A Protocol-Driven Platform for Agent-Agnostic Evaluation of LLM Agents

Anonymous Author(s)
Affiliation
Address
email

Abstract

The fast growth of LLM-based agents has outstripped efforts to build solid evaluation standards, causing mismatched formats and idiosyncratic reporting that often make fair comparisons tricky. To tackle this, we created a platform around a protocol-based setup that keeps evaluations separate from how agents work inside, zeroing in on what we can actually see and measure. At the heart is our Remote Agent Integration Protocol (RAIP), a no-frills connection with just two endpoints: one for retrieve data schemas (/info) and another for running tasks (/invoke). Hook it up with TaskConfig—our handy layer for crafting clever input templates and grabbing outputs steadily via JMESPath—and switching agents turns effortless, no extra fiddling needed. Our benchmark structure ensures reproducibility by locking in versions. In addition, we consolidate prevalent metrics into 11 well-defined types within 4 categories for comprehensive evaluation. To check if it works, we ran a Course Advisor task with 30 examples, pitting a Teaching Assistant agent against three datasets—searching, details, and recommendation—while also comparing function-calling and workflow-graph architectures. The outcomes highlight RAIP's smooth swaps, no changes needed to data or scoring. In the end, this kind of standard could lead to evaluations that feel more even-handed, and expandable.

1 Introduction

The transition from static language models to interactive agentic systems, which incorporate planning, memory, and tool-use capabilities, marks a notable step forward in artificial intelligence. Yet assessing these agents often proves challenging. With agents handling extended interactions and grounding in external environments, the array of mismatched interfaces creates real hurdles, making fair comparisons harder to achieve. Different frameworks come with their own distinct systems, but when users—especially agent developers—want to test according to specific needs, it becomes tough to find a suitable evaluation setup. We see these issues as arising from a core protocol problem. While current frameworks provide clear benefits, they usually demand deep ties to specific ecosystems, like AgentBoard [22]—which is limited to particular environments—or SWE-bench [11], confined to software engineering domains. This not only limits cross-architecture studies but also makes accurate evaluations for new use cases difficult, requiring time-consuming specific setups that are hard to reproduce. Instead of enforcing uniformity on agent internals, our approach standardizes the boundary between the evaluation harness and the agent itself. This fosters an "observable black-box" methodology, where exchanges stay open and auditable without constraining design choices. Through our experiments, we uncover that a thin, standards-aligned boundary allows seamless agent swaps, without tweaking datasets or metrics.

Motivated by this observation, we designed a protocol-centric evaluation platform that decouples assessments from internal agent details, incorporating key elements such as:

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

- **Metric Standardization**: Categorizes metrics into 4 groups with 11 types, drawn from a survey of existing benchmarks to ensure comprehensive evaluation.
- **Protocol Specification (RAIP)**: Defines two core endpoints—`/info` for schema negotiation and `/invoke` for typed execution—featuring JSON Schema validation.
- **Binding Mechanism (TaskConfig)**: Maps reliable ways to transform inputs and outputs via JMESPath, balancing flexibility with rigor in bridging datasets to varied agent setups.
- **Proof of Feasibility**: Implements an end-to-end platform for a specific use case and agent, with a tailored benchmark and blinded human spot checks, confirming framework-agnostic interoperability.

This pipeline aligns with the LLM evaluation workshop's aims by providing an end-to-end platform that (i) standardizes protocol and metrics (11 types across 4 categories), (ii) streamlines benchmark design and dataset binding, and (iii) enables plug-in agent connectivity with auditable execution for reproducible, cross-framework evaluation.

2 Related Work

We start by drawing a clear line between benchmarks—encompassing tasks, environments, and metrics—and evaluations, which manage the procedural aspects and how results are aggregated. This separation guides our work, especially given the fragmented landscape of efforts in this area.

