The Rules of the Game: A Survey of Rubrics for Large Language Models

Wenhan Liu[📦], **Jiajie Jin**, **Zhaoheng Huang**, **Tongyu Wen**, **Guanting Dong**, **Ziliang Zhao**, **Yutao Zhu**, **Zhicheng Dou**[‡], **Ji-Rong Wen**

📦 Project Organizer. ‡Core Supervisors.

Affiliations: Renmin University of China

Large language models (LLMs) have rapidly evolved from general text generators into increasingly capable systems for reasoning, decision-making, tool use, and long-horizon problem solving. As their application scenarios expand toward open-ended and high-stakes tasks, including deep research, medical diagnosis, multimodal generation, and agentic tool use, the question of how to specify, optimize, and evaluate model responses has become increasingly important. Simple correctness signals, holistic preference scores, and unconstrained LLM-based judgments are often insufficient for these settings, where response quality depends on multiple criteria such as factuality, completeness, safety, reasoning soundness, evidence grounding, and practical utility. Rubrics have therefore emerged as a promising mechanism for making evaluation standards explicit and operational. By decomposing broad quality expectations into structured and interpretable criteria, rubrics provide an interface for both training supervision and model evaluation.

This survey presents a comprehensive and systematic overview of rubric-based research for LLMs. We first clarify the concept of rubrics and distinguish it from closely related concepts, including reward models, verifiable rewards, and LLM-as-a-judge. We then organize existing studies along three major directions. First, we summarize existing **rubric construction methods** and organize them into four categories: direct generation, contrastive generation, iterative refinement, and online or co-evolving generation. Second, we examine how rubrics support the **training of policy models and reward models**. For policy model training, we organize existing studies by their training mechanisms. For reward model training, we categorize prior work according to the functional roles that rubrics play in reward modeling. Third, we summarize **rubric-driven task evaluation** for both general and domain-specific tasks, and discuss the evaluation benchmarks from various perspectives. Beyond consolidating existing work, we discuss a series of key **open questions**, such as rubric reward hacking, the bias in rubric-based evaluation, personalization, and rubric safety. We hope this survey can serve as a structured reference for current research and a conceptual foundation for developing rubrics as transparent, adaptive, and trustworthy interfaces for future LLM systems. *Given the rapid development of rubric-based research, we will keep this survey updated to incorporate new advances and emerging directions in this area.*

Date: May 22, 2026
Main Contact: lwh@ruc.edu.cn, dou@ruc.edu.cn
Github: https://github.com/8421BCD/Rubrics_Survey

Note: As the literature on rubrics for LLMs is expanding rapidly, this survey may not cover every relevant study. We encourage readers to contact us by email (lwh@ruc.edu.cn) or open an issue on GitHub if they notice any missing papers.

Contents

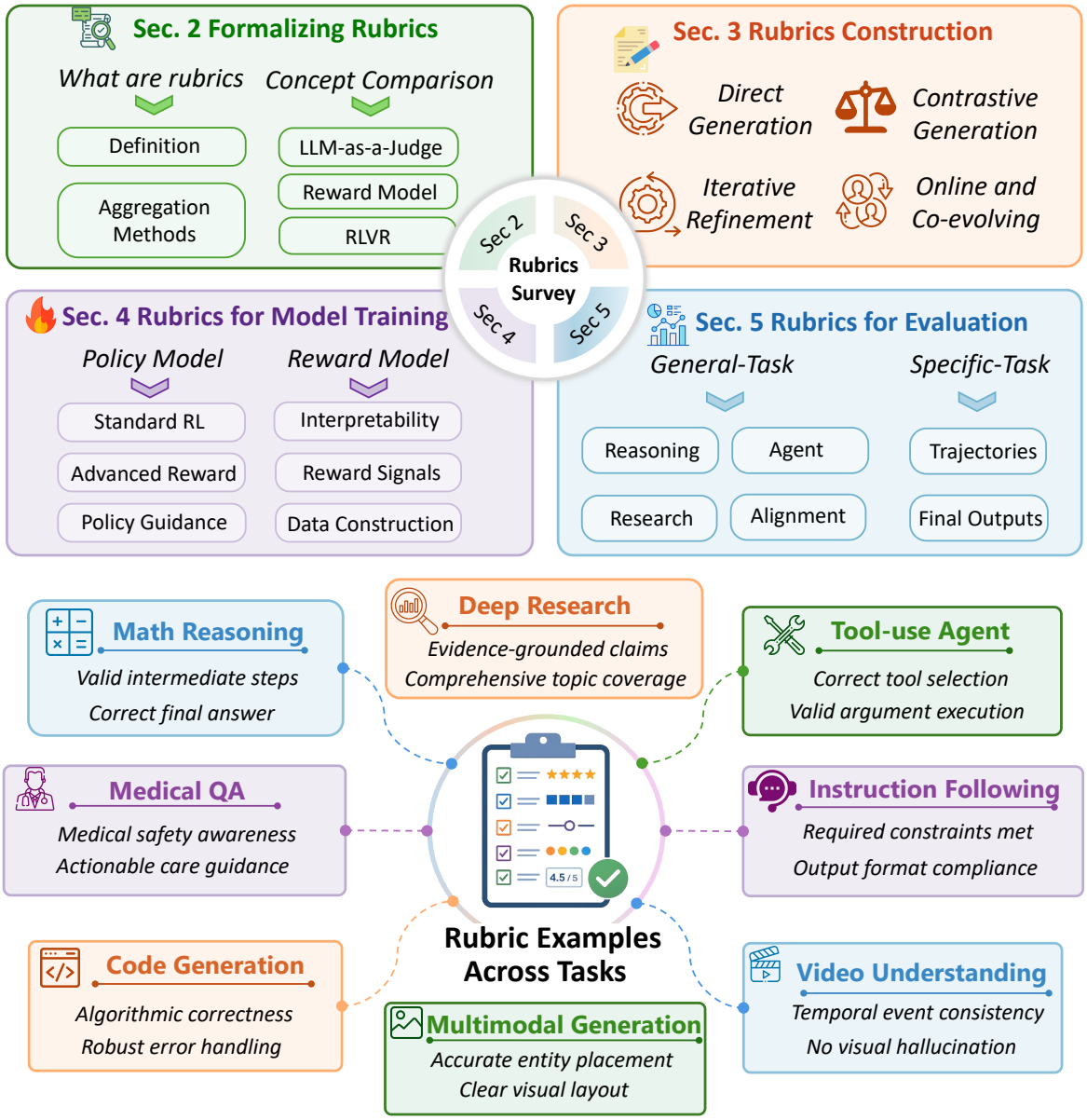

Figure 1 The upper part of the figure illustrates the chapter organization of the survey from Section 2 to Section 5. The lower part presents several concrete examples of rubrics across different tasks.

1 Introduction

The past few years have witnessed a remarkable transformation in the tasks that large language models (LLMs) are expected to handle. The early applications of LLMs focused mainly on tasks with relatively clear objectives and standardized input-output formats, such as question answering (Yang et al., 2018; Ho et al., 2020), instruction following (Ouyang et al., 2022), and code completion (Izadi et al., 2024). With the rapid development of LLM capabilities, especially in reasoning (DeepSeek-AI, 2025; Liu et al., 2025b), planning (Erdogan et al., 2025), and tool use (Dong et al., 2025a), LLMs have gradually evolved from passive answer generators into more capable AI agents. This evolution enables them to address more complex and highly open-ended tasks, including long-form report generation, professional analysis, software engineering, scientific discovery, multimodal reasoning, and long-horizon agentic tool use (Shi et al., 2025; Li et al., 2025; Wang et al., 2025c; Yang et al., 2026b; Ma et al., 2024; Liu et al., 2026c; Dong et al., 2025a,b). In these tasks,

the model is no longer only required to produce a short and verifiable answer. Instead, it must understand complex user intents, search for information automatically, reason over multiple steps, interact with external tools or environments, and generate outputs that satisfy diverse task-specific requirements.

Challenges This transformation in the task landscape makes both evaluation and model training more difficult. For many complex and open-ended tasks (Shi et al., 2025; Arora et al., 2025), there is no ground-truth answer, and response quality cannot be reliably assessed only by signals such as exact match, execution success, or binary correctness. In these tasks, high-quality responses need to satisfy task-specific requirements and constraints. For example, in research report generation (Du et al., 2025), a model-generated report should comprehensively cover the user question, synthesize evidence from reliable sources, maintain factual consistency, and present the findings in a clear and well-structured manner. In agentic tool-use tasks (Lü et al., 2026), evaluation goes beyond the final answer: the intermediate trajectory should also satisfy requirements such as subgoal completion, appropriate tool selection, and execution correctness. **These requirements create challenges not only for evaluating model outputs, but also for constructing reliable supervision signals for model optimization.** Existing reward modeling approaches provide important foundations, but they remain limited in these complex and open-ended tasks. For example, neural reward models (Zhong et al., 2025) offer scalable scalar feedback, yet they lack interpretability and generalizability in these tasks. Reinforcement learning with verifiable rewards (RLVR) (Lambert et al., 2024; DeepSeek-AI, 2025) has achieved strong progress in domains such as mathematics and code generation; however, it is limited to tasks whose answer correctness is verifiable and does not apply to open-ended tasks. LLM-as-a-judge (Gu et al., 2024) provides a more flexible alternative, but unconstrained judging prompts may lead to unstable, unreliable, or biased assessments. These limitations point to a common missing component: **structured and explicit evaluation criteria that specify the requirements a response should satisfy**.

Why Rubrics Matter *Rubrics* provide a natural way to fill this gap. Originating from educational assessment (Brookhart, 2018; Panadero and Jonsson, 2013), **rubrics decompose quality evaluation into a set of explicit criteria, each corresponding to a concrete aspect of the desired output**. The lower part of Figure 1 shows concrete examples of rubrics in various tasks. In the context of LLMs, rubrics are closely related to checklists, evaluation dimensions, criteria, and grading guidelines, but they emphasize a more structured representation of quality standards. Rather than asking a judge or reward model to produce a single holistic score, a rubric-based evaluation assesses model outputs along multiple dimensions and then aggregates these fine-grained judgments into an overall assessment. This structure is especially useful for open-ended tasks, where different responses may succeed or fail in different ways. By making evaluation standards explicit, rubrics improve the transparency, controllability, and diagnostic value of model assessment, while also offering a practical interface for transforming human preferences and task requirements into supervision signals of LLM training.

In recent years, rubrics have received increasing attention in the LLM community and have been applied to a broad range of complex tasks. Rubric-based methods have shown strong potential in evaluating and improving model capabilities in scenarios such as deep research report generation (Du et al., 2025), medical question answering (Arora et al., 2025), professional reasoning (Wang et al., 2025c), multimodal generation (Ni et al., 2026), and agentic task solving (Ma et al., 2024). Meanwhile, the target trained models have also evolved from instruction-following LLMs (Viswanathan et al., 2025; Jia et al., 2026) to reasoning LLMs (Jia et al., 2025; Sheng et al., 2026) and tool-using AI agents (Shao et al., 2025; Lü et al., 2026). Given the growing significance of rubrics and the rapid expansion of related studies, a systematic and up-to-date survey is needed to summarize this emerging research area, which motivates this survey. In this survey, we extensively review recent studies on rubrics and organize them through systematic classification and summarization, as shown in the upper part of Figure 1. Our goal is to help readers better understand what rubrics are and how they can be used for model training and evaluation. Specifically, this survey aims to address the following key questions:

Key Questions

❶ **Rubrics Definition:** What are *rubrics*, and how do they relate to relevant concepts such as reward models, verifiable rewards, and LLM-as-a-judge?

❷ **Rubrics Construction:** How are rubrics constructed for different tasks, domains, and model behaviors?

❸ **Model Training:** How are rubrics used to provide supervision signals for policy model training and reward model training?

❹ **Evaluation:** How do rubrics support the evaluation of LLMs on different tasks?

❺ **Open Questions:** What are the key challenges and promising frontiers for future research?

To address question ❶, we first define rubrics in Section 2 and compare them with related concepts, including reward models, verifiable rewards, and LLM-as-a-judge. Question ❷ is discussed in Section 3, where we review four major approaches to rubric construction: direct generation, contrastive generation, iterative refinement, and online or co-evolving generation. Question ❸ is addressed in Section 4, where we discuss how rubrics are used to train policy models and reward models based on different training mechanisms. Question ❹ is covered in Section 5, where we summarize rubric-based evaluation for both general and domain-specific tasks. The papers discussed in Sections 3–5 are illustrated in Figure 2. Finally, question ❺ is discussed in Section 6, where we identify open challenges and promising directions for future rubric-based research.

Contributions The contributions of this survey can be summarized as follows:

- **First comprehensive survey on rubrics for LLMs.** To the best of our knowledge, this is the first survey that comprehensively reviews rubric-related research for LLMs. By synthesizing recent advances in this emerging area, this survey provides readers with a deep understanding of what rubrics are, how they are constructed, and how they are used for model training and evaluation.
- **Systematic taxonomy and in-depth analysis of rubric construction.** We systematically summarize existing methods for rubric construction and organize them into four major categories: direct generation, contrastive generation, iterative refinement, and online or co-evolving generation. For each category, we provide an in-depth analysis of its input signals, generation mechanisms, refinement strategies, and representative applications.
- **Comprehensive review of rubrics for model training and evaluation.** We extensively review how rubrics are applied to model training and evaluation. For model training, we categorize existing methods according to their roles in policy model training and reward model training based on different training mechanisms. For evaluation, we introduce rubric-based evaluation benchmarks for general tasks and domain-specific tasks.
- **Discussion of open challenges and future directions.** We provide an in-depth discussion of key open challenges in rubric-based research. These discussions highlight the current limitations of rubric-based methods and outline promising directions for future research.

