[SHORT] RIFT: A RUBRIC FAILURE MODE TAXONOMY AND AUTOMATED DIAGNOSTICS

Zhengyang Qi
Snorkel AI

Charles Dickens
Snorkel AI

Derek Pham
Snorkel AI

Amanda Dsouza
Snorkel AI

Armin Parchami
Snorkel AI

Frederic Sala
Snorkel AI
Department of Computer Sciences
University of Wisconsin–Madison

Paroma Varma
Snorkel AI

ABSTRACT

Rubric-based evaluation is widely used in LLM benchmarks and training pipelines for open-ended, less verifiable tasks. While prior work has demonstrated the effectiveness of rubrics using downstream signals such as reinforcement learning outcomes, there remains no principled way to diagnose rubric quality issues from such aggregated or downstream signals alone. To address this gap, we introduce **RIFT: RubrIc Failure mode Taxonomy**, a taxonomy for systematically characterizing failure modes in rubric composition and design. RIFT consists of eight failure modes organized into three high-level categories: *Reliability Failures*, *Content Validity Failures*, and *Consequential Validity Failures*. RIFT is developed using grounded theory by iteratively annotating rubrics drawn from five diverse benchmarks spanning general instruction following, code generation, creative writing, and expert-level deep research, until no new failure modes are identified. We evaluate the consistency of the taxonomy by measuring agreement among independent human annotators, observing fair agreement overall (**87% pairwise agreement** and **0.64 average Cohen's kappa**). Finally, to support scalable diagnosis, we propose automated rubric quality metrics and show that they align with human failure-mode annotations, achieving up to 0.86 F1.

1 INTRODUCTION

Rubrics have become a central component of recent large language model (LLM) benchmarks and training pipelines, providing rich, interpretable, and scalable evaluation signals. They are widely used in evaluating open-ended generation tasks—ranging from general instruction following (He et al., 2025) to research planning (Goel et al., 2025), as well as professional applications such as healthcare, law, and finance—where fully verifiable ground truth is often unavailable (Arora et al., 2025; Akyürek et al., 2025; Shi et al., 2026). By specifying task-specific textual criteria and scoring model outputs with an LLM-as-a-judge, rubric-based evaluation can substantially improve the alignment between automated rewards and human judgments (Sirdeshmukh et al., 2025).

Despite their growing importance, principled evaluation of rubric quality remains largely unexplored. Considerable effort has been devoted to designing human annotation workflows (Akyürek et al., 2025) and to automatically generating rubrics (Viswanathan et al., 2025; Xie et al., 2025; Liu et al., 2026; Rezaei et al., 2025), but the quality of the resulting rubrics is rarely assessed. Instead, rubric quality is typically inferred indirectly through downstream performance (e.g., reinforcement learning outcomes) or agreement between rubric-based LLM judges and human preferences. However, such downstream signals conflate rubric quality with other factors, including judge behavior and task formulation, making it difficult to isolate failures caused by the rubric itself. As a result, rubrics produced through heterogeneous pipelines cannot be meaningfully compared, and there is no principled way to characterize *how a rubric itself fails*.

We introduce **RIFT**, the **RubrIc Failure mode Taxonomy**, a taxonomy for diagnosing failures in evaluation rubrics. RIFT is derived through a grounded-theory process (Glaser & Strauss, 1967b)

based on expert critiques of diverse rubrics drawn from both human-authored and automatically generated sources. Using five representative human-curated and synthetic rubric data sources spanning general instruction following, code generation, creative writing, and expert-level deep research, we identify eight recurring failure modes and organize them into three higher-level dimensions of rubric quality: *reliability*, *content validity*, and *consequential validity*. We also observe systematic differences between human-authored and synthetic rubrics, motivating human-in-the-loop rubric creation that combines broad synthetic coverage with expert refinement. Although the taxonomy is grounded in the rubric collections studied here, the grounded-theory-based construction workflow is general and can be readily applied to new domains and rubric generation procedures.

Beyond defining the taxonomy, we study whether these rubric failures can be identified consistently. We validate RIFT through expert annotation and inter-annotator agreement among three independent human annotators, and further develop automated diagnostics that approximate RIFT labels using a combination of LLM-based classification and agreement- and stability-based signals. These diagnostics enable scalable analysis of rubric quality and make RIFT practical for real-world rubric development and iteration workflows. Our contributions are summarized as follows:

- **Taxonomy development.** We introduce RIFT, a rubric failure-mode taxonomy derived using grounded theory (Glaser & Strauss, 1967b), which identifies eight failure modes organized into three high-level categories: *Reliability*, *Content Validity*, and *Consequential Validity*.
- **Systematic empirical grounding.** We construct and analyze a dataset of 85 diverse rubrics with 255 expert annotations, drawn from five representative benchmarks and covering both human-authored and automatically generated rubrics.
- **Automated diagnostics.** We develop scalable automatic signals for detecting RIFT failure modes and achieve high agreement with expert annotations, reaching up to 0.86 F1.

2 Related Work

Task-Specific Rubrics as Evaluation. Task-specific rubrics verify LLM performance by using weighted textual criteria, evaluated by an LLM-as-a-judge (LLMaJ) and aggregated into a single reward signal. Rubric-based evaluation improves agreement with human judgments (Sirdeshmukh et al., 2025) and is widely used for benchmarks in unverifiable domains (e.g., healthcare, law, and finance) (Arora et al., 2025; Akyürek et al., 2025; Shi et al., 2026) and open-ended tasks such as general-purpose instruction following (He et al., 2025) and research planning (Goel et al., 2025).