We propose that evaluations of agents fall into three broad categories: module-based, holistic, and domain-specific approaches. Module-based work zeroes in on individual elements like planning or tool integration, yielding focused observations, though it sometimes misses the broader interplay in complete systems. Holistic assessments, by contrast, gauge overall performance on varied challenges; AgentBench [21], for example, spans from coding to navigation, while AgentBoard [22] prioritizes sustained interactions. Domain-specific benchmarks target practical applications, such as SWE-bench [11] in software engineering, VisualWebArena [13] for web navigation, or SciAgent [23] in scientific workflows—yet their tailored designs often limit broader applicability. Tools like OpenAI Evals [24] and LangSmith [16] help compute metrics, and orchestration frameworks including AutoGen [36] and LangChain [14] streamline processes, even if they tend to lock users into particular platforms.

One recurring challenge emerges when benchmarks conflate tasks and datasets, which muddles equitable comparisons—take AgentBench's informal hierarchies or SWE-bench's reliance on unversioned GitHub issues. To tackle this, we lay out a more explicit framework: tasks define the instructions and setup, datasets group examples complete with validation mechanisms, and individual examples supply the inputs alongside ground-truth references, much as seen in GLUE [33] or BIG-bench [31].

Comparisons across systems also suffer from poor interoperability, as many frameworks require deep, platform-specific ties. Our protocol counters this limitation through black-box validation that accommodates varied implementations, from LangGraph [15] to ReAct [41].

Shifting focus to metrics, we group eleven varieties into four overarching classes. Performance and Execution address result effectiveness, trajectory quality, and action validity. Content and Fidelity involve checks on linguistic fidelity and substantive alignment. Attributes and Constraints capture efficiency, behavioral attributes, safety, and robustness—factors demanding careful trade-offs in practical deployment. Meta-metrics scrutinize evaluator reliability and benchmark integrity, helping maintain stable judgments over repeated trials. A consolidated taxonomy with representative examples is provided in Appendix A.2, Table 2. All told, these elements form the basis for our push toward evaluations that are both more equitable and adaptable.

3 Methodology

Our framework is designed to standardize the evaluation pipeline by establishing a clear, validated boundary between the Agent Under Test (AUT) and the evaluation logic. This boundary is formalized through the Remote Agent Integration Protocol (RAIP), which prioritizes simplicity, strong validation, and traceability.

RAIP structures the interaction into two distinct phases: discovery and execution, as illustrated in Appendix A, Figure 3. The process begins with the `/info` endpoint (GET), which publishes

agent metadata and the `inputSchema` compliant with JSON Schema Draft 2020-12 [35]. This enables upfront negotiation of the input contract and compatibility verification. Following discovery, execution proceeds via the `/invoke` endpoint (POST), accepting {`input`, `context`} and returning {`output`, `usage`, `raw?`}, with inputs strictly validated against the retrieved schema.

To ensure predictable and robust behavior, RAIP adheres to standard HTTP semantics, including 400 Bad Request for schema validation failures (with detailed `ajvErrors[]`), 422 Unprocessable Entity for semantic mismatches, and 429 for rate limiting. Operational policies further enhance reliability: pre-flight validation on a small sample ($k \leq 5$) rejects incompatible runs early, while privacy-first redaction hides raw provider payloads by default unless `trace=true` is specified. Additionally, ETag headers and schema digests (SHA-256, following JSON Canonicalization Scheme [27]) detect schema drift and confirm the evaluated agent version. Figure 1 provides a component-level overview of the architecture, highlighting the RAIP boundary.

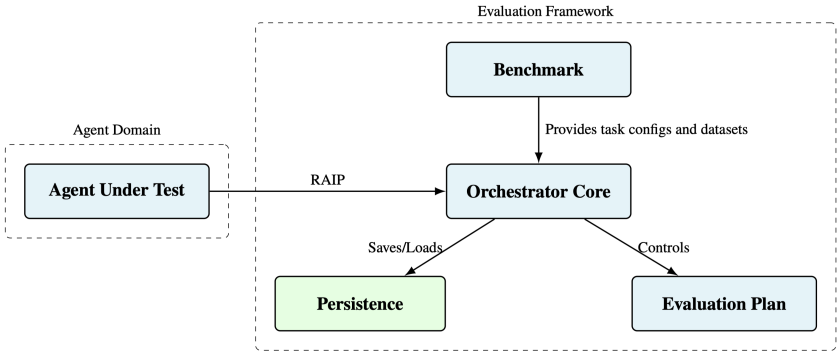

Figure 1: Component-based overview of the evaluation framework architecture.