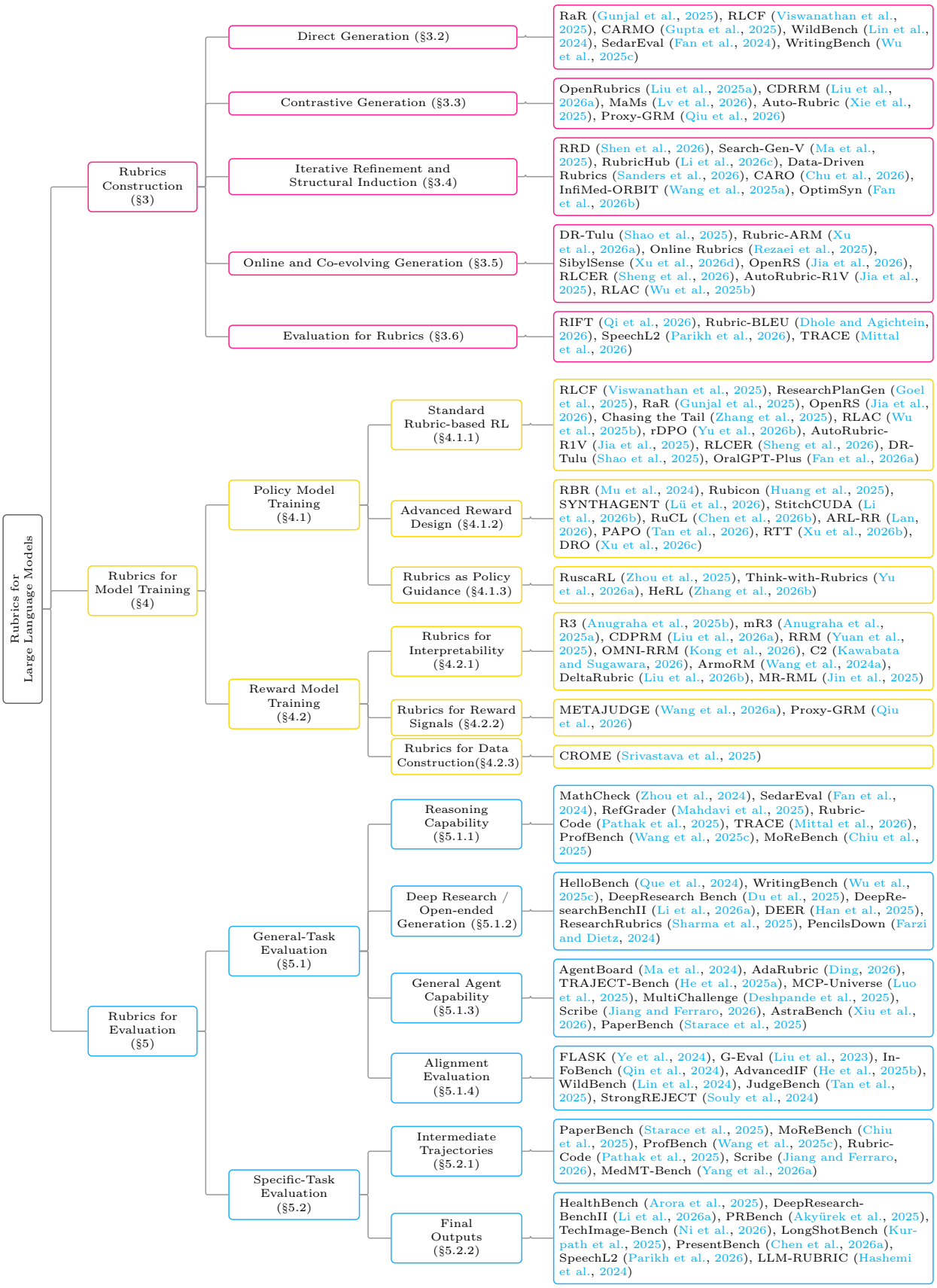

Figure 2 A comprehensive taxonomy of rubric-based methods for large language models. The taxonomy organizes existing studies into rubric construction, rubric-based model training, and rubric-driven evaluation.

2 Preliminaries: Formalizing Rubrics

2.1 Reward Modeling in Reinforcement Learning

Reinforcement learning (RL) has become a central paradigm for aligning LLMs with desired behaviors. During the RL process, reward modeling is a key component that defines how model outputs are evaluated and optimized. Given an input and a generated response, the reward function measures the quality of the response and provides feedback to guide policy updates. Therefore, the design of the reward signal plays a critical role in determining what behaviors the model ultimately learns.

Reward Models and Verifiable Rewards. Existing approaches construct reward signals in different ways. One common approach is to train neural *reward models* (Zhong et al., 2025) that approximate human preferences from annotated data, typically derived from pairwise comparisons. These models assign scalar scores to candidate responses and are widely used to guide policy optimization. Another line of work relies on Reinforcement Learning from *Verifiable Rewards* (RLVR) (Lambert et al., 2024; DeepSeek-AI, 2025), where reward signals are obtained through objective verification procedures. This paradigm leverages tasks for which solutions may be challenging to generate but can be reliably checked, such as mathematical problem solving or code generation, where correctness can be determined by matching ground-truth answers or executing test cases.

However, both approaches face limitations when applied to open-ended tasks such as deep research report generation (Shi et al., 2025; Li et al., 2025) whose responses are very long and difficult to verify. Reward models often provide coarse scalar rewards, which fail to capture the diverse aspects of open-ended responses and offer limited interpretability. In addition, they require enough annotated preference data for training and are prone to reward hacking (Guo et al., 2025). RLVR, on the other hand, is inherently limited to tasks with verifiable outcomes and does not readily extend to open-ended and non-verifiable tasks that lack explicit ground truth and require multi-dimensional evaluation.

LLM-as-a-Judge. More recently, LLM-as-a-judge (Gu et al., 2024; Li et al., 2024) has emerged as a flexible alternative, where large language models are prompted to directly evaluate the responses. While this approach reduces the need for explicit reward modeling and improves adaptability across tasks, it relies on underspecified judging prompts, where the underlying evaluation criteria are implicitly encoded in the model rather than explicitly specified. As a result, it often produces unreliable and inconsistent judgments (Wang et al., 2024b).

Limitations of Existing Reward Modeling. Despite their differences, existing reward modeling approaches share several fundamental limitations. First, these approaches typically produce a single scalar reward without explicitly modeling multiple evaluation dimensions (such as factual correctness, coverage, and credibility), making them difficult to generalize to open-ended tasks or non-verifiable domains. Second, the resulting reward signals are often not interpretable, as evaluation criteria are implicitly encoded in model parameters or prompts, making it difficult to understand or control what aspects of the response are being optimized. These limitations highlight a fundamental gap in existing reward modeling: **the lack of structured and explicit evaluation criteria**. To address this, it is desirable to decompose response evaluation into multiple explicit dimensions. This motivates the use of rubrics, which provide a structured framework that evaluates the response along different dimensions.

2.2 What Are Rubrics?

Definition. In educational assessment, a rubric is a scoring guide. It tells evaluators what aspects of an answer should be assessed, and what stronger or weaker performance looks like on each aspect. In this way, evaluation standards that might otherwise remain implicit are made explicit, easier to communicate, and more consistent across scoring and feedback (Brookhart, 2018; Panadero and Jonsson, 2013). With the rise of LLMs, rubrics have received renewed attention. In the LLM setting, they are also often referred to as *checklists*. In general, the idea is simple: a reviewer (either a human or an LLM judge) examines an output item by item according to a set of evaluation points, and then combines these judgments into an overall score or a structured report (Viswanathan et al., 2025).

For the sake of unifying later discussion across both evaluation and training, we define *a rubric* as a set of rubric items:

$$\mathcal{R} = \{(d_j,\ w_j)\}_{j=1}^{k}, \tag{1}$$

where d_j is the natural-language description of the j-th rubric item, and $w_j \in \mathbb{R}$ denotes its weight or importance. Let x denote the input or task, and let y denote the model output. For each rubric item, the judge assigns a score

$$c_j(x, y) \in [0, 1], \tag{2}$$

which indicates how well the output satisfies that item.

A simple aggregated rubric score can then be written as

$$S_{\mathcal{R}}(x, y) = \frac{\sum_{j=1}^{k} w_j\, c_j(x, y)}{\sum_{j=1}^{k} w_j}. \tag{3}$$

This normalized form is convenient because it remains comparable across tasks with different numbers of rubric items or different weighting schemes.

This item-by-item scoring scheme is essentially the same structure used when rubrics are turned into reward signals in reinforcement learning (RL) and model evaluation. In RL training, the rubric becomes part of the optimization objective (Rao and Callison-Burch, 2026; Gunjal et al., 2025), while in evaluation, it serves as a scoring function (Arora et al., 2025). More broadly, rubrics can be divided into two common types. A *holistic* rubric assigns one overall score to the output, whereas an *analytic* rubric breaks evaluation into multiple rubric items and scores them separately. For open-ended LLM tasks, analytic rubrics are often more useful. They make it easier to identify which aspect of an output fails, and they provide more fine-grained signals for both evaluation and training (Rao and Callison-Burch, 2026; Huang et al., 2025).

Weighting and Aggregation Methods. Once multiple rubric items are scored, the next question is how to combine them into a single score. In practice, the most common aggregation strategies can be grouped into three simple types.

The first is *direct averaging or summation*. In this setting, all rubric items are treated equally, and the final score is obtained by directly summing or averaging the item-level scores:

$$S_{\text{avg}}(x, y) = \frac{1}{k} \sum_{j=1}^{k} c_j(x, y), \tag{4}$$

or equivalently by using the unnormalized sum

$$S_{\text{sum}}(x, y) = \sum_{j=1}^{k} c_j(x, y). \tag{5}$$

This is the simplest and most transparent approach. It is easy to implement and interpret, and is often used when all rubric items are considered equally important.

The second is *weighted summation*. Here, different rubric items are assigned different levels of importance, and the final score is computed as a weighted combination:

$$S_{\mathcal{R}}(x, y) = \frac{\sum_{j=1}^{k} w_j\, c_j(x, y)}{\sum_{j=1}^{k} w_j}. \tag{6}$$

Compared with direct averaging, weighted summation is more flexible because it allows critical dimensions to contribute more to the final score. This is especially useful when some rubric items, such as safety or task completion, should be emphasized more than others.

The third is *implicit aggregation*. Instead of first scoring each rubric item separately and then combining them, one may provide the full rubric and the model output to a judge model, and directly ask it to produce a single overall score:

$$S_{\text{imp}}(x, y) = f_\phi(x, y, \mathcal{R}), \tag{7}$$

where f_ϕ denotes the judge model. This approach is often simpler at inference time and may reduce evaluation cost. However, it is less transparent, because the contribution of each rubric item is no longer explicitly observable.

In this paper, we mainly focus on explicit aggregation, including direct averaging and weighted summation, because these forms make the evaluation process easier to inspect and analyze. Implicit aggregation is still important as a practical alternative, especially when efficiency is a primary concern.

2.3 Comparing Rubrics with Other Key Concepts

Rubrics are often discussed together with other related concepts such as LLM-as-a-judge, reward models, and RLVR. However, these concepts operate at different levels. Rubrics mainly specify *what standards an output should be judged by*. The other concepts are more about *who performs the judgment* or *how that judgment is turned into a training signal*. For this reason, rubrics are best understood as an explicit evaluation framework rather than a specific model or learning algorithm.

2.3.1 Rubrics vs. LLM-as-a-Judge

The difference between rubrics and *LLM-as-a-judge* is the difference between the *evaluation standard* and the *evaluator*. LLM-as-a-judge refers to using an LLM to score, compare, or judge outputs (Gu et al., 2025; Zheng et al., 2023). A rubric, by contrast, specifies what should be judged: for example, factuality, completeness, style, or safety. An LLM judge may give a single overall score without using a rubric, or it may follow a rubric and score the output item by item. The latter is usually more informative because it shows not only the final score but also where the output succeeds or fails.

2.3.2 Rubrics vs. Reward Models

Rubrics also differ from *reward models*. A reward model usually outputs a scalar score directly, with the notion of quality implicitly encoded in its parameters. A rubric makes the scoring standard explicit by listing the rubric items and evaluating them one by one. In this sense, a reward model is more like a black-box scorer, while a rubric is more transparent and easier to inspect or edit (Huang et al., 2025). Researchers can revise rubric items, adjust their weights, and analyze which dimensions a model performs poorly on. At the same time, the two are not mutually exclusive: rubrics can be used to supervise or construct reward models, which will be discussed in Section 4.2.

2.3.3 Rubrics vs. RLVR

Rubrics relate differently to *RLVR* (reinforcement learning with verifiable rewards). RLVR is designed for settings where correctness can be checked automatically, such as mathematical problem solving or code generation (Wang et al., 2025b). These tasks often have a clear target answer, so reward assignment is straightforward. Many open-ended tasks, however, do not have a single correct answer. Writing, dialogue, planning, and style control are typical examples. In such cases, rubrics are useful because they provide structured reward signals through multiple explicit rubric items, instead of relying on one fully verifiable target.

2.3.4 Summary

These concepts can therefore be distinguished in a simple way: rubrics specify *what standards to evaluate against*; LLM-as-a-judge specifies *who performs the evaluation*; reward models specify *how to output a score directly*; and RLVR specifies *how rewards can be assigned through automatic verification*. The value of rubrics is not that they replace these concepts, but that they provide a clear and interpretable intermediate layer that can support both evaluation and training.

Table 1 Summary of representative rubric construction methods. Methods are organized by generation paradigm, with columns indicating the conditioning signal, quality control mechanism, output form, and whether the method supports online adaptation.

