loops.

Study limitations. This study has several limitations. The evaluation uses only 50 benchmark questions. Although BixBench-Verified-50 provides a compact and curated testbed, the sample size limits statistical power and makes individual task failures influential. The observed differences should therefore be interpreted as preliminary evidence rather than definitive estimates of component effects. Moreover, the ablations were performed only within Biomni, whose tool library, retriever, prompts, execution en-

vironment, and stopping behavior may shape the observed results. Our evaluation is further restricted to two OpenAI models used within Biomni. The interaction between critic placement, retrieval, and model capability may differ for other frontier models, open-weight models, or domain-specialized biomedical models. Finally, our cost accounting includes only model-token usage by the agent. It excludes local compute, storage, data-transfer overhead, external tool costs, and judge-model grading cost. As a result, the reported costs should be interpreted as approximate model-usage costs rather than full end-to-end operating costs.

Conclusion. We evaluated how critic placement and tool retrieval affect Biomni's performance on BixBench-Verified-50. Our results show that workflow architecture can substantially influence both the reliability and efficiency of biomedical agents. In particular, critic placement and retrieval altered not only accuracy, but also runtime, cost, and LLM-call count. These findings suggest that biomedical agents should be evaluated not only by the underlying LLM, but also by the design of the surrounding reasoning loop. Future work should scale these ablations to larger benchmarks, additional models, and more systematic failure analyses to identify agent designs that more reliably support scientific analysis.

References

Abou Ali, M., Dornaika, F., and Charafeddine, J. Agentic ai: a comprehensive survey of architectures, applications, and future directions. *Artificial Intelligence Review*, 59 (1), November 2025. ISSN 1573-7462. doi: 10.1007/s10462-025-11422-4. URL http://dx.doi.org/10.1007/s10462-025-11422-4.

Fa, D., Culjak, M., Pandza, B., and Cupic, M. Bioagent bench: An ai agent evaluation suite for bioinformatics, 2026. URL https://arxiv.org/abs/2601.21800.

Ferrag, M. A., Tihanyi, N., and Debbah, M. From llm reasoning to autonomous ai agents: A comprehensive review, 2026. URL https://arxiv.org/abs/2504.19678.

Gao, S., Fang, A., Huang, Y., Giunchiglia, V., Noori, A., Schwarz, J. R., Ektefaie, Y., Kondic, J., and Zitnik, M. Empowering biomedical discovery with AI agents. *Cell*, 187(22):6125–6151, October 2024.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Palepu, A., Sirkovic, P., Myaskovsky, A., Weissenberger, F., Rong, K., Tanno, R., et al. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

Huang, K., Zhang, S., Wang, H., Qu, Y., Lu, Y., Roohani, Y., Li, R., Qiu, L., Li, G., Zhang, J., Yin, D., Marwaha, S., Carter, J. N., Zhou, X., Wheeler, M., Bernstein, J. A., Wang, M., He, P., Zhou, J., Snyder, M., Cong, L., Regev, A., and Leskovec, J. Biomni: A general-purpose biomedical AI agent. June 2025.

Jin, Q., Yang, Y., Chen, Q., and Lu, Z. GeneGPT: augmenting large language models with domain tools for improved access to biomedical information. *Bioinformatics*, 40(2):btae075, February 2024.

Laurent, J. M., Janizek, J. D., Ruzo, M., Hinks, M. M., Hammerling, M. J., Narayanan, S., Ponnapati, M., White, A. D., and Rodriques, S. G. Lab-bench: Measuring capabilities of language models for biology research, 2024. URL https://arxiv.org/abs/2407.10362.

Mehandru, N., Hall, A. K., Melnichenko, O., Dubinina, Y., Tsirulnikov, D., Bamman, D., Alaa, A., Saponas, S., and Malladi, V. S. BioAgents: Bridging the gap in bioinformatics analysis with multi-agent systems. *Sci. Rep.*, 15 (1):39036, November 2025.

Mitchener, L., Laurent, J. M., Andonian, A., Tenmann, B., Narayanan, S., Wellawatte, G. P., White, A., Sani, L., and Rodriques, S. G. Bixbench: a comprehensive benchmark for llm-based agents in computational biology, 2025a. URL https://arxiv.org/abs/2503.00096.

Mitchener, L., Yiu, A., Chang, B., Bourdenx, M., Nadolski, T., Sulovari, A., Landsness, E. C., Barabasi, D. L., Narayanan, S., Evans, N., Reddy, S., Foiani, M., Kamal, A., Shriver, L. P., Cao, F., Wassie, A. T., Laurent, J. M., Melville-Green, E., Caldas, M., Bou, A., Roberts, K. F., Zagorac, S., Orr, T. C., Orr, M. E., Zwezdaryk, K. J., Ghareeb, A. E., McCoy, L., Gomes, B., Ashley, E. A., Duff, K. E., Buonassisi, T., Rainforth, T., Bateman, R. J., Skarlinski, M., Rodriques, S. G., Hinks, M. M., and White, A. D. Kosmos: An ai scientist for autonomous discovery, 2025b. URL https://arxiv.org/abs/2511.02824.