Building on this standardized interface, TaskConfig serves as an agent-agnostic binding layer that bridges dataset examples and heterogeneous agents. It governs two critical transformations to decouple evaluation logic from agent internals. First, input templating maps example variables (e.g., `{{user_query}}`) to the agent's expected input shape, such as a messages array or function parameters, ensuring dataset independence from the agent interface. Second, deterministic output extraction normalizes varied JSON responses using JMESPath queries [28], yielding standardized candidate values for scoring. Figure 2 illustrates this two-stage pipeline.

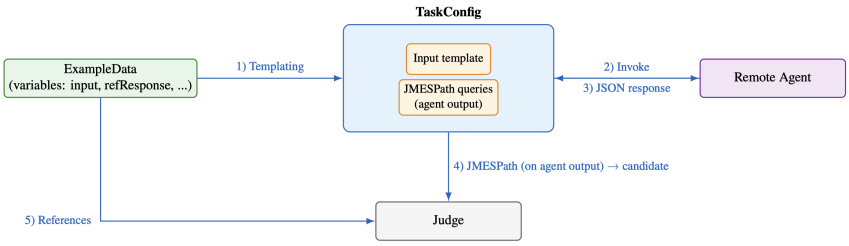

Figure 2: TaskConfig pipeline: declarative input templating and deterministic output extraction.

The framework further organizes data hierarchically (Benchmark→Task→Dataset→Example) to support granular diagnosis and consistent score aggregation via weighted means. Strict version-locking per run—for the agent, benchmark (including TaskConfigs and data), environment, and judge configuration—ensures reproducibility. Beyond version-locking, an extensible evaluation plan configuration allows teams to set up and expand metric families (e.g., add safety, constraint, or reliability checks) without modifying datasets or agent adapters, enabling incremental evolution of evaluation coverage.

4 Experimental Setup and Results

We conducted a feasibility study to validate the protocol-driven approach in enabling interoperable evaluation across distinct agent architectures. See Appendix A.3 (Fig. 4) for the end-to-end Course

Advisor analysis workflow visualization. The setup utilized the Course Advisor benchmark (N=30 examples across three datasets: Course Search, Details, and Career Path Recommendations) to compare two heterogeneous agents—A1 (Azure-based function-calling) and A2 (LangGraph-style workflow agent)—integrated solely via the RAIP interface. Controls ensured consistency, with identical benchmark versions, aggregation policies, and TaskConfig semantics, though input/output templates were adjusted to match each agent's schemas. An external, stronger judge model—distinct from the agents' generators—provided automated scoring.

The study demonstrated strong interoperability: The platform negotiated schemas via `/info` and executed tasks through `/invoke` for both agents, with pre-flight sanity checks confirming conformance before batch runs. Critically, swapping agents required no modifications to datasets or metric code, only TaskConfig template tweaks, affirming the agent-agnostic design. A system-level of agent execution snapshot with the finalized UI is provided in Appendix A.4 (Fig. 5).

Performance metrics served as feasibility signals at this pilot scale, with comparable outcome efficacy (A1: 53.5%, A2: 53.2%). This close alignment is expected because both agents target the same task objectives with harmonized prompt semantics and similar underlying model capabilities, leading naturally to convergent performance at this scale. The evaluation plan for this use case instantiated three metric families: (i) LLM-as-a-Judge (Content & Fidelity), (ii) Time to Success (Attributes & Constraints), and (iii) Pass Rate (Performance & Execution). Per-dataset judge scores and efficiency indicators appear in Table 1, where A2 showed slightly better response structure and lower average time to success (5.35s vs. 5.73s), reflecting architectural trade-offs.

	Judge Score (%)				Efficiency (Overall)		
Agent	Search	Details	Path Recommandations	Overall	Time to Success (s)	Pass Rate	Error
A1 (Azure)	58.3	38.1	62.1	**53.5**	5.73	100%	0%
A2 (LangGraph)	57.0	37.5	60.7	**53.2**	5.35	100%	0%

Table 1: Course Advisor: scores and overall efficiency.