Method	Paradigm	Conditioning Signal	Quality Control	Online	Key Contribution
RaR (Gunjal et al., 2025)	Direct	Query + answer	None	✗	Instance-specific binary rubrics as RL rewards
RLCF (Viswanathan et al., 2025)	Direct	Query only	None	✗	Instruction-derived checklists outperform reward models
CARMO (Gupta et al., 2025)	Direct	Query + answer	None	✗	Context-aware criteria for reward modeling
OpenRubrics (Liu et al., 2025a)	Contrastive	Preference pair	Discriminative filter	✗	Contrastive rubric generation with consistency filtering
CDRRM (Liu et al., 2026a)	Contrastive	Preference pair	Contrast profiling	✗	Contrast-then-synthesis two-stage pipeline
MaMs (Lv et al., 2026)	Contrastive	Human preferences	Generator learning	✗	Trained rubric generator from preference data
Auto-Rubric (Xie et al., 2025)	Contrastive + Refine	Preference pair	Verify + compress	✗	End-to-end: generate, verify, compress to Theme-Tips
RRD (Shen et al., 2026)	Refinement	Query + answer	Decompose-filter cycle	✗	Recursive decomposition addressing four failure modes
Search-Gen-V (Ma et al., 2025)	Refinement	Query + evidence	Nugget verification	✗	Nugget-as-rubric for maximal verifiability
RubricHub (Li et al., 2026c)	Refinement	Query + answer	Coarse-to-fine pipeline	✗	Large-scale rubric dataset with progressive difficulty
Data-Driven (Sanders et al., 2026)	Structural	Failure trajectories	Error taxonomy	✗	Rubrics from failure mode classification
CARO (Chu et al., 2026)	Structural	Rubric set	Separability optim.	✗	Confusion-aware inter-dimension optimization
InfiMed-ORBIT (Wang et al., 2025a)	Refinement	Query + answer	Coarse-to-fine rubrics	✗	Domain-specific rubric-guided incremental RL
OptimSyn (Fan et al., 2026b)	Refinement	Training utility	Influence-guided optim.	✗	Rubric optimization via influence estimation
DR Tulu (Shao et al., 2025)	Online	Rollout history	Variance filtering	✓	Rollout-based rubric evolution with buffer
Rubric-ARM (Xu et al., 2026a)	Online	Policy + judge	Alternating RL	✓	Rubric generation as latent action selection
Online Rubrics (Rezaei et al., 2025)	Online	Streaming pairs	Online decision	✓	Rubric elicitation as online optimization
SibylSense (Xu et al., 2026d)	Online	Memory + probes	Adversarial probing	✓	Memory-augmented adversarial rubric discovery
RLAC (Wu et al., 2025b)	Online	Critic interaction	Adversarial critic	✓	Failure-focused rubrics via adversarial critics

3 Rubrics Construction

The quality of rubrics is the foundation of their downstream applications including model training and task evaluation, which will be discussed in Section 4 and Section 5. Logically prior to both applications, this section focuses on the following question: **how are rubrics themselves constructed?** We review methods that explicitly propose, decompose, filter, compress, or dynamically update rubrics. A summary of representative methods is provided in Table 1.

We organize existing methods into four paradigms (as illustrated in Figure 3) that form a clear evolutionary trajectory: *direct generation* (Section 3.2) → *contrastive generation* (Section 3.3) → *iterative refinement* (Section 3.4) → *online and co-evolving generation* (Section 3.5). This trajectory is driven by progress along two coupled dimensions: the **conditioning signal** fed to the generator (from a bare query, to preference pairs, to full rollout histories) and the **quality control mechanism** applied after generation (from none, to verification and structural decomposition, to online co-evolution with the policy). Each subsequent paradigm enriches the conditioning signal, strengthens quality control, or both. For example, contrastive generation upgrades the signal from a single response to preference pairs; iterative refinement adds verification and decomposition on top of either signal; and online generation further extends the signal to rollout trajectories while making quality control continuous. Before discussing these four paradigms, we introduce a unified analytical framework (Section 3.1) that makes the comparison concrete.

3.1 Definition of Rubrics Construction

The rubric construction process can be formalized as follows: given an input context $\mathcal{C}$ (which may include a query q, one or more candidate responses, supporting evidence, or response preference pairs), a rubric generator $\mathcal{G}$ produces a set of evaluation criteria $\mathcal{R} = \mathcal{G}(\mathcal{C})$, where each criterion $r_k \in \mathcal{R}$ specifies a concrete and assessable quality dimension. To compare methods on equal footing, we characterize each method along three dimensions: what it takes as input (*conditioning signal*), what it produces (*output form*), and what properties the produced rubrics are expected to satisfy (*quality requirements*).

Conditioning Signal. The richness of the input context $\mathcal{C}$ fundamentally determines which types of rubrics can be generated. The types of input context $\mathcal{C}$ can be roughly divided into the following four categories:

- **Query only** (q): The rubrics are generated solely from the query along with the task description.
- **Query + answer** (q, y) or **query + evidence** $(q, \mathcal{E})$: rubrics are generated based on a high-quality answer or supporting evidence.
- **Query + preference pair** (q, y^+, y^-): rubrics are derived by contrasting high-quality and low-quality responses, thereby highlighting discriminative evaluation dimensions.
- **Policy rollouts during training**: rubrics are derived from, or updated against, the trajectories produced by the policy itself, so that the criteria reflect behaviors actually exhibited during training.

Output Form. Existing methods produce rubrics in several common forms: *binary atomic rubrics* where each criterion gives a yes/no judgment (Gunjal et al., 2025; Viswanathan et al., 2025); *hierarchical rubrics* where coarse dimensions are split into finer sub-criteria (Shen et al., 2026; Li et al., 2026c); *positive and negative rubrics* that separately describe desired qualities and typical failure modes (Shao et al., 2025; Huang et al., 2025); *nugget-as-rubric*, where atomic facts extracted from evidence are used directly as rubric items (Ma et al., 2025); and *query-agnostic themes or meta-rubrics* that apply across instances rather than to a single query (Xie et al., 2025; Jia et al., 2026).

Quality Requirements. A high-quality rubric set must simultaneously satisfy several requirements: **discriminativeness** demands that rubrics reliably distinguish high-quality from low-quality responses; **coverage** requires rubrics to span all critical evaluation axes of a task; **atomicity and verifiability** ensure each criterion is sufficiently fine-grained to be unambiguously assessed; **alignment** requires rubric directions to be consistent with true quality preferences; **redundancy and correlation control** prevents overlapping criteria from causing duplicated scoring and noise amplification; and **dynamic robustness** requires rubrics to remain effective as the evaluated policy model continues to evolve.

Within this framework, improvements in subsequent methods can be understood as addressing three shortcomings of *naive direct generation*: *insufficient conditioning signals*: the input does not contain enough information for the generator to identify what really matters; *lack of quality control*: no mechanism verifies whether the produced rubrics are discriminative, atomic, or non-redundant; *static assumptions*: the rubric set is fixed once and for all, even though the policy or task distribution it scores keeps changing. Early works that introduced rubric-like evaluation structures (Ye et al., 2024; Kim et al., 2024) demonstrated the value of explicit evaluation dimensions but did not systematically investigate how rubrics themselves should be generated. More recent work has started to treat rubric generation as a standalone research problem, with growing attention to its underlying conditioning signals, quality control mechanisms, and structural design.

3.2 Direct Generation

The most fundamental paradigm for rubric construction is **direct generation**: given a query q (optionally accompanied by a candidate answer y or supporting evidence $\mathcal{E}$), a strong LLM produces a set of evaluation rubrics in a single prompt. This paradigm is enabled by the strong instruction-following and abstract summarization capabilities of modern LLMs, which can infer appropriate evaluation dimensions from a task description and a candidate response. Direct generation offers three key advantages: it requires no preference pairs or pre-trained rubric generators, making cold-start deployment straightforward; it scales readily to new tasks and domains; and it naturally produces instance-specific evaluation criteria tailored to each query.

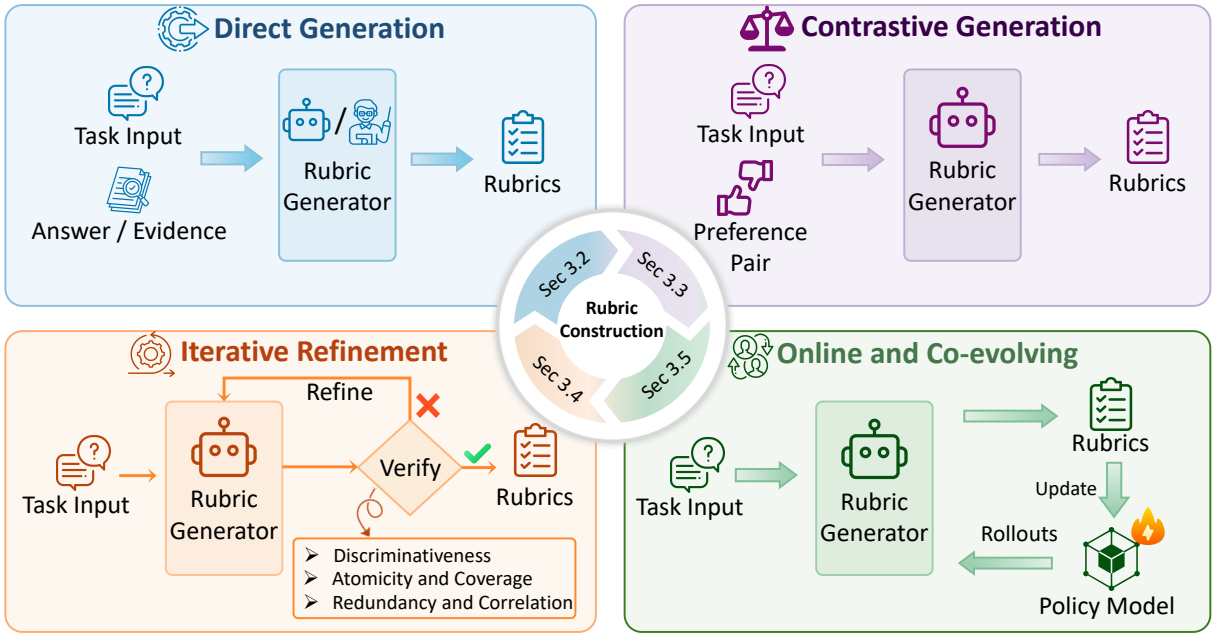

Figure 3 Overview of four different paradigms for constructing rubrics.

RaR (Gunjal et al., 2025) is a representative work in this paradigm. Given a query and candidate answer, it prompts an LLM to generate multiple self-contained binary rubrics in a single pass, each covering an independent quality dimension with an associated importance weight, and subsequently aggregates rubric scores into a scalar reward for RL training. The key contribution of this work is the formal establishment of instance-specific rubrics as an explicit research object. RLCF (Viswanathan et al., 2025) takes a complementary approach by extracting checklists directly from instructions, demonstrating that effective rubric generation does not require pairwise preferences. A well-designed prompt over the instruction alone can produce evaluation criteria superior to monolithic reward models. CARMO (Gupta et al., 2025) extends the direct generation paradigm by dynamically generating context-aware criteria for each query to serve downstream reward models, indicating that even within the one-shot setting, context adaptability has become a recognized concern.

Beyond methods that generate rubrics for policy model training, several evaluation benchmarks also adopt the direct generation approach to construct task-specific rubrics. WildBench (Lin et al., 2024) employs task-specific checklists for evaluation on real user tasks. SedarEval (Fan et al., 2024) proposes self-adaptive rubrics that match each question with structured scoring and deduction rules, representing an early effort in automated question-specific rubric construction. WritingBench (Wu et al., 2025c) derives evaluation criteria directly from writing instructions. HealthBench (Arora et al., 2025) and PaperBench (Starace et al., 2025) rely on human experts to write rubrics: the former uses physician-authored rubrics for medical conversations, while the latter constructs expert-designed hierarchical rubrics for evaluating research replication.

Despite its simplicity and scalability, direct generation exhibits several structural shortcomings that motivate more sophisticated approaches. Coverage depends entirely on what the LLM can enumerate in a single call, with no mechanism for detecting omissions or filling gaps, and the granularity of generated criteria is often inconsistent: some are overly abstract (e.g., "the response should be helpful") while others are excessively specific. Most critically, generating rubrics from a single response provides no **contrastive signal**: without simultaneously observing both high-quality and low-quality responses, the model cannot reliably tell whether a rubric genuinely discriminates between response qualities, leading to rubrics that are correct but vacuous. These observations suggest a simple solution: conditioning rubric generation on explicit preference pairs so that the generator can identify the aspects that distinguish good responses from bad ones.

3.3 Contrastive Generation

Relying solely on task instructions to generate evaluation rubrics makes it difficult to identify which dimensions are truly critical for distinguishing response quality. An intuitive solution is to leverage human-annotated preference pairs. Such data is relatively easy to obtain because many deployed applications already provide thumbs-up/down buttons or side-by-side comparison interfaces that continuously collect user preferences at scale, yielding a rich and naturally growing source of contrastive supervision. When the input is extended from (q, y) to (q, y^+, y^-), namely a query paired with chosen and rejected responses, the generator can directly target the question *"why is response A better than response B?"* to extract evaluation dimensions. Pairwise preferences thus provide a natural discriminative signal for rubric generation, fundamentally addressing the discriminativeness deficit identified in direct generation.

Existing contrastive methods vary widely in how they exploit preference signals, from direct prompting to learnable generation models. At the most basic level, **direct contrastive prompting** feeds a preference pair (q, y^+, y^-) to an LLM and asks it to enumerate the dimensions on which y^+ outperforms y^-, e.g., factual correctness, depth of reasoning, or instruction adherence; these dimensions are then assembled into a rubric set. CDRRM (Liu et al., 2026a) extends this idea with a two-stage "Contrast-then-Synthesis" pipeline: it first analyzes the main differences between the two responses across multiple aspects, and then summarizes these differences into clearer and more reusable rubrics. This shows that effective rubric construction requires more than simply listing observed differences.