Issues With LLM Evaluation Benchmarks. Recent studies show that LLM benchmarks suffer from unreliable outcome verification, including widely used benchmarks such as SWE-bench-Verified (Chowdhury et al., 2024) and τ-bench (Yao et al., 2024) (Zhu et al., 2025b). Beyond outcome verification, naive LLM-as-Judge evaluation can introduce systematic bias and overconfident estimates which requires calibrated reporting (Lee et al., 2026). Prior work also reports substantial evaluation variance (Madaan et al., 2024) and other weaknesses in current benchmarking practices (Reuel et al., 2024; Eriksson et al., 2025), motivating the need for more reliable evaluation design.

Failure Mode Taxonomy. Constructing failure mode taxonomies via human annotation is a common approach for identifying challenges in LLM-based systems. Zhu et al. (2025a) synthesises human analyses of agent trajectories into a unified failure taxonomy. Ma et al. (2025) and Cemri et al. (2025) further adopt grounded theory (Glaser & Strauss, 1967a) to iteratively build failure taxonomies through multi-rater consensus.

Prior work on benchmark reliability has largely focused on verifiable, agentic settings, while rubric-based evaluation in open-ended domains still lacks principled methods for assessing rubric quality in isolation. We therefore focus on evaluating rubric design and composition across datasets, with a particular emphasis on identifying failure modes. To our knowledge, RIFT is the first failure-mode taxonomy specifically designed to uncover structural shortcomings in rubrics themselves.

3 The RubrIc Failure mode Taxonomy

We define the **RubrIc Failure mode Taxonomy (RIFT)** as a generic framework for identifying and classifying failure modes in evaluation rubrics (Tab. 1). To avoid imposing a top-down schema,

Table 1: RIFT: RubrIc Failure Mode Taxonomy. See Appendix C for complete descriptions.

Category	Failure mode	Description	IRR		
			C-κ	K-α	PWA
Reliability Failures	Subjective	Use of inherently subjective evaluative terms without an anchor to objective expectations.	0.59	0.60	86.7%
	Non-Atomic	Missing a parseable, consistently scorable structure, or presence of bundled criteria that prevents partial credit.	0.54	0.54	86.7%
	Ungrounded	One or more criteria require verification against information that is factually checkable, but fail to provide sufficient grounding or bounds.	0.73	0.74	86.7%
Content Validity Failures	Misaligned or Rigid	The rubric (a) grades the wrong objective, or (b) imposes unnecessarily strict requirements.	0.57	0.58	80.0%
	Missing Criteria	The rubric provides no criterion to evaluate at least one expected requirement.	0.69	0.69	86.7%
Consequential Validity Failures	Hackable	A responder could easily achieve a top score without materially improving quality.	0.74	0.72	93.3%
	Low Signal	The rubric would give nearly equivalent scores to many different-quality responses.	0.86	0.86	93.3%
	Redundant Criteria	Two or more criteria evaluate the same requirement, such that the same behavior is rewarded or penalized multiple times.	0.70	0.67	86.7%
Overall (mean)			0.64	0.60	87.4%

RIFT is developed using **grounded theory** (Glaser & Strauss, 1967b), in which failure modes are iteratively derived from expert annotations and open-ended feedback. The resulting taxonomy organizes rubric failures into three core categories: *Reliability* (consistency and reproducibility of judgments), *Content Validity* (alignment between rubric criteria and the intended evaluation target), and *Consequential Validity* (downstream usefulness and discriminative power of the rubric).

Data Sources. To ground the taxonomy in a wide range of settings, we analyze 85 rubrics with 255 expert annotations drawn from five data sources, spanning both human-authored and automatically generated rubrics:

- *Human-crafted.* ADVANCEDIF (He et al., 2025) provides expert-authored evaluation criteria for complex instruction following in general conversational tasks, and RESEARCHRUBRICS (Sharma et al., 2025) focuses on factual grounding and multi-step analytical reasoning for deep research agents.
- *Synthetic.* WILDCHECKLISTS (Viswanathan et al., 2025) derives fine-grained rubric requirements from observed failure patterns, OPENRUBRICS (Liu et al., 2026) constructs rubrics via contrastive analysis of preference pairs, and AUTORUBRICS (Xie et al., 2025) induces reward-oriented rubrics through symbolic search. Together, these datasets and generation frameworks cover a broad range of domains, including coding and creative writing.

Taxonomy Development Pipeline. We develop RIFT using a constant comparative method over four sequential iterations, with the 85 rubrics split into four subsets and processed in order. Each iteration consists of three individual expert annotations followed by panel taxonomy discussion and revision; the first iteration additionally includes a bootstrapping step to initialize the taxonomy.

1. *Bootstrapping (first iteration only).* Experts independently write open-ended *rubric critiques* for each rubric; the critiques are provided to GPT-5.2 to propose an initial taxonomy.
2. *Labeling and feedback.* Experts label each rubric using the current taxonomy and document failure modes not captured by existing categories, as well as issues in the taxonomy itself (e.g., description inaccuracies and structural inconsistencies).

Table 2: Prevalence of RIFT failure modes in human-crafted vs. synthetic rubrics. Entries are the fraction of rubrics in the subset annotated with the failure mode.

	Subjective	Non-Atomic	Ungrounded	Misaligned /Rigid	Missing	Redundant	Low Signal	Hackable
Human-curated	52.6%	26.3%	42.1%	63.2%	47.4%	26.3%	21.1%	0.0%
Synthetic	86.7%	60.0%	46.7%	20.0%	36.7%	23.3%	40.0%	13.3%

3. *Panel refinement.* The three-expert panel reviews all feedback and updates the taxonomy, after which the next annotation round begins.
4. *Convergence and finalization.* The process terminates when experts agree that no new failure modes and rubrics revisions are needed, yielding the final taxonomy in Tab. 1.