As meta-metrics, blinded human spot-checks (n=30) revealed moderate agreement with the automated judge (Pearson r=0.43), reported separately to confirm instrumentation reliability and highlight areas for human-in-the-loop refinement.

5 Discussion

Prioritizing a protocol-first perspective is essential because, while prior work has advanced novel metrics and benchmarks, integration friction remains a significant barrier to comparative studies. By treating evaluation as a protocol problem, RAIP provides a thin, standards-aligned boundary that reduces brittle glue code, enforces early validation, and enhances traceability across heterogeneous systems. Nevertheless, this feasibility study is constrained by its small scale (n=30), single domain, and limited number of agents, and potential judge variance remains a threat to validity, even though partially mitigated by controls. Furthermore, security features (e.g., authentication) were design-only in this proof of concept, so we present these results as evidence of feasibility rather than definitive performance leaderboards.

6 Conclusion

We demonstrated that a protocol-level boundary (RAIP + TaskConfig) enables agent-agnostic interoperability: heterogeneous agents can be swapped without modifying task, dataset, or metric logic. The explicit task→dataset→example hierarchy together with version-locking of artifacts and configurations strengthens reproducibility and longitudinal comparability. Our feasibility study (n=30, single advising domain, limited agent set) provides evidence of practicality rather than performance ranking. Despite scope constraints, early validation hooks and standardized meta-metrics improved traceability and fair aggregation. This protocol-first stance preserves internal innovation while normalizing evaluation surfaces. Future work will extend to multi-domain scenarios and introduce robustness, safety, and judge reliability probes. We invite community collaboration to refine the protocol surface and expand open meta-metric suites.

References

[1] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[2] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

[3] Laura Dietz, Oleg Zendel, Peter Bailey, Charles L. A. Clarke, Ellese Cotterill, Jeff Dalton, Faegheh Hasibi, Mark Sanderson, and Nick Craswell. Principles and guidelines for the use of llm judges. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, ICTIR '25, page 218–229. ACM, July 2025.

[4] Luca Gioacchini, Giuseppe Siracusano, Davide Sanvito, Kiril Gashteovski, David Friede, Roberto Bifulco, and Carolin Lawrence. AgentQuest: A modular benchmark framework to measure progress and improve LLM agents. In Kai-Wei Chang, Annie Lee, and Nazneen Rajani, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 185–193, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[5] Hitesh Goel and Hao Zhu. Lifelongsotopia: Evaluating social intelligence of language agents over lifelong social interactions, 2025.

[6] Dewi S. W. Gould, Bruno Mlodozeniec, and Samuel F. Brown. Skate, a scalable tournament eval: Weaker llms differentiate between stronger ones using verifiable challenges, 2025.

[7] Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware LLM reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855, Vienna, Austria, July 2025. Association for Computational Linguistics.

[8] Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5880–5895, Vienna, Austria, July 2025. Association for Computational Linguistics.

[9] Takashi Ishida, Thanawat Lodkaew, and Ikko Yamane. How can i publish my llm benchmark without giving the true answers away?, 2025.

[10] Alon Jacovi, Andrew Wang, Chris Alberti, Connie Tao, Jon Lipovetz, Kate Olszewska, Lukas Haas, Michelle Liu, Nate Keating, Adam Bloniarz, Carl Saroufim, Corey Fry, Dror Marcus, Doron Kukliansky, Gaurav Singh Tomar, James Swirhun, Jinwei Xing, Lily Wang, Madhu Gurumurthy, Michael Aaron, Moran Ambar, Rachana Fellinger, Rui Wang, Zizhao Zhang, Sasha Goldshtein, and Dipanjan Das. The facts grounding leaderboard: Benchmarking llms' ability to ground responses to long-form input, 2025.

[11] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024.

[12] Su Kara, Fazle Faisal, and Suman Nath. Waber: Evaluating reliability and efficiency of web agents with existing benchmarks. In *ICLR 2025 Workshop on Foundation Models in the Wild*, April 2025.

[13] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[14] LangChain Inc. Langchain framework. Documentation, 2023.

[15] LangChain Inc. Langgraph: State-graph framework for llm applications. Product documentation, 2024. Accessed: 2025-09-01.