However, generating rubrics in a single call from one preference pair has a clear weakness: the produced rubrics are often **tied to the specific differences of that pair** and do not generalize to other instances of the same task. Moreover, in the high-reward tail region where quality gaps between responses become subtle, directly generated rubrics often lack sufficient discriminative power (Zhang et al., 2025). These limitations motivate a second family of methods that adds **discriminative filtering** as a post-generation quality gate: candidate rubrics are first generated and then evaluated on held-out preference pairs, retaining only those criteria that demonstrably distinguish chosen from rejected responses. OpenRubrics (Liu et al., 2025a) synthesizes rubrics at the per-criterion level from (q, y^+, y^-) and then filters them by checking that the kept rubrics give consistent and discriminative scores on held-out pairs. Proxy-GRM (Qiu et al., 2026) closes the loop further: a proxy model uses the generated rubrics to predict preferences on held-out pairs, and prediction accuracy is fed back as a signal to improve the rubric generator.

To improve the generalizability of rubric generation beyond individual instances, a third line of work adopts **generator learning**, which distills the contrastive rubric generation process into dedicated models or aggregates instance-level results into query-agnostic general principles. For example, MaMs (Lv et al., 2026) and Auto-Rubric (Xie et al., 2025) train a specialized rubric generator using preference pairs. This shift from prompt-based extraction to learnable generators is important because it allows rubric generation patterns learned from existing data to be reused across tasks and domains, rather than generating rubrics based on a general-purpose LLM for each new instance.

Contrastive generation thus equips rubric construction with its first principled source of discriminative signal, but two issues remain unsolved and motivate the next paradigm. First, pairwise signals emphasize *what differs* rather than *what matters*: the most salient dimension of difference between two responses may not correspond to the most critical evaluation axis of the task. Second, without additional verification, decomposition, and consolidation, the resulting rubrics tend to remain pair-specific rather than forming a coherent, systematic evaluation structure for the task. The next subsection therefore turns to methods that explicitly target these two issues by treating rubric construction as an iterative refinement problem.

3.4 Iterative Refinement

Although direct and contrastive generation provide scalable ways to produce rubrics, the generated criteria are often imperfect in structure and reliability. They may be too coarse-grained to support stable judgment, fail to cover important evaluation aspects, overlap with one another, or remain insufficiently verifiable. Therefore, a line of work treats rubric construction not as a one-shot generation problem, but as an iterative refinement process that improves the quality of an initially generated rubric set before it is used for evaluation or training. These observations have motivated a body of work that focuses on *iteratively improving rubric quality* with

respect to the desiderata introduced in Section 3.1.

Concretely, existing efforts are organized around three complementary quality dimensions: (1) *discriminativeness*: whether individual rubric items genuinely distinguish response quality; (2) *atomicity and coverage*: whether rubrics are sufficiently fine-grained and span all critical evaluation axes; and (3) *redundancy and correlation control*: whether the rubric set is compact, non-redundant, and transferable. The following three subsubsections address each dimension in turn.

3.4.1 Verification-Driven Refinement

A growing consensus in the community holds that rubric quality should not be judged solely by whether rubrics *look like* evaluation criteria; rather, the critical test is whether they can reliably discriminate between high-quality and low-quality responses in actual comparisons. This recognition has given rise to a propose-evaluate-revise paradigm: candidate rubrics are first generated, then tested by a verifier or judge for discriminative power, and those that fail are iteratively revised.

For example, Auto-Rubric (Xie et al., 2025) instantiates this loop by testing each candidate rubric on held-out preference pairs to check whether its scores systematically favor chosen over rejected responses; rubrics that fail this check are sent back to the generator and revised, and the generate-verify-revise cycle repeats until the rubric set reaches a target discriminative threshold. (Its downstream compression of the verified rubrics into transferable "Theme-Tips" will be discussed separately in Section 3.4.3.) OptimSyn (Fan et al., 2026b) pushes verification one step further by replacing preference-based checks with *downstream training utility*: it uses an influence-estimation procedure to quantify each synthetic sample's contribution to the target model's objective, and treats this signal as the RL reward for a rubric generator, showing that rubric quality can be grounded in measurable training impact rather than surface-level preference alignment.

The significance of this paradigm is that rubric generation is no longer a one-shot text production task, but rather a generate-and-verify loop with quality control embedded in the construction process itself.

3.4.2 Structural Decomposition

Even when a rubric is discriminative, it can still be unreliable if a single item mixes several quality dimensions together. Coarse-grained rubrics make judge scores less stable and hide important quality differences under overly broad labels. This motivates **structural decomposition**: recursively splitting coarse rubrics into finer-grained, atomic sub-criteria.

RRD (Shen et al., 2026) provides the most systematic treatment of this problem. It identifies four fundamental failure modes of naive rubric generation including insufficient coverage, dimension conflation, directional misalignment, and inter-criterion redundancy. Furthermore, they propose a recursive decompose-filter cycle that alternates between expanding rubrics into fine-grained sub-criteria and filtering out redundant or misaligned items. Qworld (Gao et al., 2026) solves the problem from another angle: performing hierarchical criteria expansion around the implicit evaluation axes of each question, treating high-coverage criterion generation as a systematic exploration of the question space. Search-Gen-V (Ma et al., 2025) pushes verifiability to the extreme with its nugget-as-rubric approach: instead of expressing rubrics in abstract language, it converts atomic information points (nuggets) from retrieved evidence directly into rubric items, ensuring each can be directly verified. RubricHub (Li et al., 2026c) implements an automated coarse-to-fine generation pipeline that progressively refines principles into detailed rubrics with increasing discriminative difficulty, producing a large-scale high-quality rubric dataset.

Several works extend structural decomposition to specific domains: Pathak et al. (2025) introduces the question-specific rubric generation for code evaluation; DeepResearch Bench II (Li et al., 2026a) employs a hybrid pipeline combining LLM extraction with expert revision to derive fine-grained rubrics from expert reports; RefGrader (Mahdavi et al., 2025) automatically derives problem-specific rubrics for mathematical proof scenarios from reference solutions and error analyses; RubricRAG (Dhole and Agichtein, 2026) introduces retrieval-augmented rubric generation, anchoring criteria in domain knowledge; and InfiMed-ORBIT (Wang et al., 2025a) converts vague open-ended evaluation standards into verifiable, multi-dimensional fine-grained rubrics for medical consultation and uses them as RL rewards, demonstrating the value of rubric decomposition in high-stakes domains.

3.4.3 De-duplication and Compression

After large-scale rubric generation, redundancy is inevitable: multiple rubrics may describe the same underlying quality dimension in different words, causing duplicated scoring and noise amplification during aggregation. More fundamentally, rubric generation should not be limited to writing criteria for individual instances, it should also discover transferable evaluation structures from local criteria.

Auto-Rubric (Xie et al., 2025) addresses this through its query-agnostic compression stage: clustering and distilling verified instance-specific rubrics into reusable "Theme-Tips". This pipeline embodies the dual objective of rubric generation—producing per-instance evaluation standards while simultaneously learning transferable structures from local criteria. RRD (Shen et al., 2026) supplements decomposition with relevance weighting and correlation-aware filtering to control redundancy in expanded criterion sets.

An alternative structural induction approach is represented by Sanders et al. (2026). Rather than constructing rubrics from positive quality dimensions, this work induces **error taxonomies** from reasoning failure trajectories, demonstrating that rubrics can equally be constituted by systematic classification of frequent failure modes. CARO (Chu et al., 2026) targets a more subtle quality issue: even when rubric sets are free of redundancy, blurred boundaries between dimensions can cause judge confusion. By explicitly optimizing inter-dimension separability, this work demonstrates that high-quality rubric sets require not only comprehensive coverage and fine granularity, but also **clear inter-dimension boundaries**.

Taken together, these verification, decomposition, and compression techniques suggest that rubric generation is no longer treated as simple text generation, but increasingly as a problem of structural design and refinement. However, these methods still view rubric construction as an *offline* process: rubrics are created before policy model training and then kept fixed throughout. As the policy improves, some rubric criteria may gradually lose their ability to distinguish strong outputs from weak ones, and a fixed set of rubrics can be more easily exploited through reward hacking. This limitation points to the need for rubrics that can adapt over the policy training.

3.5 Online and Co-evolving Generation

All methods discussed above are *offline*: rubrics are constructed before training begins and do not change afterwards. The methods reviewed in this subsection drop this assumption and let rubrics **co-evolve with the policy or its rollout distribution**. We organize them by *how the update is performed*, which gives three increasingly autonomous categories. (i) **Rollout-based heuristic updating** (Section 3.5.1) keeps a fixed update rule and refreshes the rubric pool from the latest trajectories; the rubric generator itself is not trained. (ii) **Online and alternating optimization** (Section 3.5.2) makes the rubric generator a *trainable* component that is updated jointly with the policy or judge through streaming preferences or alternating RL. (iii) **Self-evolving, adversarial, and memory-driven** methods (Section 3.5.3) go one step further by actively probing for blind spots and proposing new rubrics for failures the system has not yet covered. The boundary between (i) and (iii) is admittedly soft—a method like DR-Tulu sits closer to (i) because its update rule is fixed—but this axis (*who decides what to add to the rubric set*: a fixed heuristic, a learned generator, or an adversarial discoverer) gives the cleanest separation we have found.

Online rubric generation is motivated by three main problems. **First**, static rubrics may become less useful as the policy improves. Criteria that were once able to distinguish strong responses from weak ones may eventually be satisfied by most candidates, making the reward signal weak. **Second**, fixed rubrics may lead to reward hacking. The policy may learn to match the superficial features of the rubrics without genuinely improving response quality. **Third**, in long-horizon agentic tasks, some failure modes only appear during policy exploration and are difficult to predict before training.

3.5.1 Rollout-Based Evolving Rubrics

DR-Tulu (Shao et al., 2025) is a foundational work in this direction. It initializes rubrics from queries and retrieved evidence, then regenerates and updates rubrics after each training round based on the latest policy rollouts. Specifically, it generates positive and negative rubrics based on the most recent trajectories, maintains a rubric buffer, and applies variance-based filtering to retain only those criteria that remain discriminative under the current policy distribution. This design reveals a key insight: for long-horizon agentic tasks, **rubric**

generation must incorporate behavioral patterns newly exposed during training, lest the reward signal rapidly lose its guiding value.

3.5.2 Online and Alternating Optimization of Rubric Generators

A second line of work treats rubric updates not as heuristic steps but as **learnable components** jointly optimized with the policy or judge. Unlike the rollout-based approach above, here the rubric *generator* is itself trained, and updates are driven by streaming preferences or alternating RL rather than by replaying recent trajectories.

OnlineRubrics (Rezaei et al., 2025) continuously adjusts the rubric set in response to a stream of pairwise comparisons that arrive during training, modeling rubric selection as an online decision problem that progressively refines the set to maximize its explanatory power over the latest observed preferences—this online stream of preferences is what makes it an evolving rather than static method. Rubric-ARM (Xu et al., 2026a) formalizes rubric generation as a *latent action* within an alternating reinforcement learning framework: the rubric generator and judge are trained in alternation, so the generator's outputs change with each round of policy/judge updates rather than remaining fixed. Together these two works show that "co-evolving with training" need not rely on rollout replay; it can equally come from online preference streams or alternating optimization.

3.5.3 Self-Evolving, Adversarial, and Memory-Driven Rubrics

The most recent wave of research explores increasingly dynamic rubric adaptation mechanisms. SibylSense (Xu et al., 2026d) maintains a rubric memory bank and employs adversarial probing to discover evaluation blind spots, triggering the generation of new rubrics targeting discovered weaknesses. OpenRS (Jia et al., 2026) introduces pairwise adaptive rubrics and meta-rubrics—higher-order criteria governing how rubrics are generated and updated—creating a self-organizing rubric ecosystem. RLCER (Sheng et al., 2026) demonstrates self-evolving rubrics in chain-of-thought reasoning scenarios, iteratively refining rubrics based on their actual effectiveness in guiding reasoning policies. AutoRubric-R1V (Jia et al., 2025) distills process-level rubrics from successful reasoning trajectories in multimodal settings, demonstrating that rubrics can emerge from behavioral exemplars as well as quality contrasts. RLAC (Wu et al., 2025b) dynamically discovers failure-focused rubrics through adversarial critic interaction, positioning rubric generation as a continuous failure mode discovery mechanism.

Online and co-evolving methods thus complete the evolutionary arc traced throughout this section: from single-pass text production, through discriminative extraction and offline structural optimization, to continuous co-adaptation with the training process. The central question has shifted from "how to construct high-quality rubrics offline" to "how to continuously discover quality dimensions and failure modes not yet covered by the current rubric set". The emergence of negative rubrics, adaptive rubrics, and self-evolving rubrics reflects the community's recognition that reward hacking and distribution shift are first-class challenges in rubric construction—challenges that static methods, however carefully engineered, cannot fully address.

3.6 Evaluation for Rubrics

After rubrics are constructed, a fundamental question naturally arises: *how can the quality of rubrics themselves be quantitatively evaluated?* Since downstream model training and evaluation heavily depend on rubric quality, developing reliable evaluation frameworks for rubrics has become an increasingly important research problem. Along this line, several studies have explored this problem from different perspectives.

Qi et al. (2026) propose RIFT, a qualitative failure mode taxonomy designed to diagnose structural flaws in rubric composition. By categorizing failures into reliability, content validity, and consequential validity, they provide a principled way to identify issues such as subjectivity, non-atomicity, and missing criteria that are often conflated in downstream signals. Dhole and Agichtein (2026) investigate the relationship between intrinsic textual alignment and extrinsic utility. They introduce permutation-invariant metrics like Rubric-BLEU to quantify semantic similarity to expert standards, while demonstrating that a rubric's true quality must be grounded in its downstream effectiveness—specifically its ability to improve a judge's discriminative accuracy in preferring high-quality over low-quality responses. Parikh et al. (2026) examine the impact of construct clarity and the separation of evaluation dimensions. Their analysis highlights that

Table 2 Representative works on rubric-based policy model training. Methods are grouped by training paradigm. "Reward Level" indicates whether rubric-based rewards are assigned to the final answer only or to the full generated trajectory. "Evaluation Task" denotes the downstream task used for evaluation. "Model Type" specifies the policy class: "Instruction-Following LLM" generates direct answers only, "Reasoning LLM" generates both reasoning and answers, and "Agent" refers to tool-using LLM systems with multi-step reasoning.