Validation and Agreement Analysis. We assess theoretical saturation and consistency of RIFT through a final manual annotation round. We observe fair to substantial agreement across identified failure modes, with an average Cohen's kappa (C-κ) of 0.64, Krippendorff's alpha (K-α) of 0.60, and a pairwise agreement rate (PWA) of 87%. A per–failure-mode analysis shows that some categories are more difficult to annotate consistently. In particular, *Misaligned or Rigid*, which captures cases where a rubric evaluates the wrong objective or imposes overly strict requirements, has the lowest PWA, reflecting variation in annotators' thresholds for misalignment and strictness. In contrast, annotators show high agreement on the *Low Signal* category, consistently identifying rubrics that fail to provide sufficient evaluation signal.

Correlation With Human Preferences. On OPENRUBRICS and AUTORUBRICS, we use judge–human preference agreement as an external validation signal for RIFT and find that rubrics with more RIFT failure mode labels are significantly more likely to exhibit judge–human misalignment (Pearson's r = 0.162, p = 0.0021; average count = 3.66 vs. 3.15, $p <$ 1e-4) failure modes for misaligned vs. aligned cases), providing quantitative evidence that RIFT captures rubric properties relevant to downstream human-aligned evaluation.

Human-Crafted vs Synthetic Comparison. Tab. 2 summarizes the prevalence of RIFT failure modes identified in human-crafted vs. synthetic rubrics. Human-crafted rubrics tend to be more *reliable* and *consequentially valid*, but are more often *Misaligned or Rigid*, reflecting overly strict or incorrect assumptions. *Missing Criteria* arises in both settings: synthetic rubrics are often encompassing without precision, which can fail to operationalize criteria, while human-crafted rubrics tend to be more specific but can still omit some requirements. These patterns motivate *human-in-the-loop* rubric creation pipelines that use synthetic generation for broad coverage, then rely on expert review to sharpen and correct misalignments.

4 AUTOMATED RIFT EVALUATORS

To enable scalable assessment of rubric quality under RIFT, we develop automated evaluators for each failure mode. We evaluate these signals on a held-out test set of 50 rubrics drawn from ADVANCEDIF, RESEARCHRUBRICS, WILDCHECKLISTS, OPENRUBRICS, and AUTORUBRICS, which are manually annotated using the RIFT taxonomy. We implement several automated signals and examine how well they align with expert diagnostic labels:

1. **LLM-as-Judge (LLMaJ).** A rubric-conditioned failure-mode classifier trained using grounded-theory annotations and the RIFT taxonomy. We use GPT-5.2 as the classifier and provide the taxonomy as part of the prompt (see Appendix B).
2. **Inter-rater reliability (IRR).** An agreement-based signal computed as pairwise agreement (PWA) over rubric-conditioned preference labels produced by four preference labelers (GPT-5 mini, Claude Haiku 4.5, Gemini 3 Flash, and GPT-5.2.) The preferences are generated over all pairs of responses generated by six models (GPT-5 mini, GPT-5.2, Claude Haiku 4.5, Claude Sonnet 4.5, Gemini 3 Flash, and Gemini 3 Pro.)
3. **Alignment.** An accuracy-based signal measuring how often weaker preference labelers (GPT-5 mini, Claude Haiku 4.5, and Gemini 3 Flash) agree with GPT-5.2 on the same rubric-conditioned response pairs.

Table 3: Automated RIFT evaluator alignment, as measured by F1, with expert annotations.

Failure mode	LLMaJ F1	IRR F1	Alignment F1	Reward variance F1
Subjective	**0.861**	0.843	0.854	0.843
Non-Atomic	0.644	**0.667**	0.656	0.629
Ungrounded	0.444	0.625	0.625	**0.717**
Misaligned or Rigid	**0.750**	0.554	0.554	0.562
Missing Criteria	**0.686**	0.585	0.576	0.567
Hackable	0.000	**0.400**	**0.400**	0.267
Low Signal	**0.667**	0.525	0.517	0.519
Redundant Criteria	**0.643**	0.429	0.444	0.467

4. **Reward variance.** A stability-based signal defined as the variance of the aggregate rubric score produced by a rubric-conditioned LLMaJ (GPT-5.2) over four independent responses generated by GPT-5 mini for each test input.

Alignment with Expert Annotations. Results are reported in Tab. 3. For each evaluator, failure-mode pair, we report F1 at the threshold that maximizes F1 on the test set. The RIFT LLMaJ achieves moderate to good alignment on a subset of failure modes, Subjective and Misaligned or Rigid. However, there is significant room for improvement of the LLMaJ on the Non-Atomic, Ungrounded, and Hackable failure modes. We recommend using non-LLMaJ-based methods to complement the LLMaJ in these cases as a concrete strategy to improve detection for these challenging categories. We expect further refinements of the LLMaJ via prompt tuning and curating in-context examples would further improve the performance of the LLMaJ.

Grouping failure modes into reliability, content, and consequential categories highlights observed difficulties of automated rubric-quality evaluation. Reliability failures are identified more accurately, likely because they result in inconsistent rubric-based judgments that non-LLMaJ signals measure directly, and because RIFT definitions provide a clear scoring guidance for LLMaJs. In contrast, content and consequential failures require detecting substance gaps or inaccuracies in rubric criteria and anticipating downstream effects.

5 Conclusions and Future Work

We introduced RIFT: a taxonomy for rubric failure modes derived from expert annotations of real evaluation rubrics and iterative refinement. A key takeaway from the RIFT development and annotation process is that synthetic rubrics often provide broad but imprecise coverage, while human-crafted rubrics are typically sharper but can be misaligned or overly rigid. Combining generation with targeted expert review can capture coverage while correcting misalignments.

