Evaluating Behavioral Failures in Coding Agents

Anonymous Authors
Affiliation Withheld for Review
anonymous@anonymous.com

Abstract

Coding agents are becoming increasingly autonomous, capable of generating, modifying, and executing code across multi-turn human–AI workflows. However, current evaluation pipelines still primarily rely on aggregate correctness metrics, preference-based ranking, repair success, or benchmark-driven evaluations, providing limited insight into behavioral failures that emerge across multi-turn coding interactions. In this work, we introduce RHCA, a human-centered framework for behavioral analysis of coding agents. RHCA organizes interaction analysis across four dimensions: Reasoning Transparency, Helpfulness, Consistency, and Context Alignment. We further introduce a structured failure taxonomy, RHCA-guided failure tags, and reusable evaluation cards for analyzing behavioral failures in multi-turn coding interactions. Overall, this work contributes a behaviorally granular and human-centered framework for analyzing coding-agent failures beyond aggregate evaluation.

1 Introduction

Coding agents are increasingly integrated into real-world software development workflows. Despite rapid progress in coding-agent capabilities, current evaluation pipelines continue to rely on aggregate-centric evaluation and outcome-level metrics. While these approaches support scalable evaluation, they provide limited insight into behavioral failures that emerge across multi-turn human–AI coding interactions (Figure 1). This paper investigates the following research question: *How can we systematically analyze behavioral failures in coding agents beyond aggregate evaluation metrics?* Overall, this work contributes RHCA, a human-centered framework for structured behavioral analysis of coding agents, together with RHCA-guided failure tags and reusable evaluation cards supported by empirical illustrations of behavioral failures in coding workflows.

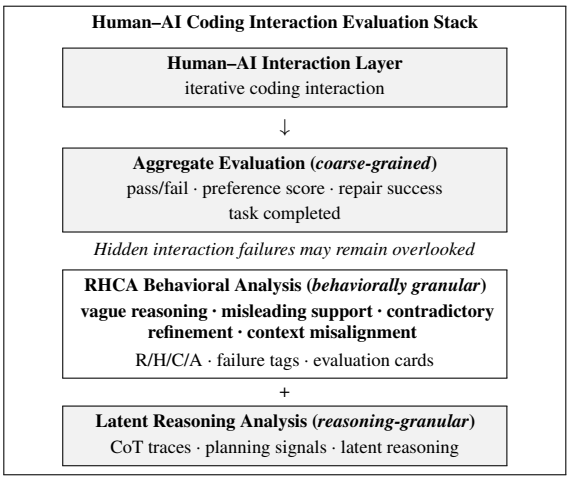

Figure 1: Aggregate evaluation may overlook observable failures during multi-turn human–AI coding interactions. RHCA complements latent reasoning analysis through structured behavioral evaluation.

2 Related Work

2.1 Aggregate-Centric Evaluation

Current LLM evaluation pipelines primarily rely on aggregate metrics, benchmark-based comparison, and preference-driven evaluation (Hutchinson et al., 2022; Burnell et al., 2023). Benchmarks such as GLUE (Wang et al., 2018), HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and SWE-Bench (Saha et al., 2024) evaluate task correctness, or repair success through coarse-grained metrics such as accuracy, pass/fail rates, and preference ranking. Automated metrics including BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and BERTScore (Zhang et al., 2019) further support scalable proxy-based evaluation for text generation. More recent evaluation practices incorporate preference modeling and model-based judgments, including LLM-as-a-judge frameworks such as MT-Bench (Zheng et al., 2023) and Arena-Hard-Auto (Li et al., 2024b). Existing studies suggest that preference-based optimization and automated evaluation may encourage undesirable behaviors

or systematic biases, including unstable preference judgments (Sharma et al., 2024; Li et al., 2025; Szymanski et al., 2025). Together, these limitations motivate evaluation approaches beyond proxy-centric optimization toward more interpretable and human-centered assessment of system behavior (Musgrave et al., 2020).

Recent agent-evaluation research further argues that single-metric evaluation remains insufficient for understanding agent behavior, motivating dimensions such as consistency, robustness, and safety beyond aggregate task success (Rabanser et al., 2026). Related work explores scalable failure discovery, performance estimation, runtime debugging, and executable recovery for coding agents and agentic systems (Huang et al., 2026b; Rahardja et al., 2026; Gu et al., 2026; Huang et al., 2026a). While these approaches improve repair capability and automated debugging, they still rely on coarse-grained outcome metrics, repair success, or performance optimization (Laban et al., 2025).