Nair, S., Gunsalus, L., Orcutt-Jahns, B., Rossen, J., Lal, A., Donno, C. D., Celik, M. H., Fletez-Brant, K., Xie, X., Bravo, H. C., and Eraslan, G. Agentic systems are adept at solving well-scoped, verifiable problems in computational biology. April 2026.

Panigrahi, S. S., Videnović, J., and Brbić, M. Heurekabench: A benchmarking framework for ai co-scientist, 2026. URL https://arxiv.org/abs/2601.01678.

Phylo. Evaluating AI agents in biology. https://phylo.bio/blog/evaluating-ai-agents-in-biology, February 2026. Accessed: 2026-5-8.

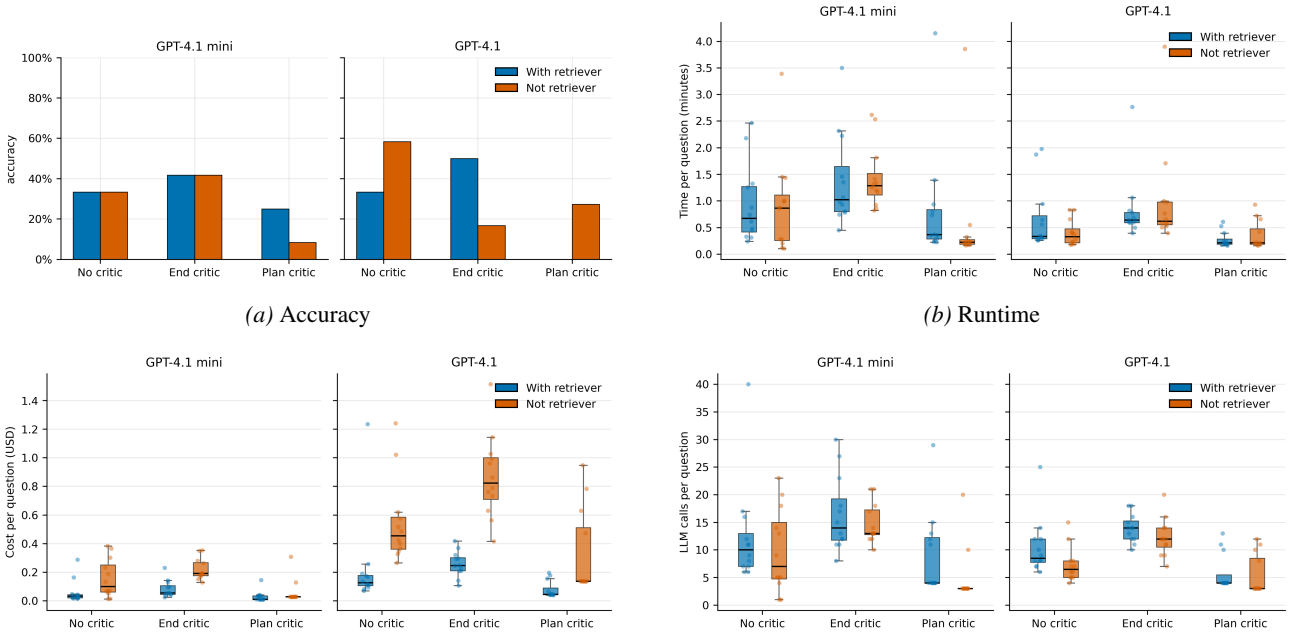
Qi, C., Wang, W., Jiang, S., Liu, Q., Song, X., Fang, H., and Wei, Z. Artificial intelligence agents for biological research: a survey. *Brief. Bioinform.*, 27(1), January 2026.

Sokolova, K., Kosenkov, D., Nallamotu, K., Vedula, S., Sokolov, D., Sapiro, G., and Troyanskaya, O. G. An evidence-grounded research assistant for functional genomics and drug target assessment. December 2025.

Wang, Z., Danek, B., and Sun, J. Biodsa-1k: Benchmarking data science agents for biomedical research, 2025. URL https://arxiv.org/abs/2505.16100.

Xu, B. Ai agent systems: Architectures, applications, and evaluation, 2026. URL https://arxiv.org/abs/2601.01743.