We also implemented automated RIFT evaluators to detect and label failure modes, enabling scalable analysis. We expect expert-in-the-loop workflows to remain important for assessing content and consequential validity. Experts can verify coverage, relevance, and missing criteria (with LLM critics to accelerate annotation (McAleese et al., 2024)), while consequential validity may require methods that directly measure a rubric's impact in downstream use cases.

RIFT is not exhaustive. Scaling the annotation process and validating RIFT more broadly (e.g., across domains, tasks, and stakeholder settings) are important directions for future work. Moreover, our there is room for improving our automated evaluators' alignment with expert judgments. Finally, future work should explore techniques that more directly measure the downstream impacts of each failure mode and measure the impact of RIFT guided rubric refinements on training pipeline or evaluation benchmark outcomes.

REFERENCES

Afra Feyza Akyürek, Advait Gosai, Chen Bo Calvin Zhang, Vipul Gupta, Jaehwan Jeong, Anisha Gunjal, Tahseen Rabbani, Maria Mazzone, David Randolph, Mohammad Mahmoudi Meymand, Gurshaan Chattha, Paula Rodriguez, Diego Mares, Pavit Singh, Michael Liu, Subodh Chawla, Pete Cline, Lucy Ogaz, Ernesto Hernandez, Zihao Wang, Pavi Bhatter, Marcos Ayestaran, Bing Liu, and Yunzhong He. Prbench: Large-scale expert rubrics for evaluating high-stakes professional reasoning, 2025. URL https://arxiv.org/abs/2511.11562.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health, 2025. URL https://arxiv.org/abs/2505.08775.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent llm systems fail?, 2025. URL https://arxiv.org/abs/2503.13657.

Neil Chowdhury, James Aung, Chan Jun Shern, Oliver Jaffe, Dane Sherburn, Giulio Starace, Evan Mays, Rachel Dias, Marwan Aljubeh, Mia Glaese, Carlos E. Jimenez, John Yang, Leyton Ho, Tejal Patwardhan, Kevin Liu, and Aleksander Madry. Introducing swe-bench verified, 2024. URL https://openai.com/index/introducing-swe-bench-verified/.

Maria Eriksson, Erasmo Purificato, Arman Noroozian, Joao Vinagre, Guillaume Chaslot, Emilia Gomez, and David Fernandez-Llorca. Can we trust ai benchmarks? an interdisciplinary review of current issues in ai evaluation, 2025. URL https://arxiv.org/abs/2502.06559.

Barney G. Glaser and Anselm L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Company, Chicago, 1967a.

Barney G. Glaser and Anselm L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Company, 1967b.

Shashwat Goel, Rishi Hazra, Dulhan Jayalath, Timon Willi, Parag Jain, William F. Shen, Ilias Leontiadis, Francesco Barbieri, Yoram Bachrach, Jonas Geiping, and Chenxi Whitehouse. Training ai co-scientists using rubric rewards, 2025. URL https://arxiv.org/abs/2512.23707.

Yun He, Wenzhe Li, Hejia Zhang, Songlin Li, Karishma Mandyam, Sopan Khosla, Yuanhao Xiong, Nanshu Wang, Xiaoliang Peng, Beibin Li, Shengjie Bi, Shishir G. Patil, Qi Qi, Shengyu Feng, Julian Katz-Samuels, Richard Yuanzhe Pang, Sujan Gonugondla, Hunter Lang, Yue Yu, Yundi Qian, Maryam Fazel-Zarandi, Licheng Yu, Amine Benhalloum, Hany Awadalla, and Manaal Faruqui. Advancedif: Rubric-based benchmarking and reinforcement learning for advancing llm instruction following, 2025. URL https://arxiv.org/abs/2511.10507.

Chungpa Lee, Thomas Zeng, Jongwon Jeong, Jy yong Sohn, and Kangwook Lee. How to correctly report llm-as-a-judge evaluations, 2026. URL https://arxiv.org/abs/2511.21140.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. Openrubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment, 2026. URL https://arxiv.org/abs/2510.07743.

Xuyan Ma, Xiaofei Xie, Yawen Wang, Junjie Wang, Boyu Wu, Mingyang Li, and Qing Wang. Diagnosing failure root causes in platform-orchestrated agentic systems: Dataset, taxonomy, and benchmark, 2025. URL https://arxiv.org/abs/2509.23735.

Lovish Madaan, Aaditya K. Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks, 2024. URL https://arxiv.org/abs/2406.10229.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. LLM critics help catch LLM bugs, 2024. URL https://arxiv.org/abs/2407.00215.

Anka Reuel, Amelia Hardy, Chandler Smith, Max Lamparth, Malcolm Hardy, and Mykel J. Kochenderfer. Betterbench: Assessing ai benchmarks, uncovering issues, and establishing best practices. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 21763–21813. Curran Associates, Inc., 2024. doi: 10.52202/079017-0685. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/26889e8359e7ef8a7f5d77457364ca55-Paper-Datasets_and_Benchmarks_Track.pdf.

MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Bing Liu, Yunzhong He, and Afra Feyza Akyürek. Online rubrics elicitation from pairwise comparisons, 2025. URL https://arxiv.org/abs/2510.07284.

Manasi Sharma, Chen Bo Calvin Zhang, Chaithanya Bandi, Clinton Wang, Ankit Aich, Huy Nghiem, Tahseen Rabbani, Ye Htet, Brian Jang, Sumana Basu, Aishwarya Balwani, Denis Peskoff, Marcos Ayestaran, Sean M. Hendryx, Brad Kenstler, and Bing Liu. Researchrubrics: A benchmark of prompts and rubrics for evaluating deep research agents, 2025. URL https://arxiv.org/abs/2511.07685.