[16] LangChain Inc. *LangSmith: Evaluation and Tracing Platform*, 2025. Accessed: 2025-09-01.

[17] Ido Levy, Ben wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. ST-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents. In *ICML 2025 Workshop on Computer Use Agents*, 2025.

[18] Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild, 2024.

[19] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[20] Teng Lin. Mebench: Benchmarking large language models for cross-document multi-entity question answering, 2025.

[21] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. In *ICLR*, 2024.

[22] Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn LLM agents. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[23] Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, Aixin Sun, Hany Awadalla, and Weizhu Chen. Sciagent: Tool-augmented language models for scientific reasoning, 2024.

[24] OpenAI. Openai evals. GitHub repository, 2023. Evaluation framework.

[25] Yichen Pan, Dehan Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, and Zhengyang Wu. Webcanvas: Benchmarking web agents in online environments. In *Agentic Markets Workshop at ICML 2024*, 2024.

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[27] Anders Rundgren, Bret Jordan, and Samuel Erdtman. Json canonicalization scheme (jcs). Technical Report 8785, RFC Editor, May 2020. Defines a canonical representation for JSON data to support cryptographic operations like hashing and signing. Accessed: September 2025.

[28] James Saryerwinnie. Jmespath specification. Online Specification, 2013. Formal grammar and specification for JMESPath, a query language for JSON. Maintained by the JMESPath community; latest updates as of 2025. Accessed: September 2025.

[29] Lin Shi, Chiyu Ma, Wenhua Liang, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by LLMs, 2024.

[30] Lucas Spangher, Tianle Li, William F. Arnold, Nick Masiewicki, Xerxes Dotiwalla, Rama Kumar Pasumarthi, Peter Grabowski, Eugene Ie, and Daniel Gruhl. Chatbot arena estimate: towards a generalized performance benchmark for LLM capabilities. In Weizhu Chen, Yi Yang, Mohammad Kachuee, and Xue-Yong Fu, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 1016–1025, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[31] Aarohi Srivastava et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2023.

[32] {Aayush Atul} Verma, Amir Saeidi, Shamanthak Hegde, Ajay Therala, {Fenil Denish} Bardoliya, Nagaraju MacHavarapu, {Shri Ajay Kumar} Ravindhiran, Srija Malyala, Agneet Chatterjee, Yezhou Yang, and Chitta Baral. Evaluating multimodal large language models across distribution shifts and augmentations. In *Proceedings - 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2024*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pages 5314–5324. IEEE Computer Society, 2024. Publisher Copyright: © 2024 IEEE.; 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2024 ; Conference date: 16-06-2024 Through 22-06-2024.

[33] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi, editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[34] Zilong Wang, Jingfeng Yang, Sreyashi Nag, Samarth Varshney, Xianfeng Tang, Haoming Jiang, Jingbo Shang, and Sheikh Muhammad Sarwar. RRO: LLM agent optimization through rising reward trajectories. In *Second Conference on Language Modeling*, 2025.

[35] Austin Wright, Henry Andrews, Ben Hutton, and Greg Dennis. Json schema: A media type for describing json documents. Internet-Draft draft-bhutton-json-schema-01, Work in Progress, June 2022. Draft 2020-12 specification for JSON Schema, defining structure and validation for JSON data. Accessed: September 2025.

[36] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

[37] Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, Ruoxi Jia, Bo Li, Kai Li, Danqi Chen, Peter Henderson, and Prateek Mittal. SORRY-bench: Systematically evaluating large language model safety refusal. In *The Thirteenth International Conference on Learning Representations*, 2025.

[38] Tianqi Xu, Linyao Chen, Dai-Jie Wu, Yanjun Chen, Zecheng Zhang, Xiang Yao, Zhiqiang Xie, Yongchao Chen, Shilong Liu, Bochen Qian, Anjie Yang, Zhaoxuan Jin, Jianbo Deng, Philip Torr, Bernard Ghanem, and Guohao Li. Crab: Cross-environment agent benchmark for multimodal language model agents, 2025.

[39] Yusuke Yamauchi, Taro Yano, and Masafumi Oyamada. An empirical study of llm-as-a-judge: How design choices impact evaluation reliability, 2025.