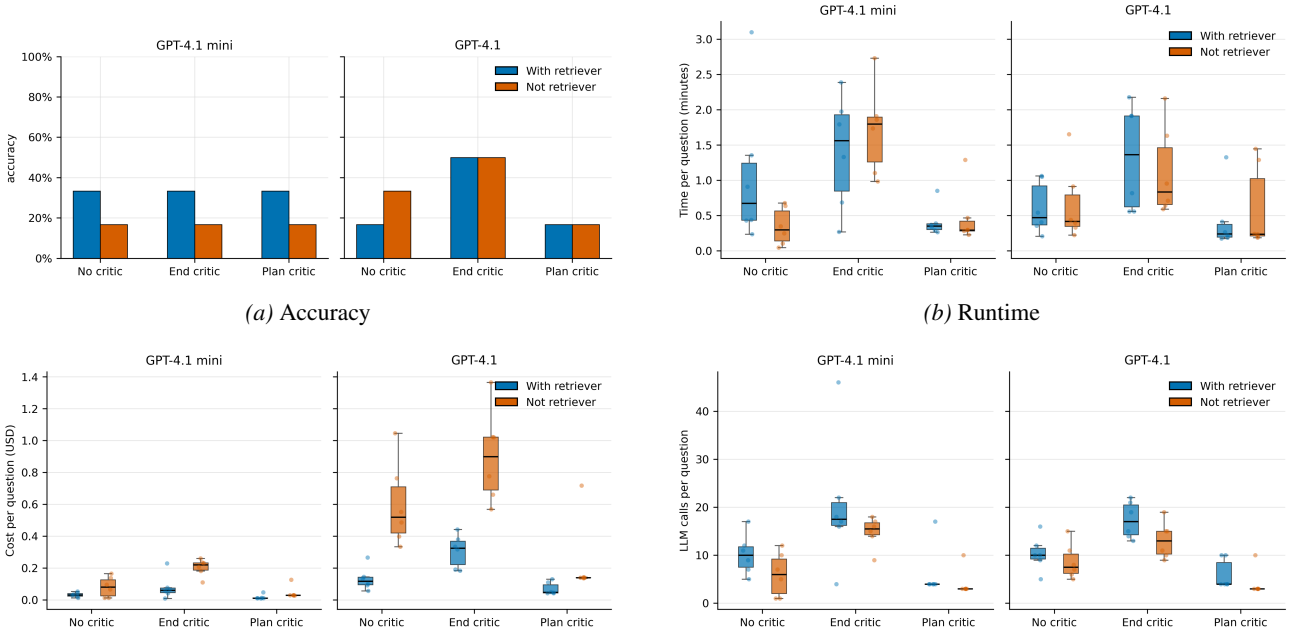
A. Verifier-Stratified Metrics



(a) Accuracy *(b)* Runtime

(c) Cost *(d)* LLM calls

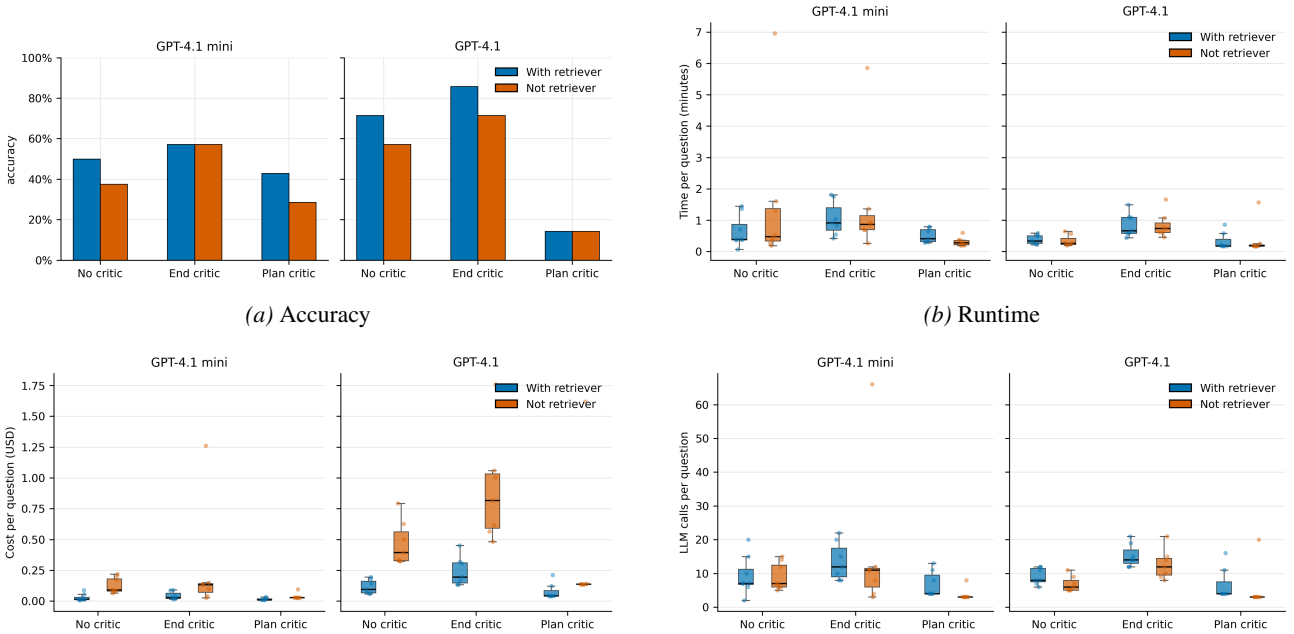
Figure 2. Accuracy, runtime, estimated cost, and LLM-call count for `llm_verifier` questions.



(a) Accuracy *(b)* Runtime

(c) Cost *(d)* LLM calls

Figure 3. Accuracy, runtime, estimated cost, and LLM-call count for `str_verifier` questions.



(a) Accuracy

(b) Runtime

(c) Cost

(d) LLM calls

Figure 4. Accuracy, runtime, estimated cost, and LLM-call count for `range_verifier` questions.