Yuzhen Shi, Huanghai Liu, Yiran Hu, Gaojie Song, Xinran Xu, Yubo Ma, Tianyi Tang, Li Zhang, Qingjing Chen, Di Feng, Wenbo Lv, Weiheng Wu, Kexin Yang, Sen Yang, Wei Wang, Rongyao Shi, Yuanyang Qiu, Yuemeng Qi, Jingwen Zhang, Xiaoyu Sui, Yifan Chen, Yi Zhang, An Yang, Bowen Yu, Dayiheng Liu, Junyang Lin, Weixing Shen, Bing Zhao, Charles L. A. Clarke, and Hu Wei. Plawbench: A rubric-based benchmark for evaluating llms in real-world legal practice, 2026. URL https://arxiv.org/abs/2601.16669.

Ved Sirdeshmukh, Kaustubh Deshpande, Johannes Mols, Lifeng Jin, Ed-Yeremai Cardona, Dean Lee, Jeremy Kritz, Willow Primack, Summer Yue, and Chen Xing. Multichallenge: A realistic multi-turn conversation evaluation benchmark challenging to frontier llms, 2025. URL https://arxiv.org/abs/2501.17399.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models, 2025. URL https://arxiv.org/abs/2507.18624.

Lipeng Xie, Sen Huang, Zhuo Zhang, Anni Zou, Yunpeng Zhai, Dingchao Ren, Kezun Zhang, Haoyuan Hu, Boyin Liu, Haoran Chen, Zhaoyang Liu, and Bolin Ding. Auto-rubric: Learning to extract generalizable criteria for reward modeling, 2025. URL https://arxiv.org/abs/2510.17314.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, Xiaoteng Ma, Xiaodong Yu, Gowtham Ramesh, Jialian Wu, Zicheng Liu, Pan Lu, James Zou, and Jiaxuan You. Where llm agents fail and how they can learn from failures, 2025a. URL https://arxiv.org/abs/2509.25370.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Jasjeet Sekhon, Jacob Steinhardt, Antony Kellermann, Sarah Schwettmann, Matei Zaharia, Ion Stoica, Percy Liang, and Daniel Kang. Establishing best practices for building rigorous agentic benchmarks, 2025b. URL https://arxiv.org/abs/2507.02825.

A Appendix

B Prompts

This section provides the prompt templates used in the RIFT pipeline. Template variables are denoted with double curly braces (e.g., {{variable}}).

B.1 LLM-AS-A-JUDGE ANNOTATION PROMPT

The following prompt template is used by the automated LLM-as-a-Judge (LLMaJ) evaluator described in Section 4. For each rubric under evaluation, the prompt is populated with the complete RIFT taxonomy—including all failure mode descriptions and pass/fail examples—alongside the rubric's input context and rubric text. The LLMaJ returns structured output: for each identified failure mode, the model provides the failure mode label, a justification for why the failure mode applies, and a direct quote from the rubric exhibiting the issue.

```
You are an expert at evaluating rubric quality.  Analyze the following
rubric against the failure mode taxonomy and identify any issues.  The
rubric is designed to evaluate the quality of an AI model's response
to a given prompt.

## Failure Mode Taxonomy

[For each failure mode in the taxonomy:]

### {{failure_mode.label}}
Description:  {{failure_mode.description}}

**Pass Examples** (rubric does NOT exhibit this failure mode):
[For each pass example:]
- Input:  {{example.input_context[:150]}}...
Rubric:  {{example.rubric[:200]}}...

**Fail Examples** (rubric DOES exhibit this failure mode):
[For each fail example:]
- Input:  {{example.input_context[:150]}}...
Rubric:  {{example.rubric[:200]}}...

[End of taxonomy loop. If no failure modes are defined:]
No failure modes defined yet - suggest any issues you observe.

## Input Context
{{input_context}}

## Rubric to Evaluate
{{rubric}}

## Task
Identify which failure modes from the taxonomy apply to this rubric
(if any).
```

The LLMaJ is configured to return a structured JSON response conforming to the following schema:

```
{
  "suggested_labels": [
    {
      "label": "<failure mode label from taxonomy>",
      "justification": "<why this failure mode applies>",
      "quote": "<specific rubric quote exhibiting the issue>"
    },
    ...
  ]
}
```

B.2 TAXONOMY REFINEMENT PROMPT

The following prompt template drives the iterative taxonomy refinement process described in Section 3 (Tab. 1). During each iteration of the grounded theory pipeline, expert annotator feedback—comprising open-ended rubric critiques and taxonomy critiques—is batched and provided to GPT-5.2 alongside the current taxonomy state. The model proposes refinements (merges, additions, clarifications, splits, removals, or renames), which are then reviewed and finalized by the expert panel.

```
You are an expert at analyzing rubric quality feedback and refining
failure mode taxonomies.  Your task is to output a complete refined
failure mode taxonomy.

## Original Failure Mode Taxonomy

This is the original taxonomy before any refinements in this session:
[For each failure mode: label and description. If none defined: "No failure modes have been defined
yet."]

## Current Running Refinement

This is the taxonomy as refined so far in this session (may be
identical to original if this is the first batch):
[For each failure mode: label, description, rationale, and counts of pass/fail examples. If none refined:
"No refinements have been made yet."]

## Annotator Feedback to Analyze

Below are annotations with two types of critiques:
- Rubric Critique:  Issues the annotator observed in the rubric that
were NOT captured by the original taxonomy labels (may suggest new
failure modes)
- Taxonomy Critique:  Critique of the ORIGINAL taxonomy (unclear
definitions, overlapping categories, missing categories, etc.).  Note:
these critiques were written against the original taxonomy, not the
running refinement.

[For each annotation in the batch:]

Annotation {{loop.index}}
Input Context:  {{item.input_context}}
Rubric:  {{item.rubric}}
Rubric Critique:  (issues not captured by original taxonomy)
{{item.rubric_critique or "None provided"}}
Taxonomy Critique:  (critique of the original taxonomy)
{{item.failure_mode_critique or "None provided"}}

[End of annotation loop]
```