Overall, existing approaches emphasize aggregate correctness, preference ranking, repair success, or system reliability, while providing limited support for structured behavioral diagnosis across multi-turn human–AI coding interactions.

2.2 Behavioral Failure Analysis and Human-Centered Evaluation

Prior work increasingly argues that evaluation of generative AI systems should account for contextual adaptation, human behavior, and deployment settings beyond static correctness-based metrics (Selbst et al., 2019; Amershi et al., 2019; Slade et al., 2024). Most existing benchmarks remain static or single-turn despite real-world human–AI interaction being inherently dynamic and multi-turn (Ibrahim et al., 2025; Ji et al., 2023). Recent work has begun to explore conversational consistency and long-horizon interaction evaluation (Abdulhai et al., 2026; Su et al., 2026). However, many approaches still assess interaction quality through aggregate preferences or coarse outcome-level signals rather than structured behavioral analysis across interaction trajectories (Burnell et al., 2023).

Prior work on AI coding assistants studies developer workflows through taxonomy-based analysis of interaction activities (Mozannar et al., 2024; Chen et al., 2026; Cemri et al., 2026; Lynch et al., 2025). These studies reveal challenges in user understanding, interaction inefficiencies, and agent adoption, but primarily focus on productivity or system-level reliability rather than structured analysis of contextual interaction failures during coding collaboration.

Recent work on AI monitorability further argues that final outputs alone remain insufficient for evaluating increasingly capable AI systems, motivating process-level evaluation through observable monitoring signals (Guan et al., 2025; OpenAI, 2026). Existing work also explores grading workflows, evaluation infrastructure, annotation reliability (Anthropic, 2026), and broader behavioral values in deployed AI systems (Huang et al., 2025). However, these approaches primarily focus on latent reasoning visibility, large-scale monitoring, or descriptive behavioral analysis rather than structured analysis of multi-turn human–AI coding interactions. RHCA complements these approaches through behaviorally granular analysis of observable interaction failures in human-facing coding systems.

Collectively, existing approaches provide limited support for structured analysis of behavioral failures such as vague reasoning, misleading support, contradictory refinement, and context misalignment across multi-turn human–AI coding interactions. Our work instead introduces a behaviorally granular and human-centered framework through RHCA dimensions, structured failure taxonomy, and reusable evaluation cards for analyzing coding-agent failures beyond aggregate task success.

3 RHCA Dimensions

RHCA provides a framework for analyzing interaction quality across iterative human–AI interaction. The framework is informed by a scoping review of 126 sources (Arksey and O'malley, 2005) across AI evaluation, human-centered AI, and HCI, spanning conference papers, journal articles, and prior literature from pre-LLM systems to recent generative AI research. We further incorporated authentic interaction logs from educational coding-support settings to identify interaction patterns in human-facing LLM systems. Through iterative analysis, four dimensions emerged: Reasoning Transparency (R), Helpfulness (H), Consistency (C), and Context Alignment (A). Together, these dimensions support structured analysis of interaction failures in coding agents.

Table 1 summarizes the RHCA dimensions and

representative interaction indicators.

Reasoning Transparency (R) assesses whether system explanations remain interpretable and traceable for users (Nauta et al., 2023; Arrieta et al., 2020; Guidotti and Ruggieri, 2019; Zhao et al., 2026).

Helpfulness (H) measures whether responses provide actionable and pedagogically appropriate guidance that supports user progress (Kapur, 2008; Amershi et al., 2019; Gligorea et al., 2023; Faruk et al., 2025).

Consistency (C) captures whether behavior remains coherent across turns without contradiction or instability (Kirmayr et al., 2026; Li et al., 2024a; Zhao et al., 2024; Nauta et al., 2023).

Context Alignment (A) reflects whether responses remain relevant to user intent and evolving interaction context (Prasad et al., 2025; Amershi et al., 2019; Wang et al., 2024).

