Encoding Disagreement: Human Oversight Challenges in LLM-as-a-Judge Systems

Charles Chiang[1], Simret Gebreegziabher[1], Annalisa Szymanski[1], Yukun Yang[1], Hyo Jin Do[2], Zahra Ashktorab[2], Werner Geyer[2], Toby Li[1] and Diego Gomez-Zara[1]

[1]*University of Notre Dame, Notre Dame, IN*
[2]*IBM Research, Yorktown Heights, NY*

Abstract

The LLM-as-a-judge paradigm has emerged as a promising approach to scaling evaluation and oversight by delegating judgment tasks to LLMs. However, existing approaches to developing and validating evaluation criteria largely assume a single evaluator, simplifying the inherently social, negotiated, and contested nature of human oversight in real-world settings. As a result, critical forms of human judgment (e.g., value disagreement, domain-specific interpretation, and diversity of views) are often flattened or lost when encoded into LLM-based judges. To examine this problem in depth, we present a formative study examining how multiple human stakeholders collaboratively create, negotiate, and refine evaluation criteria for LLM-as-a-judge systems. Our findings reveal challenges in human oversight, including difficulties in establishing shared understanding, aligning values across stakeholders with different expertise and priorities, and translating nuanced human judgments into criteria that are interpretable and actionable for LLM judges. Based on these insights, we derive design implications for interactive systems that support collaborative criteria creation, including mechanisms for supporting multi-stakeholder participation, iterative alignment, and transparency over how human judgments are encoded into automated evaluators. We argue that supporting oversight of LLM-as-a-judge systems requires shifting from individual evaluators to collective oversight practices, and that designing for this complexity is essential to avoid misalignment and unaccountable AI evaluation pipelines.

Keywords

LLM-as-a-judge, Consensus Building, Multi-Stakeholder, Large Language Models

1. Introduction

As large language models (LLMs) are increasingly deployed across different domains, developing robust mechanisms for oversight and evaluation of LLM outputs has become a more central concern. While human oversight remains the gold standard for assessing quality, it is inherently costly, requiring large quantities of time, data, and knowledgeable evaluators. Traditional metrics such as BLEU and ROUGE fail to holistically capture the quality of generated outputs.

LLM-as-a-judge has emerged as a scalable alternative, in which LLMs themselves evaluate model outputs according to explicitly defined criteria [1, 2]. Prior work has shown that strong LLM-judges can achieve moderate to high agreement with human judgments across tasks such as ranking responses and summarization scoring. Crucially, these LLM-based judges require clearly articulated and nuanced evaluation criteria to produce accurate and reliable assessments. These criteria are typically created through an iterative process, by which a human evaluator refines and adjusts criteria until the LLM judge reproduces their judgments. A growing ecosystem of tools supports this criteria definition and testing process. Systems such as *Promptfoo* [3], *Stax* [4], and *Chainforge* [5] support users in defining and testing evaluation criteria across different models. *EvaluLLM* takes this a step further, allowing users to run evaluations with different prompts and criteria metrics [6]. Other systems like *EvalGen* automatically generate criteria from a history of criteria generation [7]. *MetricMate* [8], *Evallm* [9], and *EvalAssist* [10] support users in criteria generation, iteration, and testing. Evalet breaks the criteria

Joint Proceedings of the ACM Intelligent User Interfaces (IUI) Workshops 2026, March 23-26, 2026, Paphos, Cyprus
*Corresponding author.
cchiang3@nd.edu (C. Chiang); dgomezara@nd.edu (D. Gomez-Zara)
0009-0008-6079-4355 (C. Chiang)

down into fragments, which highlight different aspects of an output depending on its relevance to different criteria [11].

Despite this progress, existing systems largely assume a single evaluator defining and refining evaluation criteria. In practice, criteria development is often a collaborative process that involves multiple stakeholders with different expertise, values, and interpretations. Current tools provide limited support for making such disagreements visible, negotiating competing perspectives, or coordinating consensus around evaluation criteria. This creates a bottleneck in which subjective perspectives remain implicit or unconsidered, and where relevant domain expertise or stakeholder viewpoints are excluded. In addition, norms of collaboration in traditional software development may be insufficient for creating human-centered AI systems [12, 13]. Since the output range of LLMs is so vast, precise criteria are vital to control the outputs of a model. Without adequate consensus between evaluators, the value of additional perspectives is diminished.