[40] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024.

[41] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.

[42] Daoguang Zan, Zhirong Huang, Ailun Yu, Shaoxin Lin, Yifan Shi, Wei Liu, Dong Chen, Zongshuai Qi, Hao Yu, Lei Yu, Dezhi Ran, Muhan Zeng, Bo Shen, Pan Bian, Guangtai Liang, Bei Guan, Pengjie Huang, Tao Xie, Yongji Wang, and Qianxiang Wang. Swe-bench-java: A github issue resolving benchmark for java, 2024.

[43] Yi Zhan, Longjie Cui, Han Weng, Guifeng Wang, Yu Tian, Boyi Liu, Yingxiang Yang, Xiaoming Yin, Jiajun Xie, and Yang Sun. Towards database-free text-to-SQL evaluation: A graph-based metric for functional correctness. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4586–4610, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[44] Yuxiang Zhang, Jing Chen, Junjie Wang, Yaxin Liu, Cheng Yang, Chufan Shi, Xinyu Zhu, Zihao Lin, Hanwen Wan, Yujiu Yang, Tetsuya Sakai, Tian Feng, and Hayato Yamana. Toolbehonest: A multi-level hallucination diagnostic benchmark for tool-augmented large language models, 2024.

[45] Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. SafetyBench: Evaluating the safety of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15537–15553, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[46] Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Robert Tang, Heng Ji, and Jiaxuan You. MultiAgentBench : Evaluating the collaboration and competition of LLM agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8580–8622, Vienna, Austria, July 2025. Association for Computational Linguistics.

A Appendix

This appendix augments the main experiments with: (i) a protocol sequence trace (RAIP discovery and execution), (ii) a normalized metrics taxonomy spanning performance, content fidelity, attributes, and meta-evaluation, and (iii) the end-to-end Course Advisor analysis workflow (Fig. 4) illustrating schema negotiation, heterogeneous agent interchangeability, automated judging, and blinded human validation.

A.1 RAIP Sequence Diagram

The RAIP sequence clarifies discovery (capabilities + schema digest) and validated execution with a uniform error surface.

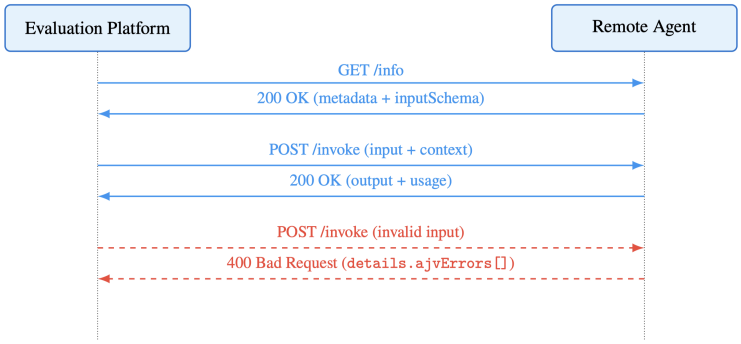

Figure 3: RAIP sequence: discovery via /info (schema negotiation + digest) followed by validated execution via /invoke with a standardized error model.

A.2 Metrics taxonomy

Metric family	Representative examples
I. Performance & Execution	
Outcome Efficacy	Success / Pass / Resolve rate [42]; Win rate [2]; TrueSkill rating [6].
Trajectory Quality	Average return / reward [34]; Progress rate [4]; Checkpoint completion (key-nodes) [25].
Action Validity	Grounding accuracy [10]; Invalid action rate [38]; Repetition / hallucination (tool actions) [44].
II. Content & Fidelity	
Linguistic Fidelity	ROUGE [19]; BLEU [26]; F1 (entity-attributed) [20]; Exact Match [43].
Content Assessment	LLM-as-a-Judge scores (WB-Score/WB-Reward) [18]; Checklist accuracy [25]; Human preference [2].
III. Attributes & Constraints	
Efficiency	Time to success [12]; Average steps []; Token/$ cost efficiency [7].
Behavioral Attributes	Cooperation / competition [46]; Compliance [17]; Confidence / consistency [39]; Social goal completion [5].
Safety	Safety score [45]; Policy adherence [17]; Rejection rate [37].
Robustness	Jailbreak / attack success [1]; Distribution shift [32]; pass@k (tool/API) [40].
IV. Meta-metrics	
Evaluator Reliability	Human alignment [8]; Judge stability / position bias [29]; Expert correlation [30].
Benchmark Integrity	Suitableness / validity [3]; Ground-truth protection [9].