Dimension	Indicators
Reasoning Transparency (R)	explanation clarity reasoning steps
Helpfulness (H)	actionable guidance support for progress
Consistency (C)	coherence across turns stable behavior
Context Alignment (A)	goal relevance context awareness

Table 1: RHCA dimensions and representative interaction indicators.

4 Methodology

4.1 Data Source and Analysis

We refine RHCA through real-world interaction data. We study coding agents in expressive programming environments involving debugging, refinement, and user-driven creative interaction. Using an educational music-coding environment as an illustrative setting, we analyzed over 1,400 interaction logs from an IRB-approved classroom study involving 81 students. Unlike conventional coding benchmarks that primarily emphasize task correctness, expressive music-coding interactions combine programming tasks with open-ended creative goals and novice-facing learning support, requiring iterative clarification, contextual adaptation, and multi-turn refinement during coding interactions. These characteristics provide richer interaction signals for analyzing behavioral failures beyond aggregate evaluation. Despite growing interest in coding agents, expressive and novice-facing programming interactions remain relatively underexplored in current evaluation research. From these consented logs, we identified common interaction patterns including clarification, debugging, and refinement, and constructed 13 multi-turn scenarios (2–4 turns) to capture representative coding-agent behaviors for structured behavioral analysis beyond aggregate evaluation. Scenario selection and refinement were further reviewed during annotation calibration and expert discussion.

4.2 Annotation Protocol

We evaluated model responses across four RHCA dimensions: Reasoning Transparency (R), Helpfulness (H), Consistency (C), and Context Alignment (A). Each dimension was scored on a 3-point scale (1 = Poor, 2 = Partial, 3 = Good). Detailed scoring rubrics are provided in Appendix A. Failure tags were primarily assigned when interactions received a score of 1 in a given RHCA dimension, indicating clear behavioral failures. For partially successful interactions (score = 2), tags were optionally assigned when notable behavioral limitations were observed, while score-3 interactions typically did not receive failure tags.

Reasoning Transparency evaluates explanation clarity for novice users. Helpfulness measures whether responses provide actionable support. Consistency captures whether behavior remains coherent across turns. Context Alignment reflects whether responses adapt appropriately to user intent and interaction context.

Annotators participated in calibration involving independent scoring, disagreement discussion, and rubric refinement. Responses were anonymized and randomized prior to annotation. Inter-rater agreement was measured using Fleiss' κ across three annotators on RHCA dimension ratings for the 13 multi-turn evaluation scenarios.

5 RHCA Behavioral Analysis

5.1 Behavioral Failures

We introduce failure tags to support structured diagnosis of behavioral failures in coding agents. Table 2 summarizes the interaction failure tags organized under RHCA dimensions.

The following example illustrates how behavioral failures may remain hidden under aggregate evaluation. Although the model responses appear correct at the outcome level, the interaction pro-

D	Failure Tag	Description
R	vague reasoning	Explanation lacks clarity for novice users.
R	missing reasoning	Skips important reasoning steps.
H	wrong diagnosis	Identifies the incorrect issue.
H	no clear steps	Lacks actionable guidance.
H	misleading support	Appears helpful but leads users incorrectly.
C	inconsistency	Contradicts prior responses.
C	contradictory refinement	Refinement changes direction across turns.
A	ignores context	Fails to use prior interaction context.
A	misreads intent	Misunderstands user goals or requests.
A	overly generic	Response remains insufficiently specific.

Table 2: Interaction failure tags organized under RHCA dimensions.

vides limited reasoning transparency and weak support for novice users. Many responses judged successful under aggregate evaluation still exhibited behavioral failures, including misleading support (23.5%, 8/34), suggesting that correctness alone is insufficient to characterize coding-agent behavior. Common failures also included vague reasoning, contradictory refinement, and context-related failures. This example illustrates how aggregate evaluation may overlook important behavioral limitations during multi-turn human–AI coding interactions.

5.2 RHCA Evaluation Card

To support interpretable behavioral analysis, we further organize RHCA observations through reusable evaluation cards. Table 3 presents an illustrative RHCA evaluation card adapted from a coding-support interaction, showing behavioral failures across iterative human–AI collaboration.

We use structured human annotation to analyze behaviorally granular interaction failures in coding agents. Agreement across annotators was substantial ($\kappa = 0.620$–0.685, mean $\kappa = 0.653$), indicating that RHCA dimensions were interpretable and consistently applied across evaluation scenarios.