Method	Reward Level	Evaluation Task	Model Type
I. Standard Rubric-based RL			
RLCF (Viswanathan et al., 2025)	Answer-level	Instruction Following	Instruction-Following LLM
ResearchPlanGen (Goel et al., 2025)	Answer-level	Research Plan Generation	Reasoning LLM
RaR (Gunjal et al., 2025)	Answer-level	Deep Research, Complex Reasoning	Instruction-Following LLM
OpenRS (Jia et al., 2026)	Answer-level	Instruction Following	Instruction-Following LLM
Chasing the Tail (Zhang et al., 2025)	Answer-level	General Instruction Following	Instruction-Following LLM
RLAC (Wu et al., 2025b)	Answer-level	Factual Text Generation, Code Generation	Instruction-Following LLM
rDPO (Yu et al., 2026b)	Answer-level	Multimodal Understanding and Reasoning	Instruction-Following LLM
AutoRubric-R1V (Jia et al., 2025)	Trajectory-level	Multimodal Reasoning	Reasoning LLM
RLCER (Sheng et al., 2026)	Trajectory-level	Math and General Knowledge Reasoning	Reasoning LLM
DR Tulu (Shao et al., 2025)	Trajectory-level	Deep Research	Agent
OralGPT-Plus (Fan et al., 2026a)	Trajectory-level	Dental Multimodal Diagnosis	Agent
Agent-World (Dong et al., 2026)	Trajectory-level	Long-horizon Tool Use	Agent
II. Advanced Reward Design			
RBR (Mu et al., 2024)	Answer-level	Safety Alignment	Instruction-Following LLM
Rubicon (Huang et al., 2025)	Answer-level	Creative Writing, Instruction Following	Reasoning LLM
SYNTHAGENT (Lü et al., 2026)	Trajectory-level	Agentic Tool Use	Agent
StitchCUDA (Li et al., 2026b)	Answer-level	Code Generation	Agent
RuCL (Chen et al., 2026b)	Trajectory-level	Visual Reasoning	Reasoning LLM
ARL-RR (Lan, 2026)	Answer-level	Deep Research	Reasoning LLM
PAPO (Tan et al., 2026)	Trajectory-level	Math Reasoning, Code Generation	Reasoning LLM
RTT (Xu et al., 2026b)	Answer-level	Instruction Following	Instruction-Following LLM
DRO (Xu et al., 2026c)	Answer-level	Paragraph Revision, Medical QA, Reasoning	Reasoning LLM
III. Rubrics as Policy Guidance			
RuscaRL (Zhou et al., 2025)	Trajectory-level	Medical, Writing, Instruction Following	Reasoning LLM
Think-with-Rubrics (Yu et al., 2026a)	Trajectory-level	Instruction following	Instruction-Following LLM
HeRL (Zhang et al., 2026b)	Answer-level	Instruction Following, Writing, Medical	Reasoning LLM

ambiguously defined or overlapping criteria, such as conflating features between fluency and prosody, directly degrades human inter-rater reliability. Interestingly, automated models perform better on these overlapping dimensions than on the clearly defined "accuracy" construct, as temporal features are computationally easier to capture than segmental quality. This reveals that a rubric's effective operationalization depends not solely on distinct conceptual boundaries, but also on the computational accessibility of its underlying features. Mittal et al. (2026) introduce the TRACE framework to reveal the "implicit weight misalignment" between LLM judges and human developers. By automatically extracting criteria from preference pairs, they quantify how judges systematically deviate from human intuition across different task modalities; for instance, LLMs may overvalue functional logic in code completion while humans prioritize code clarity, indicating that rubric evaluation must verify whether a model's internal application of criteria aligns with specific task contexts.

4 Rubrics for Model Training

After rubric construction, one of its important application is to support **policy model training**. Once constructed, rubrics provide structured supervision by decomposing response quality into explicit and interpretable criteria, which can then be transformed into rewards for downstream optimization. Beyond policy optimization, rubrics have also been increasingly adopted in **reward model training**, where they are used to provide more stable and reliable supervision signals for policy learning. This section introduces these two directions separately: Section 4.1 focuses on policy model training, while Section 4.2 discusses reward model training.

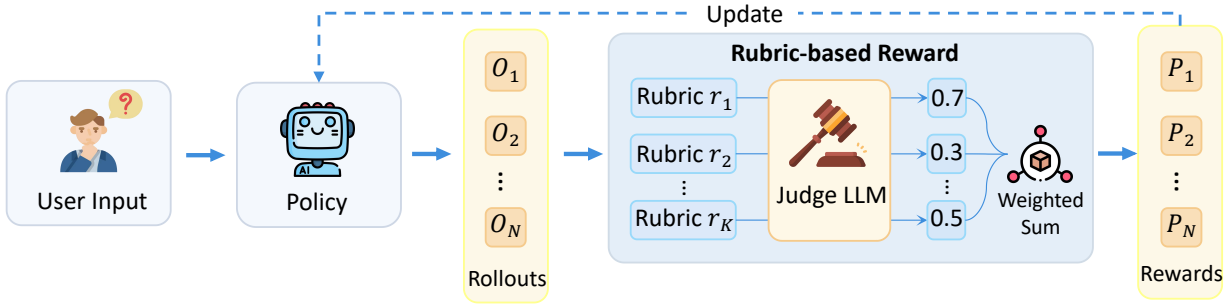

Figure 4 A standard rubric-based policy training pipeline with GRPO algorithm.

4.1 Policy Model Training

Rubrics have emerged as a principled and interpretable way to define reward signals for reinforcement learning (RL) in policy model training. Instead of relying on opaque reward models or rule-based scoring functions, rubrics explicitly decompose evaluation into a set of human-understandable criteria, enabling fine-grained supervision for complex, open-ended tasks. A standard approach is to use rubrics to score model-generated responses and aggregate these scores into a scalar reward, which is then used to optimize the policy via RL algorithms such as PPO (Schulman et al., 2017) or GRPO (Shao et al., 2024b).

Recently, a growing number of works have begun to explore how to leverage rubrics for more effective policy model training in RL. Broadly, these works can be categorized into two main directions: (1) advanced reward design, and (2) Rubrics as Policy Guidance. The former focuses on addressing the limitations of naive scalar aggregation of multi-dimensional rubric scores, aiming to provide more stable, expressive, and reliable reward signals for policy model training. The latter, in contrast, seeks to move beyond treating rubrics as purely post-hoc evaluation tools, and instead leverages them as prior input or generation guidance during policy training to steer exploration toward higher-quality rollouts, thereby improving both the upper-bound performance of the policy and the convergence efficiency of RL. We first introduce the standard rubric-based RL paradigm and then discuss these two directions in detail. The papers mentioned in this section are summarized in Table 2.

4.1.1 Standard Rubric-based RL

The standard paradigm applies a set of rubrics to evaluate the model outputs and converts them into scalar rewards for RL training. Formally, given an input x and a set of associated rubrics $\mathcal{R}_x = \{(r_k, w_k)\}_{k=1}^{K}$, where each r_k denotes an evaluation criterion and w_k its corresponding weight, the quality of a policy-generated rollout O is assessed via a rubric-based scoring function:

$$S(x, O) = \frac{\sum_{k=1}^{K} w_k \cdot \text{Judge}(r_k, O)}{\sum_{k:w_k>0} w_k}. \tag{8}$$

Here, $\text{Judge}(r_k, O)$ denotes a scoring function (often implemented by a judge LLM) that evaluates the extent to which the response O satisfies rubric r_k. A typical rubric-based RL training pipeline with GRPO is illustrated in Figure 4. Depending on the task and policy model type, rubric-based evaluation can be applied at two different levels: answer-level (Viswanathan et al., 2025; Goel et al., 2025; Gunjal et al., 2025; Jia et al., 2026; Zhang et al., 2025; Wu et al., 2025b) and trajectory-level such as training reasoning LLMs (Jia et al., 2025; Sheng et al., 2026) or tool-calling agents (Shao et al., 2025; Fan et al., 2026a; Dong et al., 2026). The aggregated rubric score $S(x, O)$ can then be used as a scalar reward signal for RL optimization, where standard algorithms such as PPO or GRPO are employed to maximize the expected reward. Alternatively, rubric scores can also be used to construct preference pairs (Yu et al., 2026b), which are then used to optimize the policy model using preference learning algorithms such as DPO (Rafailov et al., 2023).

4.1.2 Advanced Reward Design

Reward design plays a central role in RL, as it directly determines the optimization signal that guides policy updates. In standard rubric-based RL, rewards are typically constructed by aggregating multiple rubric scores with fixed or human-designed weights into a single scalar. Such scalar aggregation is often coarse and inflexible, and is prone to reward hacking (Mahmoud et al., 2026), leading to unstable and suboptimal policy optimization.

To address these issues, recent work has explored more stable and advanced reward aggregation strategies. RBR (Mu et al., 2024) learns aggregation weights for rubric-based rewards via pairwise preference optimization, where the weights are trained to assign higher scores to preferred responses than less preferred ones, providing a data-driven alternative to fixed heuristic weighting. Rubicon (Huang et al., 2025) proposes advanced aggregation strategies, such as veto mechanisms, saturation-aware aggregation, and pairwise interaction modeling, moving beyond linear combinations to capture non-linear interdependencies between rubric dimensions. SYNTHAGENT (Lü et al., 2026) and StitchCUDA (Li et al., 2026b) further develop structured reward formulations by integrating rubric-based signals with environment-based feedback (*e.g.*, subgoal completion, interaction constraints, and execution requirements) into unified reward functions, along with hard constraints (*e.g.*, veto-style gating) to mitigate reward hacking. RuCL (Chen et al., 2026b) extends reward design with a curriculum-based strategy that dynamically adjusts the weights of rubrics with different difficulty levels throughout training. It stratifies rubrics into easy and hard subsets, where easier rubrics are emphasized in early stages to stabilize learning, while more challenging rubrics are gradually prioritized in later stages. Different from the weighting-based approaches mentioned above, ARL-RR (Lan, 2026) observes that aggregating multiple rubrics into a single scalar reward can cause the policy to lose its awareness of rewards from different rubrics, which may suppress useful learning signals. Instead of compressing rubrics, it optimizes rubrics sequentially, alternating across them during training, and dynamically selects the next rubric via a lightweight search-based strategy based on task performance.

Beyond improving rubric reward aggregation, some work focuses on designing reward signals that better align with the RL algorithms. PAPO (Tan et al., 2026) identifies that instability can arise from how reward signals are incorporated into RL algorithm GRPO, where joint normalization of outcome and process rewards may obscure useful gradients. It addresses this by decoupling these signals at the advantage level, thereby preserving effective learning signals while avoiding interference between correctness and reasoning quality. RTT (Xu et al., 2026b) further improves the alignment between rubric-based rewards and RL optimization by addressing the sparsity of response-level rewards. It bridges response-level scoring and token-level policy learning by measuring the contribution of individual tokens to rubric rewards, and proposes RTT-GRPO, which combines response-level and token-level advantages within a unified framework. DRO (Xu et al., 2026c) uses rubrics for rollout group-level rejection rather than direct reward aggregation in unverifiable tasks. It applies query-specific binary rubrics as feasibility gates to discard rollout groups that fail to satisfy basic task requirements, preventing GRPO from producing misleading relative advantages among uniformly low-quality rollouts.

4.1.3 Rubrics as Policy Guidance

Reinforcement learning for LLMs often suffers from inefficient exploration in complex tasks, where policies are prone to diversity collapse or become trapped in local optima (Wu et al., 2025a; Dai et al., 2025). As a result, high-quality solutions are sparse and difficult to discover through unguided sampling. In contrast, rubrics contain structured and interpretable criteria, providing explicit guidance that can steer exploration toward more promising regions of the solution space.

Building on this intuition, recent work has explored how to leverage rubrics as explicit guidance for improving exploration in reinforcement learning. RuscaRL (Zhou et al., 2025) incorporates rubrics directly into the generation process as structured scaffolds, decomposing complex tasks into a sequence of rubric-aligned sub-steps. By iteratively refining outputs along different evaluation dimensions, it reduces the search space and improves credit assignment, enabling more efficient discovery of high-quality reasoning trajectories. Think-with-Rubrics (Yu et al., 2026a) further internalizes rubrics into the policy's generation process. Instead of using rubrics only as post-hoc reward criteria, it trains the model to first generate a structured rubric and then produce an answer conditioned on this self-generated rubric. During reinforcement learning, the policy

Table 3 Representative methods using rubrics for reward model training. Methods are grouped by their primary usage of rubrics, including interpretability, reward signal shaping, and data construction.