The prompt further specifies a detailed **taxonomy philosophy** and **refinement guidelines** that constrain the model's proposed changes:

```
## Taxonomy Philosophy

CRITICAL: This taxonomy will be used by human annotators.  The primary
goal is to create a taxonomy that is:

    • Compact: Aim for 7–10 total failure modes. Fewer distinct categories is ALWAYS better than
      many granular ones.
    • Easily distinguishable: A human should be able to distinguish between any two fail-
      ure modes in under 30 seconds.  If two categories require careful reading to tell apart, they
```

should be merged or their distinction should be clarified by refining the label names and or the description.
- `Actionable`: Each category must be clearly applicable without ambiguity.

```
Consolidation over proliferation:  When in doubt, MERGE rather than
add.  Two failure modes that are 80% similar should become one
category, not two.  The cost of a slightly imperfect merge is far
lower than the cost of a bloated, hard-to-use taxonomy.

## Guidelines
```

- `Clear descriptions`: Each failure mode description must be clear, specific, and actionable. The description should explicitly specify HOW to determine if a rubric exhibits this failure mode. An annotator should be able to read the description and confidently apply it to any rubric.
- `No overlapping failure modes`: The taxonomy should not contain failure modes with overlapping meanings. If two labels capture the same concept, merge them or refine them to make them distinct. Do NOT add a new failure mode if its meaning already exists under a different label.
- `Self-contained rationales`: Each rationale must be a self-contained justification that will be used for manual review. It should explain WHY this failure mode exists, what evidence from critiques supports it, and how it differs from other failure modes. A reviewer should understand the rationale without needing to see the original critiques.
- `Cumulative applicability`: The refined taxonomy must be applicable to ALL critiques that have been seen in this session (including previous batches), not just the current batch. Do not remove or change failure modes in ways that would make them inapplicable to earlier critiques that supported them.

```
## Task

Analyze BOTH the rubric critiques and taxonomy critiques above.
Before adding any new failure modes, first consider whether existing
categories should be merged.

FIRST: Consider merging existing failure modes when:
```

- Two or more categories have similar descriptions or capture closely related issues
- Categories are difficult to distinguish without careful reading
- A broader category could capture multiple narrower ones without losing important distinctions
- The taxonomy has grown beyond 12 failure modes

```
PREFERRED action - merge:  Combine overlapping, redundant, or closely
related labels into one.  This is the most important refinement
action.  If you're unsure whether two categories are distinct enough,
merge them.

Add new failure modes ONLY when ALL of the following are true:
```

- The issue is clearly NOT capturable by ANY existing failure mode (even with minor rewording)
- The issue appears in MULTIPLE critiques (not just one annotation)
- The new category is easily distinguishable from ALL existing categories
- Adding it would NOT push the taxonomy beyond 12 failure modes

```
Other refinement actions:
```

- clarify: Make a label's description clearer, more specific, or more actionable (especially clarifying HOW to identify the failure mode)

```
    • split: Divide an overly broad label into more specific ones
      (use sparingly---only when a category is genuinely too broad to
      apply consistently)
    • remove: Eliminate labels that are not useful, are duplicates,
      or are too similar to other categories
    • rename: Change a label name to be more descriptive

Output:
1. failure_modes: The complete list of failure modes after applying
changes. Each failure mode should have:

    • label: concise identifier (e.g., contradictory_criteria,
      missing_edge_cases)
    • description: clear, specific, and actionable description that
      explains HOW to determine if a rubric has this failure mode
      (what to look for, what conditions must be met)
    • rationale: a self-contained justification for this failure
      mode that can be understood without seeing the original
      critiques. Explain why it exists, what patterns it captures,
      and how it differs from related failure modes. If this is a
      NEW category, explicitly explain why it cannot be captured by
      any existing category.
    • examples: REQUIRED: 3-5 pass_examples AND 3-5 fail_examples
      for each failure mode. Multiple diverse examples are essential
      for annotator training. You may use real examples from the
      annotations or synthesize clear illustrative examples. Each
      example should illustrate a distinct scenario or nuance.

2. changes_summary: A list of strings describing what changes
you made (e.g., "Added 'contradictory_criteria' based on rubric
critiques", "Clarified description of 'ambiguous_criterion'", "Merged
'x' and 'y' into 'z'")

If no changes are needed based on these critiques, return the current
running refinement unchanged with an empty changes_summary.
```

C RIFT: RUBRIC FAILURE MODE TAXONOMY

This section provides the complete descriptions of each failure mode in the RIFT taxonomy (Tab. 1), including detailed decision rules for annotation. Each failure mode includes: (1) when to apply the label, (2) how to determine whether a rubric exhibits the failure mode, and (3) boundary conditions specifying when *not* to apply the label and which alternative label to consider instead. For each failure mode, we also provide illustrative pass examples (rubrics that do *not* exhibit the failure mode) and fail examples (rubrics that *do* exhibit the failure mode). Note: in the "Do NOT apply" cross-references below, CSV-internal label identifiers have been replaced with the display names used in Tab. 1.

C.1 RELIABILITY FAILURES

Reliability failures lead to inconsistent grading across annotators or evaluation runs, reducing the reproducibility and trustworthiness of rubric-based evaluation.

Subjective. Apply when the rubric uses inherently subjective evaluative terms (e.g., "clear," "appropriate," "credible," "comprehensive," "professional," "engaging," "well-written," "good sources") and does NOT sufficiently anchor them with objective expectations.

How to determine:

- Identify criteria dominated by inherently subjective terms.