Overall, RHCA provides reusable evaluation artifacts for structured diagnosis of behavioral failures across iterative human–AI coding interactions.

T	Interaction	RHCA Observation
1	**User:** How do I create a beat? **Model:** Explains beat strings and provides a simple `makeBeat()` example.	Clear explanation with actionable guidance (R3, H3).
2	**User:** How can I make my loop repeat without manually adding more sounds? **Model:** Suggests increasing the loop range but does not explain continuous repetition.	Partially helpful response with incomplete support (H2).
3	**User:** I want the beat to repeat continuously. **Model:** Repeats earlier suggestions and introduces unrelated explanations about sound lists.	Vague reasoning and context misalignment (R1, A1).
Failure Tags: vague reasoning (R1); no clear steps (H2); contradictory refinement (C2); context misalignment (A1)		

Table 3: Illustrative RHCA evaluation card showing behavioral failures across multi-turn coding interactions

6 Discussion

Our findings indicate that aggregate evaluation alone is insufficient for understanding failures in multi-turn human–AI coding interactions. Model outputs may appear correct under standard metrics while still exhibiting behavioral failures, such as vague reasoning or context-related failures. RHCA complements traditional evaluation approaches by analyzing interaction failures beyond task correctness. It enables structured portable evaluation across diverse research and deployment contexts, including smaller-scale studies and pre-deployment settings. The framework's structured dimensions, failure tags, and reusable evaluation artifacts also support scalable evaluation through hybrid human–AI evaluation workflows. More broadly, our work suggests that evaluating human-facing AI systems requires structured behavioral analysis alongside aggregate evaluation, and that RHCA may generalize to other human-facing agent systems.

7 Limitations and Future Work

This work presents an initial step toward structured human-centered evaluation of coding-agent behavior. At this stage, RHCA focuses on observable interaction behaviors rather than latent reasoning processes or internal model states. Future work may further investigate connections between observable interaction failures and latent reasoning dynamics. Additional directions include extending RHCA across broader domains and supporting scalable behavioral evaluation.

References

Marwa Abdulhai, Ryan Cheng, Donovan Clay, Tim Althoff, Sergey Levine, and Natasha Jaques. 2026. Consistently simulating human personas with multi-turn reinforcement learning. *Advances in Neural Information Processing Systems*, 38:52920–52957.

Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, and 1 others. 2019. Guidelines for human-ai interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–13.

Anthropic. 2026. Demystifying evals for ai agents. Engineering report, Anthropic.

Hilary Arksey and Lisa O'malley. 2005. Scoping studies: towards a methodological framework. *International journal of social research methodology*, 8(1):19–32.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, and 1 others. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Ryan Burnell, Wout Schellaert, John Burden, Tomer D Ullman, Fernando Martinez-Plumed, Joshua B Tenenbaum, Danaja Rutar, Lucy G Cheke, Jascha Sohl-Dickstein, Melanie Mitchell, and 1 others. 2023. Rethink reporting of evaluation results in ai. *Science*, 380(6641):136–138.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2026. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Valerie Chen, Ameet Talwalkar, Robert Brennan, and Graham Neubig. 2026. Code with me or for me? how increasing ai automation transforms developer workflows. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–19.

Lawal Ibrahim Dutsinma Faruk, Debajyoti Pal, Suree Funilkul, Thinagaran Perumal, and Pornchai Mongkolnam. 2025. Introducing casux: A standardized scale for measuring the user experience of artificial intelligence based conversational agents. *International Journal of Human–Computer Interaction*, 41(9):5274–5298.

Ilie Gligorea, Marius Cioca, Romana Oancea, Andra-Teodora Gorski, Hortensia Gorski, and Paul Tudorache. 2023. Adaptive learning using artificial intelligence in e-learning: A literature review. *Education Sciences*, 13(12):1216.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. 2026. Safe: Multitask failure detection for vision-language-action models. *Advances in Neural Information Processing Systems*, 38:40041–40076.

Shengyue Guan, Jindong Wang, Jiang Bian, Bin Zhu, Jian-Guang Lou, and Haoyi Xiong. 2025. Evaluating llm-based agents for multi-turn conversations: A survey. *ACM Transactions on Intelligent Systems and Technology*.