To better understand the potential opportunities and challenges of employing LLM-as-a-judge approaches in collaborative settings, we present a formative study consisting of ten interviews with LLM evaluators in a variety of roles, discussing how to design for diverse human oversight in LLM-as-a-judge systems. Based on this motivation, our research questions are:

- **RQ1:** *How do multi-stakeholder teams negotiate and encode evaluation criteria for LLM-as-a-judge systems?*
- **RQ2:** *What forms of judgment are lost or transformed in this process?*
- **RQ3:** *What are the main issues for human-AI alignment when multiple stakeholders are involved in creating evaluation criteria?*

2. Formative Study

We conducted a formative qualitative study using semi-structured interviews with 10 participants who had experience working on LLM-based systems and participating in LLM evaluation efforts. The goal of this study was to understand how different teams create LLM systems, with particular attention to how individuals in different roles contribute to, coordinate around, and make decisions during the evaluation process. These insights inform the design of tools and workflows to better support LLM evaluation across stakeholders.

Interviews lasted approximately one hour and followed a semi-structured protocol. Prior to the interview, participants completed a short pre-study survey collecting demographic information and background experience. During the interviews, participants were asked to describe their role in LLM evaluation, how they interacted with other stakeholders (e.g., developers, domain experts, end users), and how evaluation activities were conducted over time. Interview questions probed topics such as coordination across roles, iteration practices, communication strategies, conflicts or breakdowns in the evaluation process, and how requirements from different stakeholders were managed. This study protocol was approved by the University of Notre Dame's Institutional Review Board (IRB).

2.1. Participants

Participants were recruited through a combination of social media and professional networks (LinkedIn, Slack), word of mouth, and the authors' institution newsletters. Recruitment material asked for participants with experience working on a team to evaluate LLM outputs, and we pre-screened participants to ensure they had sufficient experience collaborating with others in evaluation contexts and to capture a diverse set of participant roles and perspectives.

The final sample included 10 participants, with four from industry and six from academia. Participants occupied a range of roles spanning both technical and knowledge-based responsibilities in LLM evaluation, including frontend and backend developers, pedagogy domain experts, and hybrid roles combining development and domain expertise. Technical participants were primarily involved in modeling and training LLMs, testing and quality assurance, and end-user testing. Domain experts

contributed through evaluation criteria development, design feedback, testing, and data provisioning. Several participants held multiple roles within their teams, reflecting the interdisciplinary and collaborative nature of LLM evaluation work. Five participants were recruited from the same project, where they worked as domain-knowledge evaluators for a pedagogical chatbot. Additional participant details are provided in Table 1.

2.2. Qualitative Analysis

All interviews were conducted and recorded via Zoom and transcribed using online transcription tools. We analyzed the interview transcripts using thematic analysis [14]. Three researchers independently reviewed the transcripts, first engaging in close reading to familiarize themselves with the data and recording notes. Each researcher then generated initial codes to capture recurring patterns and ideas across the dataset. The research team met to discuss and reconcile these codes, after which codes were iteratively grouped into candidate themes. These themes were reviewed, refined, and synthesized into higher-level key insights, which are reported in Section 2.3.

2.3. Key Insights

2.3.1. KI1: Collaborative criteria generation highlights divisions

We found that developers and knowledge workers often work in close conjunction in inter-disciplinary teams. While developers drive system development, knowledge workers are called upon to evaluate its progress. Even though evaluation is a critical, time-consuming, and iterative process, knowledge worker participation is often limited [7]. Knowledge workers were not always involved in day-to-day progress, as many developers expressed that they need time to develop the system to a satisfactory phase before calling upon the knowledge workers for evaluation. Additionally, knowledge workers may need technical guidance to understand how to contribute meaningfully. P4 stated, *"Because most of it, when it's technical, it's, like, over my head ... not always extremely interesting in those meetings."* However, developers can be reluctant to take time to do this: *"...they just don't really like the process"* (P1). This gap in technical expertise can lead to suboptimal evaluations that place an additional burden on development experts. P2 described their work as "*converting [domain knowledge] to technical expertise" - since "business people always are proposing different solutions as well, but technically, those weren't feasible or effective."*