Method	Usage of Rubrics	Rubrics in Input	Reasoning in Output	Training Scheme	Key Contribution
R3 (Anugraha et al., 2025b)	Ctrl. & Interp.	✓	✓	SFT	Rubric-conditioned reward reasoning enabling rubric-agnostic and explainable evaluation.
mR3 (Anugraha et al., 2025a)	Ctrl. & Interp.	✓	✓	SFT	Multilingual rubric-conditioned reward reasoning across languages.
CDPRM (Liu et al., 2026a)	Ctrl. & Interp.	✓	✓	SFT	Rubric-conditioned reward reasoning grounded in distilled task-specific rubrics.
RRM (Yuan et al., 2025)	Ctrl. & Interp.	✓	✓	SFT+RL	Reduces false-positive reasoning via rubric-based process-level rewards.
OMNI-RRM (Kong et al., 2026)	Ctrl. & Interp.	✓	✓	SFT+RL	Format-aware reward shaping enforcing reasoning grounded in all provided rubrics.
C2 (Kawabata and Sugawara, 2026)	Ctrl. & Interp.	✓	✓	RL	Requires explicit judgment of whether rubrics are helpful or misleading, rewarding correctness of this utility prediction.
DeltaRubric (Liu et al., 2026b)	Ctrl. & Interp.	✓	✓	RL	Decomposes evaluation into plan-and-execute stages, rewarding self-generated rubrics that rectify baseline evaluative errors.
ArmoRM (Wang et al., 2024a)	Ctrl. & Interp.	✓	✗	MSE+BT	Decomposes reward into interpretable rubric dimensions with MoE aggregation.
Critic Rubrics (Wang et al., 2026b)	Ctrl. & Interp.	✓	✗	MSE	Joint prediction of sparse outcomes and dense rubric-aligned signals.
MR-RML (Jin et al., 2025)	Ctrl. & Interp.	✓	✗	BT	Uses geometric projection to compute rubric-aligned scores.
METAJUDGE (Wang et al., 2026a)	Reward Signals	✗	✓	RL	Aligns reward reasoning with rubric-derived human rationales.
Proxy-GRM (Qiu et al., 2026)	Reward Signals	✗	✓	SFT+RL	Proxy-guided reward shaping for self-generated rubrics before judgment.
CROME (Srivastava et al., 2025)	Data Construction	✗	✗	BT	Uses rubrics as causal intervention tools to construct robust preference datasets.

is optimized with both reference-rubric consistency and self-rubric consistency, so that rubrics serve as an explicit intermediate plan that guides response generation before the final answer is produced. HeRL (Zhang et al., 2026b) leverages rubrics as feedback signals by converting unmet rubrics into hindsight experience, which is injected as in-context guidance to improve model outputs. This feedback-driven mechanism provides supervision on how to improve a given response and helps explore higher-quality responses beyond the current policy distribution. These approaches demonstrate that rubrics can serve not only as evaluation signals but also as structured guidance for policy optimization, transforming exploration from unguided sampling into a more directed and efficient process.

4.2 Reward Model Training

Beyond their role in policy model training, rubrics have also been increasingly adopted in reward model training, which is essential for providing scalable, stable, and cost-effective reward signals compared to directly querying LLMs during policy optimization, and also enables broader applications such as offline evaluation, filtering, and ranking. Existing studies demonstrate that rubrics can support reward model training from multiple perspectives, as illustrated in Figure 5. Specifically, they have been applied to improve the interpretability of reward models, to provide dense and informative reward signals during optimization, and to facilitate the construction of high-quality training data. These three directions highlight the diverse roles that rubrics can play in different aspects of reward model training. Representative papers mentioned in this section are shown in Table 3.

4.2.1 Rubrics for Interpretability

Traditional reward models typically produce a single scalar score as output, offering limited transparency into the underlying evaluation process. In such settings, the evaluation criteria used by the model remain implicit, making it difficult to determine which aspects of a response contribute to the final score or whether the model follows the intended evaluation standards.

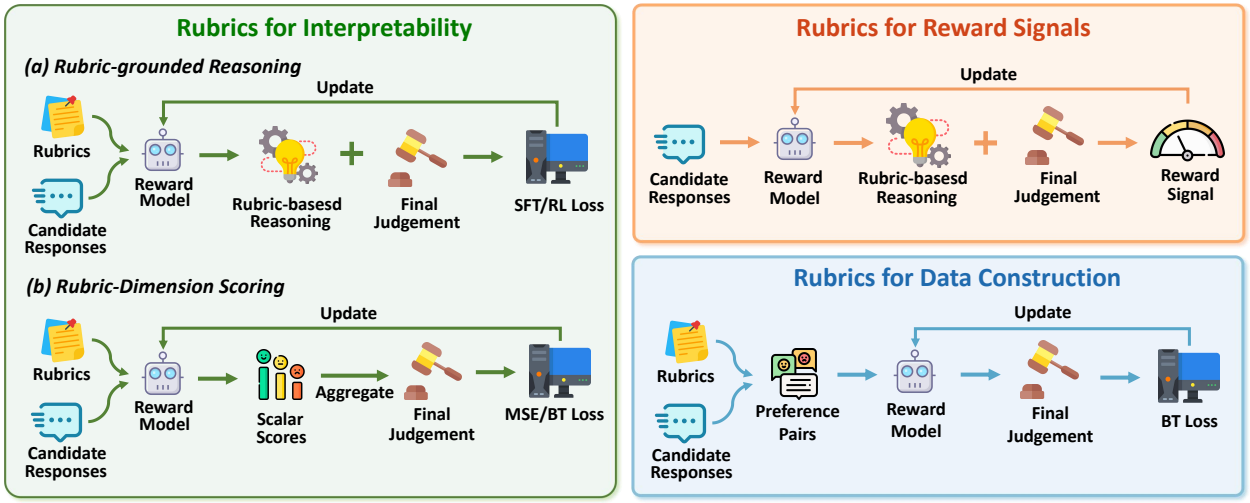

Figure 5 Overview of three key directions in which rubrics support reward model training, including enhancing interpretability, providing dense reward signals, and enabling high-quality training data construction.

Introducing explicit rubrics into reward modeling transforms the evaluation process into a structured procedure guided by predefined rubrics. Instead of relying on latent evaluation patterns, the reward model is encouraged to evaluate responses according to clearly specified rubrics, either by generating rubric-grounded reasoning to support preference judgments or by assigning structured scores to individual rubric dimensions. This structured use of rubrics improves interpretability by exposing the rationale behind preference judgments and clarifying how different evaluation criteria contribute to the final reward.

Rubric-grounded Reasoning. In this line of work, rubrics are incorporated into reward models by requiring the model to generate rubric-grounded reasoning before producing the final preference judgment. Rubrics are provided as part of the input, guiding the model to compare candidate responses across multiple rubric dimensions and produce structured reasoning prior to outputting a final decision.

Most methods in this line rely on supervised fine-tuning using rubric-conditioned data generated by stronger teacher models, where the supervision signals typically include both preference labels and teacher-generated reasoning aligned with the rubrics. Representative works adopt this paradigm with different focuses. R3 (Anugraha et al., 2025b) trains a reward model capable of performing interpretable scoring across diverse rubric settings, enabling consistent reasoning under heterogeneous rubrics. mR3 (Anugraha et al., 2025a) extends this framework to multilingual settings by leveraging large-scale multilingual data, allowing rubric-grounded evaluation across different languages and rubric conditions. CDPRM (Liu et al., 2026a) first deeply contrasts preference data to distill customized rubrics, and then trains reward models to generate reasoning trajectories and final preference judgments strictly grounded in these rubrics.

Beyond supervised fine-tuning, several studies further use reinforcement learning (RL) to refine rubric-grounded preference prediction. RRM (Yuan et al., 2025) defines the RL reward function based on the accuracy of preference judgments. Building upon this design, OMNI-RRM (Kong et al., 2026) further incorporates format-aware rewards that encourage generated reasoning to reference all provided rubrics and include explicit comparative expressions. Going a step further, C2 (Kawabata and Sugawara, 2026) requires the reward model to first output a judgment of whether provided rubrics are helpful or misleading, and incorporates the correctness of this judgment into the reward signal. This incentivizes the model to learn rubric utility assessment, ultimately enabling inference-time gating that discards misleading guidelines in favor of rubric-free reasoning. DeltaRubric (Liu et al., 2026b) further extends this by jointly optimizing an instance-specific planner and a grounded verifier through multi-role RL, utilizing a relative improvement reward to prioritize the generation of rubrics that successfully flip baseline judgment errors.

Rubric-Dimension Scoring. In contrast to rubric-grounded reasoning, another line of work incorporates rubrics by assigning separate scalar scores to individual rubric dimensions and aggregating these scores into a final reward. In this paradigm, reward models are typically built upon an LLM backbone, complemented by lightweight prediction modules such as linear layers. This design enhances interpretability by making the contribution of each rubric dimension observable and explicitly revealing how different dimensions are aggregated into the final reward signal.

Representative works adopt different mechanisms to implement rubric-dimension scoring. ArmoRM (Wang et al., 2024a) trains a multi-objective reward model that predicts absolute scores for multiple rubric dimensions and employs a mixture-of-experts gating mechanism to dynamically assign weights to different dimensions based on the input context. It jointly trains dimension-level scores using mean squared error (MSE) loss while optimizing the aggregated reward with Bradley–Terry (BT) loss. Critic Rubrics (Wang et al., 2026b) transforms previously unused interaction trajectories into structured supervision signals by extracting multiple rubric-aligned behavioral features and jointly predicting these features alongside sparse human feedback. It trains reward models to predict dimension-level scores using regression objectives such as MSE loss. MR-RML (Jin et al., 2025) introduces a geometric projection-based formulation in which both responses and rubric dimensions are represented as vectors, and dimension scores are computed based on projection lengths along rubric directions. The aggregated reward is then optimized using preference-based objectives such as BT loss.

4.2.2 Rubrics for Reward Signals

Beyond their use for interpretability, rubrics can also be incorporated into the reward model optimization process to construct fine-grained reward signals during reinforcement learning. Traditional reinforcement learning for reward models often relies on relatively coarse supervision, such as binary correctness labels or preference agreement between responses. While these signals are effective for guiding overall preference alignment, they provide limited feedback on specific evaluation dimensions, making it difficult to enforce detailed evaluation standards or capture nuanced differences between responses. By introducing rubrics as structured reference units, reinforcement learning objectives can incorporate more informative supervision that reflects how well a model's response aligns with predefined evaluation criteria.

Representative works explore different strategies to incorporate rubric-based signals into reinforcement learning. METAJUDGE (Wang et al., 2026a) introduces atomic reasoning alignment, where human annotations are decomposed into atomic reasoning units that serve as rubric-level references. During reinforcement learning, the reward function evaluates the alignment between atomic reasons generated by the reward model and those extracted from human annotations, using this alignment score as part of the optimization signal to constrain reasoning quality and encourage consistency with rubric-defined criteria. Proxy-GRM (Qiu et al., 2026) adopts a different strategy by requiring the reward model to first generate rubrics before performing analysis and producing the final judgment. To further guide rubric generation quality, the framework introduces an auxiliary proxy model that independently evaluates the generated rubrics, and incorporates its assessment as an additional reward signal during reinforcement learning. This design encourages the model to produce high-quality rubrics that can support reliable downstream evaluation.

4.2.3 Rubrics for Data Construction

Rubrics also play an important role in training data construction by guiding the creation of high-quality and informative data for reward model learning. Conventional preference datasets often contain superficial cues, such as response length, formatting style, or repeated phrasing, which reward models may exploit instead of learning to assess the substantive quality of responses. By incorporating rubrics into the data generation process, it becomes possible to construct training data that emphasizes the core dimensions of response quality, reduces reliance on surface-level heuristics, and mitigates the risk of reward hacking.

A representative method in this line is CROME (Srivastava et al., 2025), which automatically extracts causal rubrics that determine answer quality and generates both contrastive samples that modify only the core quality and tie samples that alter only superficial features. This intervention forces the reward model to base its evaluations on the true causal criteria rather than surface cues, improving alignment and robustness.

Table 4 Representative rubric-driven benchmarks for general evaluation scenarios.

Benchmark	Target Scenario	Rubric Design
FLASK (Ye et al., 2024)	Alignment capability profiling	Fine-grained skill-set rubrics decompose helpfulness, truthfulness, harmlessness, and instruction adherence into explicit criteria.
G-Eval (Liu et al., 2023)	Open-ended NLG quality	Criterion lists with chain-of-thought form-filling produce transparent, rubric-grounded scoring decisions.
InFoBench (Qin et al., 2024)	Instruction following	Multi-constraint instructions are decomposed into verifiable checklist items for the fine-grained adherence scoring.
AdvancedIF (He et al., 2025b)	Complex instruction following	Atomic rubrics with all-or-nothing and anti-cheating penalties stress precise constraint satisfaction.
WildBench (Lin et al., 2024)	Open-ended chat	Task-specific checklists built from real-user tasks provide structured evidence for judge decisions.
MultiChallenge (Deshpande et al., 2025)	Multi-turn interaction	Binary criteria track cross-turn consistency, instruction memory, and conversational state tracking.
MathCheck (Zhou et al., 2024)	Mathematical reasoning	Checklist-based rubrics evaluate process validity, answer correctness, and robustness of reasoning.
SedarEval (Fan et al., 2024)	Reasoning-intensive tasks	Self-adaptive question-level rubrics provide primary/secondary criteria with deduction rules.
HelloBench (Que et al., 2024)	Long-form generation	Hierarchical primary/secondary rubrics rooted in Bloom-style capability layers for long responses.
ResearchRubrics (Sharma et al., 2025)	Deep research reports	Prompt-specific fine-grained binary rubrics explicitly score breadth, depth, and evidence grounding.
AgentBoard (Ma et al., 2024)	Multi-domain agent interaction	Capability-oriented analytic criteria evaluate planning, memory, grounding, and multi-turn execution quality.
AdaRubric (Ding, 2026)	Agent trajectory evaluation	Task-adaptive rubrics dynamically calibrate dimensions for different agent environments and objectives.
TRAJECT-Bench (He et al., 2025a)	Tool-use trajectories	Trajectory criteria separately evaluate tool selection, argument correctness, and dependency ordering.
MCP-Universe (Luo et al., 2025)	MCP-based agent ecosystems	Structured format/static/dynamic validators define explicit rubric-like criteria across heterogeneous server tasks.
$1M-Bench (Yang et al., 2026b)	Professional agentic tasks	Expert rubrics score factuality, feasibility, verbalization, and instruction compliance.
JudgeBench (Tan et al., 2025)	Judge reliability evaluation	Challenging preference pairs diagnose judge robustness under objective correctness-oriented criteria.
StrongREJECT (Souly et al., 2024)	Safety alignment	Multi-dimensional safety rubric (refusal quality, specificity, and harmful compliance) measures jailbreak resistance.