Riccardo Guidotti and Salvatore Ruggieri. 2019. On the stability of interpretable models. In *2019 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Jiangping Huang, Wenguang Ye, Weisong Sun, Jian Zhang, Mingyue Zhang, and Yang Liu. 2026a. Tracecoder: A trace-driven multi-agent framework for automated debugging of llm-generated code. *arXiv preprint arXiv:2602.06875*.

Saffron Huang, Esin Durmus, Miles McCain, Kunal Handa, Alex Tamkin, Jerry Hong, Michael Stern, Arushi Somani, Xiuruo Zhang, and Deep Ganguli. 2025. Values in the wild: Discovering and analyzing values in real-world language model interactions. *arXiv preprint arXiv:2504.15236*.

Yizheng Huang, Wenjun Zeng, Aditi Kumaresan, and Zi Wang. 2026b. Proeval: Proactive failure discovery and efficient performance estimation for generative ai evaluation. *arXiv preprint arXiv:2604.23099*.

Ben Hutchinson, Negar Rostamzadeh, Christina Greer, Katherine Heller, and Vinodkumar Prabhakaran. 2022. Evaluation gaps in machine learning practice. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 1859–1876.

Lujain Ibrahim, Saffron Huang, Lama Ahmad, Umang Bhatt, and Markus Anderljung. 2025. Towards interactive evaluations for interaction harms in human-ai systems. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 8, pages 1302–1310.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, and 1 others. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

Manu Kapur. 2008. Productive failure. *Cognition and instruction*, 26(3):379–424.

Johannes Kirmayr, Lukas Stappen, and Elisabeth André. 2026. Car-bench: Evaluating the consistency and limit-awareness of llm agents under real-world uncertainty. *arXiv preprint arXiv:2601.22027*.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, and 1 others. 2025. From generation to judgment: Opportunities and challenges of llm-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024a. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024b. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Aengus Lynch, Benjamin Wright, Caleb Larson, Stuart J Ritchie, Soren Mindermann, Evan Hubinger, Ethan Perez, and Kevin Troy. 2025. Agentic misalignment: How llms could be insider threats. *arXiv preprint arXiv:2510.05179*.

Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. 2024. Reading between the lines: Modeling user behavior and costs in ai-assisted programming. In *Proceedings of the 2024 CHI conference on human factors in computing systems*, pages 1–16.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 2020. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. 2023. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys*, 55(13s):1–42.

OpenAI. 2026. How we monitor internal coding agents for misalignment. Research report, OpenAI.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Prajish Prasad, Rishabh Balse, and Dhwani Balchandani. 2025. Exploring multimodal generative ai for education through co-design workshops with students. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–17.

Stephan Rabanser, Sayash Kapoor, Peter Kirgis, Kangheng Liu, Saiteja Utpala, and Arvind Narayanan. 2026. Towards a science of ai agent reliability. *arXiv preprint arXiv:2602.16666*.

Alfin Wijaya Rahardja, Junwei Liu, Weitong Chen, Zhenpeng Chen, and Yiling Lou. 2026. Can agent fix agent issues? *Advances in Neural Information Processing Systems*, 38:49002–49028.

Anik Kumar Saha, Md Abrar Jahin, Md Rafiquzzaman, and Muhammad Firoz Mridha. 2024. Ergonomic design of computer laboratory furniture: Mismatch analysis utilizing anthropometric data of university students. *Heliyon*, 10(14).

Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and abstraction in sociotechnical systems. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 59–68.

Mrinank Sharma, Meg Tong, Tomek Korbak, David Duvenaud, Amanda Askell, Sam Bowman, Esin Durmus, Zac Hatfield-Dodds, Scott Johnston, Shauna Kravec, and 1 others. 2024. Towards understanding sycophancy in language models. In *International Conference on Learning Representations*, volume 2024, pages 110–144.

Patrick Slade, Christopher Atkeson, J Maxwell Donelan, Han Houdijk, Kimberly A Ingraham, Myunghee Kim, Kyoungchul Kong, Katherine L Poggensee, Robert Riener, Martin Steinert, and 1 others. 2024. On human-in-the-loop optimization of human–robot interaction. *Nature*, 633(8031):779–788.