2.3.2. KI2: Criteria inclusion depends on a hierarchical decision making process.

Across industry and academia, teams all had a well-defined hierarchy that dictated who was responsible for final decisions. While no single stakeholder was consistently dominant, a dominant stakeholder typically held final say over the implementation of criteria. For example, P1 described a scenario in which given requirements were not implemented, saying *"they just picked some that they think the LLM could correctly do, and then they used it, and then definitely that's not capturing all the experience."* This structure is a hallmark of separation of concerns (SoC) as seen in traditional software development, which ignores the need to adapt model behavior to a diverse set of users and account for uncertainties in AI outputs [12]. This type of system can further emphasize hierarchically dominant interpretations of criteria, narrowing the values encoded. To create more well-rounded criteria, the knowledge workers who are crucial to the criteria-creation process should not be treated separately. Therefore, an extrinsically defined decision-making hierarchy can support co-creation while still allowing decisions about criteria inclusion to proceed.

2.3.3. KI3: Current workflows insufficiently support meaningful collaboration among knowledge workers

In our interviews, knowledge experts expressed a desire to consult with others to discuss the best way to handle specific topics or niche situations, something that was not always available to them. For

example, P10 stated *"I think that I would want to do it (evaluation) with a group"*, while P6 emphasized the importance of their ability to say, *"Let me go ask my colleagues."*

Though our participants emphasized the collaborative nature of knowledge work, they also identified some challenges in existing workflows. Due to potential unfamiliarity with LLM-as-a-judge systems or criteria creation, knowledge workers may not be immediately ready to jump into a collaborative discussion. For example, though P10 wanted to work in a group, they also mentioned wanting *"a little bit more time for like self-reflection first"*, similar to P9's request for *"more time to process"* before talking to the group. Having conversations too early in the familiarization process could lead to bias formation.

Workflows should account for collaboration between knowledge workers. This can ensure that criteria are more rigorous and reduce the burden of criteria drift. In addition, encouraging collaboration can also mitigate the effect of personal differences in criteria, as personal background and experience influence how criteria are expressed [8]. Ensuring that human oversight comes from diverse sources is crucial to creating an equitable LLM judge that doesn't just reflect the dominant interpretations of criteria.

2.3.4. KI4: Human-AI alignment is constrained by development experts

While recent studies in LLM-as-a-judge are described for one user interacting with an AI, the interdependency of the multi-user criteria creation process splits the goal communication and output assessment processes into two parts [15, 1]. On one hand, knowledge workers can evaluate responses from an LLM judge, but may not always be able to perform the technical steering or prompt engineering required to directly manipulate the LLM. On the other hand, developers can interact with the LLM but require requirements derived from knowledge workers. This complicates the human-AI alignment process as well. To complete the loop between evaluation and iteration and ensure equitable interpretation of criteria, the domain expert and developer must also align with each other. We can describe this as a *human-human-AI* alignment problem, where a human mediates the human-AI alignment process.

One team we interviewed used the 'show-and-tell' method, in which knowledge workers simply demonstrate the appropriate behavior, often paired with an incorrect judgment or response in the evaluation data. For example, in evaluating a chatbot, a domain expert may review an incorrect response and correct it, producing a correct response for the same input data. This information is then given to developers as a pair of positive and negative examples, which the LLM-judge should be trained to distinguish between. The burden is then placed upon the developers to interpret the implicit requirement and create criteria.

3. Design Goals

From our formative study's key insights, supporting oversight of LLM-as-a-judge systems requires shifting away from individual evaluator workflows and towards collective, co-creation oversight practices. Since the flattening of human judgment occurs when encoding criteria, interpersonal alignment must first be achieved, fostering both human-AI alignment alongside human-human alignment, which suggests that the goal should be to design interactive workflows that are able to support this three-body alignment problem. The criteria iteration process should support both iteration between the human and the LLM-judge and iteration through human-human discussion. (Sections 2.3.1, 2.3.4). While we do not have a solution to this alignment problem, we draw on the bidirectional human-AI alignment process [15, 16] to integrate different aspects of the alignment process. Therefore, our first design goal is:

> **DG1: Support interpersonal criteria alignment alongside bi-directional human-AI alignment, in order to more holistically facilitate the domain expert-developer-AI alignment process**.