- Check whether the rubric provides ANY anchoring attempt such as:
  - concrete checklists ("includes X/Y/Z"),
  - measurable thresholds (word count, required sections, required elements),
  - examples/anti-examples of what qualifies vs does not qualify, or
  - explicit decision rules ("count as clear if it defines the term and gives one example").
- If the rubric relies primarily on grader judgment and provides no meaningful anchors, apply.

Important clarification:

- Do NOT apply if the rubric gives examples or non-trivial decision rules that explain what the subjective term means (even if the term is still somewhat subjective).

Do NOT apply if:

- The core issue is missing expected answers/tolerances or a bounded verification procedure for a groundable requirement (use *Ungrounded*).
- The requirement is entirely absent (use *Missing Criteria*).

Illustrative examples:

	Input Context	**Rubric**
PASS	Write a professional email declining a meeting.	*2 pts: Includes a decline + proposes an alternative time. 2 pts: Uses a greeting and sign-off. 1 pt: No negative or insulting language.*
FAIL	Summarize the study.	*10 pts: The summary is clear and sufficiently detailed.*

Non-Atomic. Apply when the rubric does not provide a parseable, consistently scorable structure OR uses bundled (non-atomic) criteria that prevent consistent partial credit.

Triggers (any sufficient):

Non-atomic (bundled) criteria

- One scored item bundles multiple independently scorable requirements with no partial-credit rule or separable sub-scores (e.g., "clear, comprehensive, accurate, and well-cited" as a single 10-pt item).

Do NOT apply when:

- Subparts are separately scored or the rubric provides explicit level anchors (e.g., "1 point each for A/B/C" or a 0–2 scale per dimension with definitions).
- The rubric is scorable but uses subjective language (use *Subjective*) or is missing requirements (use *Missing Criteria*).

Illustrative examples:

	Input Context	**Rubric**
PASS	Write a short answer with two supporting reasons.	*2 pts: Answers the question. 1 pt: Reason #1 supports the answer. 1 pt: Reason #2 supports the answer. 1 pt: Total length ¡=150 words.*
FAIL	Summarize the article.	*10 pts: Summary is clear, accurate, comprehensive, concise, and engaging.*

Ungrounded. Apply when the rubric requires verification that is plausibly groundable/boundable, but the rubric does not provide the necessary grounding (answer keys/acceptable variants/tolerances/decision rules) OR does not bound the verification procedure (what to check, how much to check, and how to judge conflicts).

How to determine (any sufficient):

(A) *Groundable determinate tasks lack grading anchors.* The task has a knowable target output given fixed inputs (e.g., extraction, classification, translation, math, SQL result, code output), but the rubric provides no expected answers, acceptable variants, label mappings, tolerances, or decision rules. The criterion may be clearly worded (e.g., "totals are correct"), yet graders still lack what they need to check correctness.

(B) *Open-world requirements lack bounded audit procedure.* The rubric demands broad verification (e.g., "all facts are true," "restaurants are open right now," "fully original/no plagiarism," "links work") without bounding: what to check (scope/sample size), which sources/tools are allowed, how to resolve conflicting evidence, and the pass/fail threshold.

(C) *Measurement standard is unspecified but could be made checkable.* The rubric requires a measurement that depends on an unspecified standard (e.g., "exactly 20 pages") without defining the rendering/formatting standard or offering a workable proxy.

Do NOT apply if:

- The requirement is simply missing from the rubric (use *Missing Criteria*).
- The main issue is subjective wording without anchors (use *Subjective*). The rubric provides a representative list of examples to demonstrate expected content.

Illustrative examples:

	Input Context	**Rubric**
PASS	Extract all email addresses from the text.	*1 pt per correct email address; accepted forms include plus-addressing. Gold list of emails: a@x.com, b.y@z.org, ... Deduct 1 pt per missing email.*
FAIL	Compute the correct totals for these 30 invoices.	*10 pts: Totals are correct.*

C.2 CONTENT VALIDITY FAILURES

Content validity failures arise when rubric criteria are misaligned with the intended evaluation target, either by grading the wrong objective or by failing to cover essential requirements.

Misaligned or Rigid. Apply when the rubric (a) grades the wrong objective for the prompt or embeds incorrect assumptions, OR (b) imposes unnecessarily strict/narrow requirements not asked for by the prompt or reasonably inferred from the prompt, predictably penalizing prompt-faithful high-quality answers.

How to determine (any applies):

- Wrong task / shifted objective: makes non-requested deliverables mandatory for points.
- Incorrect embedded assumptions: assumes a context not in the prompt (jurisdiction, audience, tools, constraints) and scores accordingly.
- Penalizes good practice: scores down reasonable caveats/uncertainty/safety practices when the prompt does not forbid them.
- Arbitrary brittleness/over-constraint: mandates a specific tool/library/method/structure/formatting or false precision when multiple reasonable alternatives would satisfy the prompt.

Do NOT apply when:

- The prompt itself imposes the strictness at any point (e.g., exact JSON keys, or a direct instruction from the user earlier in a chat conversation).
- The requirement is missing entirely (use *Missing Criteria*).

- The main problem is internal contradiction (use *Self-Contradictory*).
- The main problem is rubric-level proxy gaming (use *Hackable*).

Illustrative examples:

	Input Context	Rubric
PASS	Write Python code to parse CSV.	*5 pts: Correct parsing. 3 pts: Handles quoted commas. 2 pts: Includes brief usage example. (Does not mandate pandas vs csv module.)*
FAIL	Write a haiku about winter.	*5 pts: Includes at least 5 academic citations. 5 pts: Uses APA format reference list.*

Missing Criteria. Apply when the prompt implies at least one checkable must-have requirement, but the rubric provides no criterion that allows a grader to evaluate that requirement at all.