Jin Su, Runnan Fang, Yeqiu Li, Xiaobin Wang, Shihao Cai, Pengjun Xie, Ningyu Zhang, and Fajie Yuan. 2026. U-fold: Dynamic intent-aware context folding for user-centric agents. *arXiv preprint arXiv:2601.18285*.

Annalisa Szymanski, Noah Ziems, Heather A Eicher-Miller, Toby Jia-Jun Li, Meng Jiang, and Ronald A Metoyer. 2025. Limitations of the llm-as-a-judge approach for evaluating llm outputs in expert knowledge tasks. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pages 952–966.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.

Han Wang, An Zhang, Nguyen Duy Tai, Jun Sun, Tat-Seng Chua, and 1 others. 2024. Ali-agent: Assessing llms' alignment with human values via agent-based evaluation. *Advances in Neural Information Processing Systems*, 37:99040–99088.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Weike Zhao, Chaoyi Wu, Yanjie Fan, Pengcheng Qiu, Xiaoman Zhang, Yuze Sun, Xiao Zhou, Shuju Zhang, Yu Peng, Yanfeng Wang, and 1 others. 2026. An agentic system for rare disease diagnosis with traceable reasoning. *Nature*, pages 1–10.

Xiutian Zhao, Ke Wang, and Wei Peng. 2024. Measuring the inconsistency of large language models in preferential ranking. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 171–176.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Full RHCA Annotation Rubric

Rating Scale (1–3)

A.1 1. Reasoning Transparency (R)

Does the response explain why and how in a way a novice can follow?

1 = Poor

No explanation is provided, or the reasoning is incorrect or contains hallucinated information. A novice user cannot understand how the answer was derived.

2 = Partial

Some explanation is provided, but it is incomplete, unclear, overly long, or hard to follow. Key reasoning steps are missing, skipped, or insufficiently explained. The explanation may include vague descriptions or unexplained steps, making it difficult for a novice user to fully understand how the answer works.

3 = Good

The reasoning is clearly and logically explained step by step. A novice user can understand and follow how the answer works and how the conclusion is reached.

A.2 2. Helpfulness (H)

Does the response help the user move forward on their task?

1 = Poor

Irrelevant, incorrect, or unhelpful response that does not address the task. Provides no actionable guidance or support for understanding. Only focuses on task completion, preventing the user from making meaningful progress in learning.

2 = Partial

Shows some awareness of the novice user's cognitive load. Provides some useful guidance, but it is incomplete, unclear, or lacks key steps. Support for understanding is limited. The response may include generic suggestions, partial steps, or examples that are not fully connected to the user's task. The user can make partial progress but still needs additional clarification or steps to complete the task.

3 = Good

Provides clear, accurate, and actionable guidance (e.g., steps, examples, or strategies). Supports user understanding and task completion. The user can proceed independently or complete the task. The response may also include appropriate follow-up questions to clarify the user's understanding and support deeper and reflective learning.

A.3 3. Consistency (C)

Is the response consistent with itself and prior context?

1 = Poor

Contradictory or unstable behavior across turns. Fails to incorporate prior context or focuses only on the current turn, affecting understanding and trust.

2 = Partial

Minor inconsistencies or unclear reasoning across turns. Partially incorporates prior context, but connections may be incomplete or unclear. The response may overlook earlier details, repeat information, or show weak coherence across turns, but remains somewhat understandable.

3 = Good

No contradictions; reasoning is stable and appropriately incorporates prior context. The response demonstrates reliable behavior and aligns with the user's prior interactions. It may also adapt to user

preferences and provide coherent guidance across turns.

A.4 4. Context Alignment (A)

Does the response understand and address the user's intent?

1 = Poor

Does not address the user's intent, goal, or context, or is off-topic. May introduce irrelevant, misleading, or inappropriate content that does not support the task. Fails to support the user's task or learning needs.

2 = Partial

Addresses part of the request but misses key aspects of the user's intent or context. The response is somewhat relevant but may include unnecessary or loosely connected information. Alignment with prior turns may be incomplete or inconsistent.

3 = Good

Fully addresses the user's intent, goal, and context. The response is relevant, on-target, and well-aligned with the user's needs. It supports the task effectively and fits within the ongoing interaction context.