Diverse human oversight relies on equity among the overseers. Especially with stakeholders of differing backgrounds, hierarchy, and separation of concerns over the workflow should be minimized

[12]. Our use case applies criteria to data, producing outputs that serve as a shared evaluation artifact. When criteria are easily operationalized, disagreements about how they should be applied are resolved through execution; remaining conflicts instead reflect differing expectations for how the LLM-judge should interpret them. The evaluations of proposed criteria should be conducted equally regardless of the authorship, and the acceptance of a proposal should not be a trivial decision. The outcome-level changes, as well as reviewer sentiment, should be considered before acceptance. This addresses the hierarchical narrowing of encoded values observed in Section 2.3.2. Thus, our second design goal is:

> **DG2: Emphasize a co-creation workflow, avoiding structured evaluation-specification workflows and separation of concerns.**

Lastly, the system should support collaboration at both stakeholder and individual levels, making disagreements surface more easily while preserving user agency (Section 2.3.3). To help create the interpersonal collaboration and criteria alignment, we draw upon Briggs' Consensus Building Theory (CBT) [17]. CBT frames consensus building as a cyclical process: articulating a proposal, evaluating commitment, discovering and diagnosing conflicts, and resolving disagreements through discussion, potentially leading to revised proposals. In our context, proposals correspond to criteria or criterion revisions, and disagreements may arise during their evaluation. Identifying the type of disagreement allows us to apply conflict resolution strategies more effectively, as well as support conflict resolution at multiple levels of disagreement. Therefore, our final design goal is:

> **DG3: Highlight and help multiple users collectively understand nuanced criteria and confidence levels.**

4. Future directions

Building on the design goals, we plan to create a system, **MultEval**, that supports diverse collaboration in developing LLM-as-a-judge criteria. To support DG1, **MultEval** will include a private iteration sandbox that surfaces evaluator–LLM disagreements and extends them into evaluator–author discussions, facilitating both human–AI and interpersonal alignment. To address DG2, the system will incorporate role-aware mechanisms that make stakeholder influence visible and help mitigate hierarchical dominance in criteria decisions. Finally, to realize DG3, **MultEval** will provide structured tools for disagreement diagnosis and resolution, enabling teams to surface, categorize, and negotiate nuanced differences in interpretation and confidence.

We plan to run an evaluation study to investigate the effectiveness of **MultEval** at helping teams iterate on a set of criteria. We plan to create a baseline version of **MultEval** that removes some intelligent features, allowing us to compare it against the full version. Finally, we plan to run an ablation study to investigate the specific efficacy of the features we described.

Acknowledgments

This work was partially supported in part by the Notre Dame-IBM Technology Ethics Lab, an IBM Ph.D. Fellowship, Alfred P. Sloan Foundation G-2024-22427, the U.S. National Science Foundation under grant CNS-2426395, a Google Research Scholar Award, an NVIDIA Academic Hardware Grant, an Amazon Science Award, and a gift from Adobe Inc.

5. Declaration on Generative AI

The authors used ChatGPT and Gemini to assist in writing and proofreading the paper. The usage of generative AI in writing did not affect the intent or original meaning of the text. All words written were read and checked by a human author before submission. The authors also used Gemini and Cursor to

assist in coding. The placement and inclusion of features was done solely by the authors. All decisions with respect to code was done by the authors.

References

[1] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, S. Wang, K. Zhang, Y. Wang, W. Gao, L. Ni, J. Guo, A survey on llm-as-a-judge (2025). doi:10.48550/arXiv.2411.15594.

[2] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al., Judging llm-as-a-judge with mt-bench and chatbot arena, Advances in neural information processing systems 36 (2023) 46595–46623.

[3] I. Webster, M. D'Angelo, S. Klein, G. Zang, F. Minhas, promptfoo, 2025. URL: https://promptfoo.dev.

[4] Stax, Stax - the complete toolkit for AI evaluation, https://stax.withgoogle.com/landing, 2025.

[5] I. Arawjo, C. Swoopes, P. Vaithilingam, M. Wattenberg, E. L. Glassman, Chainforge: A visual toolkit for prompt engineering and llm hypothesis testing, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, 2024, pp. 1–18.