5 Rubrics for Evaluation

Previous sections examined how rubrics are constructed and used for model training; this section focuses on how they function as operational interfaces for model evaluation. In open-ended settings, rubric-driven evaluation translates underspecified notions (*e.g.*, helpfulness, faithfulness, safety, and process soundness) into auditable criteria that support both scoring and diagnosis. Table 4 and Table 5 summarize representative benchmarks, and the discussion is organized by evaluation scope. We organize our discussion into two progressive dimensions: (1) general-task evaluation and (2) domain-specific task evaluation.

5.1 Rubrics for General Tasks Evaluation

General-task evaluation asks whether model outputs meet broad user expectations in settings where no single ground-truth answer exists. In this scenario, rubrics act as explicit evaluation contracts: they make implicit criteria visible, impose a uniform level of granularity on judgments, and leave an auditable trail that supports both scoring and error analysis. We categorize rubric-based general-task evaluation into four categories according to the underlying LLM capabilities being assessed:

5.1.1 Reasoning Capability Evaluation

Rubric-driven reasoning evaluation shifts emphasis from outcome-only correctness toward explicit assessment of *how* a model reasons. In mathematical and formal settings, MathCheck (Zhou et al., 2024), SedarEval (Fan et al., 2024), and RefGrader (Mahdavi et al., 2025) decompose reasoning into criterion-level subskills, including task understanding, intermediate validity, and final answer quality; this design enables precise failure localization. A similar paradigm appears in code generation and technical reasoning, where RubricCode (Pathak

et al., 2025) and TRACE (Mittal et al., 2026) evaluate implementation logic, error handling, and rationale consistency beyond binary pass/fail signals. In professional and normative reasoning contexts, ProfBench (Wang et al., 2025c), $1M-Bench (Yang et al., 2026b), and MoReBench (Chiu et al., 2025) adopt multidimensional criteria that jointly capture domain validity, analytical rigor, and value-sensitive consistency. Complementary rubric-construction work, including Qworld (Gao et al., 2026) and Search-Gen-V (Ma et al., 2025), further indicates that question-conditioned and verifiable criteria improve discriminative performance. Collectively, these studies suggest that robust reasoning evaluation depends on explicit separation between process quality and outcome quality.

5.1.2 Deep Research and Open-Ended Generation Evaluation

In long-form generation and deep-research tasks, quality is jointly determined by coverage, factual grounding, synthesis depth, and report organization. Recent work therefore focuses on compositional rubric design. For general long-form generation, HelloBench (Que et al., 2024) and WritingBench (Wu et al., 2025c) adopt hierarchical rubrics that evaluate content adequacy and discourse-level presentation along separate dimensions. In deep-research evaluation, DeepResearch Bench (Du et al., 2025), DeepResearchBenchII (Li et al., 2026a), DEER (Han et al., 2025), and ResearchRubrics (Sharma et al., 2025) operationalize evidence-oriented criteria for recall, analytical soundness, citation-grounded argumentation, and report-level utility. The same logic extends beyond text-only outputs: MiroEval (Ye et al., 2026) adds process-and-outcome co-assessment in multimodal research scenarios, while PencilsDown (Farzi and Dietz, 2024) demonstrates rubric-style itemization in retrieval-grounded generation. At the infrastructure layer, Autorubric (Xie et al., 2025) and RubricHub (Li et al., 2026c) show that scalable verification, normalization, and compression of rubric items are central to reliable open-ended evaluation.

5.1.3 General Agent Capability Evaluation

As evaluation targets shift from static responses to interactive execution, rubric design moves from answer-level dimensions to trajectory-level diagnostics. Capability-oriented benchmarks such as AgentBoard (Ma et al., 2024) and AdaRubric (Ding, 2026) characterize agent behavior through explicit dimensions (*e.g.*planning, memory, and grounding), improving interpretability in cross-agent comparison. Tool-use benchmarks, including TRAJECT-Bench (He et al., 2025a) and MCP-Universe (Luo et al., 2025), prioritize executable criteria over tool selection, argument validity, and dependency/order constraints in realistic ecosystems. Multi-turn assistant settings (MultiChallenge (Deshpande et al., 2025), Scribe (Jiang and Ferraro, 2026), AstraBench (Xiu et al., 2026)) extend this with process-oriented criteria for sub-goal completion, conversational consistency, and interaction reliability. Evidence from long-horizon tasks, including PaperBench (Starace et al., 2025) and DR-Tulu-style (Shao et al., 2025) evaluations, indicates that final success can mask substantial process failures, reinforcing the need for rubric designs that preserve intermediate diagnostic signals.

5.1.4 Alignment Evaluation

Alignment-oriented evaluation uses rubrics to unify heterogeneous constraints from user intent, system instruction, and safety policy within a common scoring interface. Benchmarks such as FLASK (Ye et al., 2024), InFoBench (Qin et al., 2024), AdvancedIF (He et al., 2025b), and WildBench (Lin et al., 2024) decompose alignment into checkable dimensions over instruction adherence, utility, and response quality, improving comparability across diverse prompts and interaction settings. On the evaluator side, rubric-conditioned judging in G-Eval (Liu et al., 2023), Prometheus (Kim et al., 2024), and MT-Bench/Chatbot-Arena-style (Zheng et al., 2023) protocols yields more explicit rationales and better calibration than unconstrained holistic judgments. Meta-evaluation studies, including RubricEval (Pan et al., 2026), RubricBench (Zhang et al., 2026a), and JudgeBench (Tan et al., 2025), show that reliable rubric execution by judges remains a bottleneck, especially on hard and fine-grained items. In safety-sensitive settings, StrongREJECT (Souly et al., 2024) further indicates that robust auditing increasingly depends on explicit, verifiable policy-oriented dimensions rather than latent preference proxies.

Across these four categories, a consistent trend emerges: evaluation criteria are becoming more explicit, process-aware, and scenario-adaptive. This shift improves score reliability and diagnostic granularity in open-ended settings, but also reveals the boundaries of general-task rubrics: they are designed for transfer and

Table 5 Representative rubric-driven benchmarks for domain-specific evaluation scenarios.

Benchmark	Target Scenario	Rubric Design
PencilsDown (Farzi and Dietz, 2024)	RAG systems	Query-derived rubrics assess relevance, completeness, factuality, and conciseness.
HealthBench (Arora et al., 2025)	Medical QA	Expert criteria evaluate medical correctness, completeness, and communication quality.
MedMT-Bench (Yang et al., 2026a)	Medical QA	Atomic test points evaluate memory, safety, clarification, interference robustness, and multi-intent handling.
RubricRAG (Dhole and Agichtein, 2026)	Medical QA and deep research	Retrieval-augmented criteria dynamically assess query-specific desirable behaviors and penalize failure modes.
DeepResearch Bench (Du et al., 2025)	Deep research reports	Adaptive criteria assess report quality and citation-grounded evidence collection.
DeepResearchBenchII (Li et al., 2026a)	Deep research reports	Binary expert rubrics score recall, analysis, and presentation quality.
DEER (Han et al., 2025)	Deep research reports	Taxonomy-driven criteria assess request fulfillment, analytical rigor, and document coherence.
LREAD (Park and Han, 2026)	LLM text detection	Human-calibrated rubrics target grammaticality, coherence, and structural artifacts.
PaperBench (Starace et al., 2025)	Research replication	Hierarchical rubrics grade code development, execution validity, and reproducibility.
PresentBench (Chen et al., 2026a)	Slide generation	Fine-grained rubrics assess layout, visual consistency, and content completeness.
ProfBench (Wang et al., 2025c)	Professional domains	Domain rubrics assess information extraction, reasoning, causal consistency, and readability.
MoReBench (Chiu et al., 2025)	Moral reasoning	Theory-grounded rubrics score moral identification, process coherence, and helpfulness.
RubricCode (Pathak et al., 2025)	Educational code grading	Question-specific rubrics assess data logic, algorithm correctness, formatting, and complexity.
SpeechL2 (Parikh et al., 2026)	L2 speech assessment	Multi-aspect rubrics evaluate pronunciation, fluency, and prosodic quality.
Scribe (Jiang and Ferraro, 2026)	Math and tool-use	Skill-conditioned rubrics target sub-goal correctness, execution flaws, and logical gaps.
AstraBench (Xiu et al., 2026)	Personal-assistant tool use	DAG-style rubrics evaluate task completion, tool invocation, and interaction reliability.
PRBench (Akyürek et al., 2025)	Finance and law reasoning	Weighted expert rubrics assess accuracy, process auditability, uncertainty handling, and risk disclosure.
TechImage-Bench (Ni et al., 2026)	Technical image generation	Structural rubrics assess entity presence, spatial relations, and semantic accuracy.
LongShotBench (Kurpath et al., 2025)	Long video omni-modal reasoning	Graded rubrics assess factuality and penalize temporal, entity, and hallucination errors.
DataRubrics (Winata et al., 2025)	Dataset quality assessment	Automated rubrics assess dataset documentation, reproducibility, code availability, and accountability.
LLM-RUBRIC (Hashemi et al., 2024)	Information-seeking dialogue	Neural-calibrated rubrics model objective quality and judge-specific subjective preferences.
DRESS (Yoo et al., 2025)	EFL essay scoring	Scoring content, structure, and language proficiency.
STORM (Shao et al., 2024a)	Long-form article writing	Outline-level rubrics evaluate information coverage, organization, and perspective diversity.
RINoBench (Schopf and Färber, 2026)	Research idea novelty judgment	Expert-derived 5-point rubric assessing novelty degree with multi-metric evaluation for textual justifications.

cannot fully encode domain-specific constraints, professional standards, or environment-grounded verification requirements. This limitation motivates the next subsection, which examines rubric specialization for concrete downstream domains.

5.2 Rubrics for Domain-Specific Tasks Evaluation

While general-task rubrics provide transferable evaluation interfaces, downstream applications impose domain-specific constraints that cannot be fully captured merely by broad criteria. Domain-specific task evaluation therefore emphasizes rubric specialization under concrete knowledge boundaries, safety requirements, and environmental feedback. We organize this part into two complementary views: rubric designs for intermediate trajectories and rubric designs for final outputs.

5.2.1 Rubrics for Intermediate Trajectories

As LLMs evolve into agentic systems, the evaluation paradigm is shifting from exclusively assessing static final answers to evaluating dynamic, intermediate reasoning trajectories. Trajectories in domain-specific

downstream applications are highly dependent on domain-specific environments. Therefore, evaluating these complex processes requires customized, instance-specific rubrics capable of tracking whether the LLM's intermediate reasoning aligns with unique environmental feedback and professional standards.

Recent benchmarks have increasingly adopted a strategy of decomposing complex execution processes into verifiable and fine-grained components. In the domain of information seeking, GISA (Zhu et al., 2026) provides complete human search trajectories for every query, recording search queries issued, search engine results, and click-through behaviors with timestamps. While GISA does not use trajectories as formal evaluation rubrics, its analysis reveals that alignment with human search strategies positively correlates with task performance, suggesting that process-level trajectory data can serve as a meaningful signal for diagnosing agent weaknesses. In the context of autonomous tasks like research paper replication, PaperBench (Starace et al., 2025) decomposes the task pipeline into several modules, including code development and execution. The associated rubrics are designed to pinpoint exact locations of failure and provide partial credit even if the agent is not able to finish the final task. Beyond engineering tasks, MoReBench (Chiu et al., 2025) focuses on moral reasoning tasks. Its core claim is that the logical completeness in the reasoning process is more important than the final moral decision, because moral dilemmas may not have objectively correct answers. Hence, it mainly focuses on auditing the internal thinking traces based on expert-written rubrics covering moral factor identification, process clarity, logical coherence, and outcome helpfulness.

For scenarios where **intermediate trajectories cannot be directly obtained** (*e.g.*, black-box evaluated agents), several studies instruct the models to present intermediate reasoning steps within the final outputs. ProfBench (Wang et al., 2025c) studies professional domains like PhD-level questions in physics and chemistry, as well as MBA-level issues. Its rubrics explicitly require the model to demonstrate intermediate calculation and reasoning, allowing the evaluation framework to further assess whether the information in the grounded documents is accurately extracted. Similarly, Pathak et al. (2025) study code quality evaluation by following a point-wise, question-specific rubric approach to verify logical trajectories in each code implementation step. For tool-augmented agents, Scribe (Jiang and Ferraro, 2026) standardizes intermediate evaluation by routing sub-goals to predefined skill prototypes. This constrained verification approach equips LLM judges with skill-conditioned rubrics to directly assess sub-goal completeness, execution flaws, logical gaps, and common traps. MedMT-Bench (Yang et al., 2026a) uses instance-level rubrics to evaluate long multi-turn medical dialogues. It simulates the full process of medical consultation, diagnosis, and follow-up, and checks whether models can remember earlier information, avoid being misled by irrelevant context, follow medical safety rules, ask clarifying questions when user instructions are ambiguous, and respond to multiple user requests in one turn. These requirements are converted into atomic test points, so that LLM judges can evaluate complex medical dialogues through fine-grained yes-or-no checks.

5.2.2 Rubrics for Final Outputs

While trajectory analysis provides useful information about the execution process, final outputs remain the main results shown to users. Therefore, output-oriented rubrics are still important for evaluating what users actually receive. Instead of assessing correctness alone, these output-oriented rubrics evaluate output quality from several related dimensions. Existing practices can be summarized into four dimensions:

Content Factuality. Content factuality serves as a foundational dimension of final-output evaluation. It is crucial to evaluate whether the final outputs are not only correct and comprehensive but also strictly grounded in the provided or retrieved context.