Table 2: Orthogonal metric families for LLM-agent evaluation.

A.3 Course Advisor Analysis Workflow

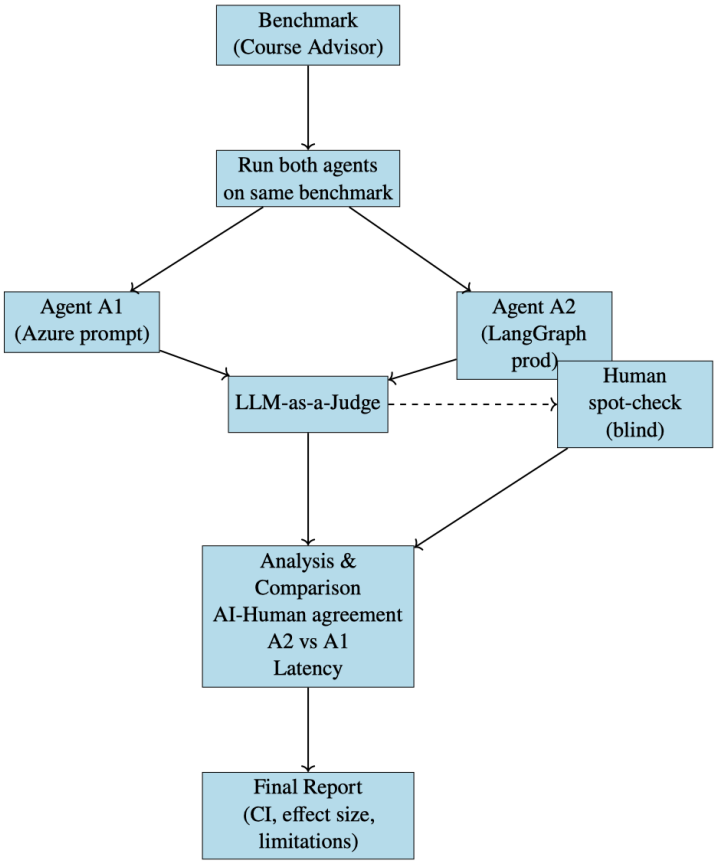

Figure 4: Course Advisor analysis workflow: single protocol, heterogeneous agents, automated judging, blinded human validation, and aggregated reporting.

A.4 System Run Overview

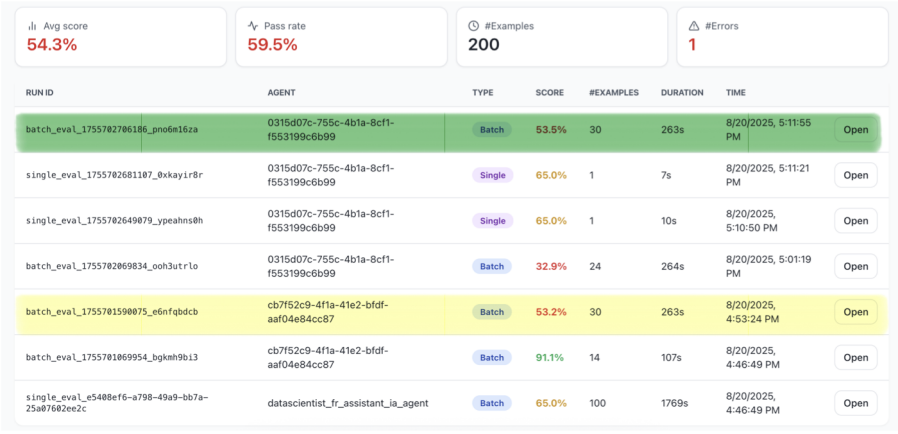

Figure 5: End-to-end dual-agent execution in the production interface: negotiated schemas, synchronized task dispatch, streaming responses, judge scoring, and aggregated reporting.