[6] M. Desmond, Z. Ashktorab, Q. Pan, C. Dugan, J. M. Johnson, Evalullm: Llm assisted evaluation of generative outputs, in: Companion Proceedings of the 29th International Conference on Intelligent User Interfaces, ACM, Greenville SC USA, 2024, p. 30–32. doi:10.1145/3640544.3645216.

[7] S. Shankar, J. Zamfirescu-Pereira, B. Hartmann, A. Parameswaran, I. Arawjo, Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences, in: Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology, 2024, pp. 1–14.

[8] S. A. Gebreegziabher, C. Chiang, Z. Wang, Z. Ashktorab, M. Brachman, W. Geyer, T. J.-J. Li, D. Gómez-Zará, Metricmate: An interactive tool for generating evaluation criteria for llm-as-a-judge workflow, in: Proceedings of the 4th Annual Symposium on Human-Computer Interaction for Work, 2025, pp. 1–18. doi:10.1145/3729176.3729199.

[9] T. S. Kim, Y. Lee, J. Shin, Y.-H. Kim, J. Kim, Evallm: Interactive evaluation of large language model prompts on user-defined criteria, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, 2024, pp. 1–21.

[10] Z. Ashktorab, M. Desmond, Q. Pan, J. M. Johnson, M. S. Cooper, E. M. Daly, R. Nair, T. Pedapati, H. J. Do, W. Geyer, Aligning human and llm judgments: Insights from evalassist on task-specific evaluations and ai-assisted assessment strategy preferences (2025). doi:10.48550/arXiv.2410.00873.

[11] T. S. Kim, H. Lee, Y. Lee, J. Seering, J. Kim, Evalet: Evaluating large language models by fragmenting outputs into functions (2025). doi:10.48550/arXiv.2509.11206.

[12] H. Subramonyam, J. Im, C. Seifert, E. Adar, Solving separation-of-concerns problems in collaborative design of human-ai systems through leaky abstractions, in: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, pp. 1–21.

[13] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al., Guidelines for human-ai interaction, in: Proceedings of the 2019 chi conference on human factors in computing systems, 2019, pp. 1–13.

[14] V. Braun, V. Clarke, Using thematic analysis in psychology, Qualitative research in psychology 3 (2006) 77–101.

[15] M. Terry, C. Kulkarni, M. Wattenberg, L. Dixon, M. R. Morris, Interactive ai alignment: Specification, process, and evaluation alignment (2024). doi:10.48550/arXiv.2311.00710.

[16] H. Shen, T. Knearem, R. Ghosh, K. Alkiek, K. Krishna, Y. Liu, Z. Ma, S. Petridis, Y.-H. Peng, L. Qiwei, et al., Towards bidirectional human-ai alignment: A systematic review for clarifications, framework, and future directions, arXiv preprint arXiv:2406.09264 2406 (2024) 1–56.

[17] R. O. Briggs, G. L. Kolfschoten, G.-J. d. Vreede, Toward a theoretical model of consensus building, AMCIS 2005 Proceedings (2005) 12.

A. Formative Study Participants

Table 1
Participant Attributes

ID	Gender	Education	Primary Industry	Stakeholder Type	Evaluation Roles
P1	F	Doctorate	Education	Frontend Developer; Domain Expert	Modeling/Training; End-user Testing
P2	F	Doctorate	Software, Information Services, and Data Processing	Backend Developer; Domain Expert	Modeling/Training
P3	M	Bachelors	Telecommunications	Backend Developer	Modeling/Training; End-user Testing
P4	M	Doctorate	Education	Domain Expert	Domain Expert
P5	M	Bachelors	Software, Information Services, and Data Processing	Backend Developer	Testing/QA; Modeling/Training; End-user Testing
P6	M	Doctorate	Pedagogy Support	Domain Expert	Testing; Data Provision; Criteria Development
P7	F	Doctorate	Education Administration	Domain Expert	Design; Criteria Development
P8	M	Doctorate	Pedagogy Support	Domain Expert	Testing; Design; Data Provision; Criteria Development
P9	F	Doctorate	Pedagogy	Domain Expert	Testing; Criteria Development
P10	M	Doctorate	Education	Domain Expert	Testing; Criteria Development