How to determine:

1. List the prompt's core requirements:
   - required deliverables/components,
   - must/must-not constraints,
   - required format/ordering/sections,
   - and genre-critical qualities the prompt clearly expects (e.g., functional correctness for code; "two sentences"; "valid JSON"; "include 10 items"; "chronological order").
2. For each requirement, check whether ANY rubric criterion covers it.
3. If one or more requirements have no corresponding criterion, apply.

Do NOT apply if:

- The rubric mentions the requirement but is vague or subjective (use *Subjective*).
- The rubric mentions the requirement but it cannot be graded consistently due to missing keys/tolerances/bounded audit steps (use *Ungrounded*).
- The rubric grades a different task or adds arbitrary constraints (use *Misaligned or Rigid*).

Illustrative examples:

	Input Context	Rubric
PASS	Return ONLY valid JSON with keys: name (string) and age (integer).	*3 pts: Output parses as JSON. 2 pts: Contains exactly keys name and age. 2 pts: name is a string; age is an integer. 1 pt: No surrounding commentary.*
FAIL	Write a 200-word email and include a subject line.	*10 pts: Tone is professional. 5 pts: Grammar and spelling are correct.*

C.3 CONSEQUENTIAL VALIDITY FAILURES

Consequential validity failures reduce the downstream usefulness and discriminative power of rubric-based evaluation, even when individual criteria may be well-defined.

Hackable. Apply when the rubric is gameable at the rubric level: a responder could easily achieve a top score by inflating proxy metrics (length, number of bullets/sections/items/citations/examples/brands, repeated keywords) without materially improving correctness, relevance, or fulfillment of the prompt—and the rubric lacks strong quality gates that tie points to substantive, prompt-aligned success.

Core question (required):

- Could I easily achieve full marks on this rubric while still not satisfying the prompt requirements or producing a low-quality response?

How to determine (any sufficient):

- Most points come from "more" ($\geq$N tips/citations/examples/pros/cons/sections) while relevance, non-duplication, correctness, and prompt-specific success conditions are weakly specified or absent.
- Rewards merely asserting attributes ("quiet," "fast Wi-Fi," "no fees") without requiring evidence, linkage to the task, or checks against duplication.
- Counting proxies dominate while key prompt requirements have only weak gates (e.g., no requirement that citations support specific claims; no requirement that items be distinct and on-topic).

Do NOT apply when:

- Quantity minimums are paired with robust quality controls that make padding ineffective (e.g., each item must be non-duplicative, tied to a specific claim or user need, and verifiably grounded/bounded).
- The main issue is that the rubric is generic and doesn't discriminate at all (use *Low Signal*).
- The main issue is a specific criterion that shifts the task or overconstrains acceptable answers (use *Misaligned or Rigid*).

Illustrative examples:

	Input Context	**Rubric**
PASS	Provide 5 study tips.	*1 pt each for 5 tips that are (a) non-duplicative and (b) each includes a concrete example of how to apply it.*
FAIL	Provide a recommendation.	*5 pts: At least 10 pros. 5 pts: At least 10 cons. (No check for relevance or duplication.)*

Low Signal. Apply when the rubric as a whole does not discriminate candidate responses well for this prompt—i.e., it would give similar (often high) scores to many substantively different-quality responses—because the criteria are all generic, conditionally irrelevant, or too easy.

How to determine (rubric-level discrimination test):

- Imagine 3–5 candidate responses ranging from weak to excellent.
- Ask: Would the rubric's criteria/weights produce nearly equivalent different scores across them based on prompt-relevant success?
- If the rubric would likely award similar scores because most criteria are low-signal (e.g., "helpful," "nice formatting," "completed the task") and there are few/no strong quality gates tied to the prompt's real success conditions, apply.

Common signals:

- Most points are allocated to generic writing quality/tone/formatting that is not central for this prompt.
- Criteria are trivially satisfied by any minimally on-task response (e.g., "schedule exists").
- The rubric could be pasted into many unrelated tasks with little or no change.

Do NOT apply when:

- The rubric is instead missing prompt-imposed must-haves (use *Missing Criteria*).
- The rubric imposes the wrong constraints/assumptions (use *Misaligned or Rigid*).

- The rubric is gameable specifically via quantity/proxies (use *Hackable*).

Illustrative examples:

	Input Context	Rubric
PASS	Return JSON only with required keys.	*6 pts: Valid JSON with required keys. 4 pts: No extra text outside JSON.*
FAIL	Return only a SQL query.	*5 pts: Response is helpful. 5 pts: Uses appropriate tone.*

Redundant Criteria. Apply when two or more rubric criteria substantially evaluate the same underlying requirement such that the same behavior is rewarded/penalized multiple times.

How to determine:

- Additional-signal test (primary): If these criteria are separate, do you get genuinely different evaluation signal, or are you just re-awarding the same property?
- Remove-one test: If removing a criterion would not meaningfully change what is evaluated (only point allocation), it is redundant.
- Includes near-duplicates and cases where one criterion fully subsumes another.

Important clarification (do NOT apply for mere dependencies):

- Do NOT apply just because criteria are related or one tends to enable another.
- If Criterion B is a prerequisite/necessary condition for Criterion A (or vice versa) but still measures a distinct dimension (e.g., "valid JSON" and "has required keys"; "code compiles" and "passes tests"), that is NOT redundancy.

Do NOT apply when criteria are related but clearly distinct checks (e.g., factual accuracy vs clarity; format compliance vs correctness; presence of citations vs whether citations support claims).

Illustrative examples:

	Input Context	Rubric
PASS	Write a research summary with citations.	*3 pts: Claims are supported by citations. 2 pts: Writing is well-organized. 2 pts: Includes limitations of the evidence.*
FAIL	Essay rubric.	*5 pts: Clear writing. 5 pts: Clarity of prose. 5 pts: Writing is easy to understand.*