For example, in Retrieval-Augmented Generation (RAG) systems, PencilsDown (Farzi and Dietz, 2024) explicitly evaluates this grounding through query-derived rubrics that measure information relevance and factual completeness. Contemporary benchmarks have shifted from holistic correctness to granular, context-aware verification. Scaling this principle to deep research agents, DeepResearchBench (Du et al., 2025) introduces the FACT framework to evaluate deep research agents. It transforms holistic correctness into a verifiable metric by checking whether generated statements are accurately supported by their cited URLs, thereby quantifying citation trustworthiness and factual abundance in long-form research reports. However, since cited sources themselves may be inaccurate, DeepResearch Bench II (Li et al., 2026a) shifts focus toward expert-derived ground truth. It employs thousands of atomic, binary rubrics to rigorously verify specific factual and numerical claims, effectively rejecting seemingly correct hallucinations. This progression marks a shift

from verifying citation consistency to enforcing granular, expert-validated accuracy. In high-stakes scenarios, HealthBench (Arora et al., 2025) utilizes physician-authored rubrics to specifically evaluate whether the model misses critical medical safety warnings. Expanding into complex professional domains, PRBench (Akyürek et al., 2025) leverages expert-curated criteria to evaluate not merely factual statement accuracy, but the precise application of facts to legal and financial rules. In context-aware and personalized environments, AstraBench (Xiu et al., 2026) evaluates factuality through the lens of IR Recall and "No Hallucination" scores across complex and context-aware personal assistant tasks, such as cross-app information retrieval and temporal scheduling, ensuring that agentic responses align with the user's evolving personal context. Furthermore, across domain-specific tasks like querying certain APIs (for financial data) and executing code interpreters, Scribe (Jiang and Ferraro, 2026) incorporates targeted penalty rubrics to audit for errors like hallucinated outputs. This effectively transforms factual verification from a surface-level check into a diagnostic mechanism for identifying underlying reasoning failures.

Factuality has also been extended beyond text to multi-modal scenarios. For instance, TechImage-Bench (Ni et al., 2026) uses structural rubrics to check fine-grained entity presence, spatial relations, and semantic accuracy in technical scientific diagrams. Likewise, in long-video understanding, LongShotBench (Kurpath et al., 2025) uses graded rubrics to penalize temporal inconsistencies, entity mismatches, and multi-modal hallucinations, helping ensure that model outputs stay faithful to complex visual evidence.

Safety Auditing. Safety auditing serves as the critical counterpart to factuality. While factuality focuses on ensuring the presence of accurate and grounded information, safety auditing strictly enforces the *absence* of harmful, non-compliant, or high-risk content. In high-stakes environments, even a factually grounded output can lead to severe failures if it violates safety constraints or industry regulations.

Accordingly, recent benchmarks have reframed output evaluation into rigorous safety auditing by introducing severe penalty mechanisms. For instance, in the medical domain, HealthBench (Arora et al., 2025) and RubricRAG (Dhole and Agichtein, 2026) use explicit negative rubrics to penalize unsafe behaviors, such as hallucinated clinical claims, dangerous triage advice, or failure to identify emergency "red flags". MoReBench (Chiu et al., 2025) further evaluates the harmlessness of final outcomes in pluralistic moral dilemmas. Beyond evaluating the direct outputs of models, comprehensive safety auditing also extends to the foundational artifacts of AI systems. For example, DataRubrics (Winata et al., 2025) applies automated quality rubrics to assess the accountability, ethical documentation, and transparency of datasets, mitigating systemic downstream risks by enforcing rigorous compliance standards at the source.

Professional Presentation and Structural Coherence. While both content factuality and safety auditing constrain the internal content, presentation and coherence determine how effectively that content is consumed and trusted by the user. In real-world deployments, outputs must satisfy domain-specific norms for organization, formatting, and tone. DEER (Han et al., 2025) uses expert-developed rubrics to assess report-level coherence and readability. ProfBench (Wang et al., 2025c) similarly grades conciseness, structure, and professional tone. $1M-Bench (Yang et al., 2026b) includes verbalization quality as a core dimension for industry-style deliverables. Crucially, evaluating professional presentation also involves assessing the naturalness of these structures. For instance, LREAD (Park and Han, 2026) demonstrates that while LLMs excel at macro-structural coherence, they frequently produce overly regularized outputs characterized by structural rigidity, mechanical consistency, and subtle "translationese" that violate authentic stylistic norms. Beyond text, PresentBench (Chen et al., 2026a) extends these ideas to multimodal artifacts through checklists for layout and design consistency. Similarly in the audio modality, recent work establishes rubric-guided frameworks for SpeechLLMs, extending the concept of presentation coherence to acoustic delivery (Parikh et al., 2026). Specifically, this maps structural coherence to *fluency* (evaluating the temporal smoothness and uninterrupted coherence of speech) and stylistic naturalness to *prosody* (capturing the expressiveness and natural rhythm of intonation).

Practical Utility and Actionability. Practical utility and actionability assess whether the generated artifact successfully resolves the user's underlying problem under realistic constraints. Beyond the internal constraints of content factuality and safety auditing, utility- and actionability-oriented rubrics evaluate whether outputs are genuinely usable and whether they help complete the user's intended goal.

For instance, in deep research tasks, DEER (Han et al., 2025) includes "request fulfillment" criteria to verify that the final report comprehensively addresses every specific question raised by the user without evasion.

Beyond mere coverage, utility demands actionable guidance. In morally complex settings, MoReBench (Chiu et al., 2025) measures "outcome helpfulness," rewarding agents that provide clear, actionable paths rather than purely abstract philosophical discussions. This demand for actionability is even more critical in specialized professional domains. PRBench (Akyürek et al., 2025) utilizes expert-curated rubrics to strictly evaluate the "practical utility" and "procedural correctness" of legal and financial advice, ensuring the generated deliverables are not merely theoretically sound, but practically viable and executable for real-world clients. Furthermore, in context-aware personal assistant scenarios, AstraBench (Xiu et al., 2026) measures end-to-end "task completion" and "conversation effectiveness," evaluating whether the agent can orchestrate state changes to achieve the user's goal. Ultimately, the goal is to ensure the reliability of context-aware assistants in real-world use. To bridge the gap between predefined rubrics and diverse human expectations, LLM-RUBRIC (Hashemi et al., 2024) calibrates multidimensional evaluations using a personalized calibration network, modeling judge-specific subjective preferences to accurately predict the overall user satisfaction of the final artifact.

In summary, domain-specific task evaluation requires a shift toward customized, domain-aware rubrics. This process must systematically audit dynamic intermediate reasoning trajectories while comprehensively assessing final outputs across four core dimensions. Specifically, it must evaluate content factuality, conduct safety auditing to prevent harmful or non-compliant outputs, enforce professional presentation to maintain structural coherence, and verify practical utility to guarantee actionable real-world execution. Ultimately, this specialized paradigm ensures that LLMs generate artifacts that are not merely safe and theoretically sound, but fully aligned with rigorous professional standards and user-centric goals.

6 Open Questions and Discussion

Although rubric-based methods have rapidly emerged as a promising interface for specifying, evaluating, and optimizing LLM behavior, many foundational challenges remain unresolved. Unlike conventional evaluation protocols or scalar reward models, rubrics expose explicit criteria that can guide both judgment and training. This transparency improves interpretability, but it also introduces new challenges related to robustness, generalization, personalization, safety, and evaluation reliability. In this section, we will discuss these challenges in detail.

6.1 Robust Rubric Design against Reward Hacking

Reward hacking remains one of the most common failure modes in reinforcement learning, and rubric-based rewards do not escape this problem (Mahmoud et al., 2026). Compared with reference-based rewards or task-specific verifiable signals, rubrics expand the space of credit assignment by exposing multiple independently checkable criteria. This broader coverage is valuable for open-ended tasks, but it also enlarges the surface through which a policy can find shortcuts that satisfy rubric wording without genuinely improving response quality. Therefore, a central open question is how to design robust rubrics that provide rich supervision while reducing the risk of being exploited during optimization.

Granularity and Scope. Existing rubric construction methods are still largely developed for general LLM generation settings, where rubrics mainly evaluate broad response-level qualities such as fluency, coherence, helpfulness, harmlessness, and overall reasonableness. Accordingly, many optimization and evaluation protocols are closer to general reward-model benchmarks, where the goal is to judge whether one response is preferable to another under relatively generic standards. However, as LLMs are increasingly used in more concrete application scenarios, especially agent-related tasks, the required rubric granularity becomes much more fine-grained. In these settings, evaluation is no longer limited to whether the final response looks reasonable, but also concerns whether the agent retrieves the right evidence, follows constraints, executes intermediate steps correctly, and integrates information into a task-completing output.

This shift raises new challenges for rubric design. For example, in search-agent evaluation, rubrics may need to assess the correctness and relevance of each retrieved evidence piece, whether necessary facts are covered, whether temporal or numerical constraints are satisfied, whether claims are properly supported by sources, and whether the final answer avoids unsupported synthesis. These criteria are more specific than general response-quality dimensions, and their design and weighting can directly affect agent training

and final behavior. If rubrics overemphasize factual coverage, agents may retrieve and repeat many facts without producing a useful synthesis; if they overemphasize concise final answers, agents may ignore important evidence or constraints. However, existing rubric construction methods still lack systematic studies of such scenario-specific design choices. Future work should therefore investigate how rubric granularity and scope should vary across application settings, and how to balance general quality dimensions with evidence-level, constraint-level, and process-level criteria in agent evaluation and training.

Criterion Coupling and Aggregation. Another important issue is the coupling among rubric criteria. Existing studies have moved beyond coarse-grained overall preference scores toward more fine-grained, multidimensional evaluation. For example, FLASK (Ye et al., 2024) decomposes overall evaluation into different alignment skills and emphasizes that different task instances may require different compositions of capabilities, while Prometheus (Kim et al., 2024) supports fine-grained evaluation of long-form outputs with customized score rubrics and further demonstrates the potential of rubric-guided evaluators as reward models. These studies suggest that rubric criteria should not be treated as isolated checklist items; rather, they jointly define the evaluative structure of model outputs. For instance, in complex planning tasks, criteria such as satisfying explicit constraints, maintaining cross-turn state consistency, producing executable answers, and avoiding unwarranted assumptions are often tightly interdependent. A failure in state tracking may directly lead to constraint violations, while an oversimplified response may simultaneously reduce completeness and executability.

This coupling is particularly important for reinforcement learning, where the reward signal is not merely a linear sum of individual rubric scores but a structured signal shaped by multiple interacting objectives. MORLAIF (Williams, 2024) decomposes complex preferences into multiple principle-specific preference models, such as toxicity, factuality, and sycophancy, and combines them through scalarization to form reinforcement learning signals. This suggests that rubric criteria may be synergistic or conflicting: some criteria jointly define necessary conditions for a high-quality response, whereas others introduce trade-offs. If such coupling is ignored, RL training may encourage the model to optimize locally easy-to-satisfy criteria while sacrificing global task success, leading to reward hacking. Future work should therefore investigate how to learn dependency graphs among criteria, design nonlinear or hierarchical reward aggregation functions, distinguish hard constraints from soft preferences, and analyze how different rubric compositions affect generalization, training stability, and interpretability.

Training-Adaptive Rubric Evolution. Static rubrics, no matter how carefully engineered offline, are especially vulnerable during RL training. A rubric that is effective for judging the initial policy may become insufficient once the policy has learned to optimize against it. This creates a dynamic form of reward hacking: the model does not merely exploit a fixed reward function, but gradually discovers weaknesses in the rubric as training proceeds.

A natural mitigation, exemplified by works such as DR-Tulu (Shao et al., 2025), OnlineRubrics (Rezaei et al., 2025), SibylSense (Xu et al., 2026d), and RLAC (Wu et al., 2025b), is to let rubrics co-evolve with the policy. Under this perspective, rubrics are not static evaluation artifacts, but adaptive supervision mechanisms that can incorporate newly discovered failure modes into the criterion set. Such self-evolving rubrics may make the evaluation contract progressively stricter as the model improves, thereby reducing the chance that the policy repeatedly exploits outdated criteria. However, this direction also raises unresolved algorithmic questions: how often should rubrics be updated, how can update frequency be balanced against optimization stability, how can the rubric set avoid overfitting to the current policy distribution, and how to characterize the convergence of a joint policy–rubric optimization process. Designing principled training-adaptive rubric evolution mechanisms, rather than relying on ad-hoc heuristic updates, remains a critical open problem.

6.2 Generalization of Rubric-Based Reward Models

A complementary challenge concerns the generalization of rubric-based reward models. Rubrics have increasingly been embedded into reward modeling pipelines, either as explicit criteria used by judge models or as structured supervision for training dedicated reward models. Recent attempts—ranging from rubric-grounded reasoning models such as R3 (Anugraha et al., 2025b) and CDPRM (Liu et al., 2026a) to dimension-decomposed scorers such as ArmoRM (Wang et al., 2024a) and MR-RML (Jin et al., 2025)—show that rubrics can improve

the controllability and interpretability of reward modeling. Instead of producing only an opaque scalar score, such models can decompose evaluation into dimensions, expose intermediate judgments, or aggregate rubric-level scores into a final reward. This makes them attractive as reusable reward sources for policy optimization.

However, existing rubric-based reward models are still far from generally reliable. Fine-grained rubrics are often constructed for specific domains, tasks, or benchmarks, which limits their transferability to settings with different objectives, output formats, or evaluation standards. A reward model trained on rubric supervision from one domain may overfit to the surface form of those rubrics, the distribution of training responses, or the scoring habits of the judge used to generate supervision. When applied to new tasks, such a model may fail to adapt to new rubrics, misinterpret domain-specific requirements, or assign high scores to responses that satisfy familiar criteria while missing task-critical dimensions. This problem is especially severe in specialized domains such as medicine, law, finance, and scientific reasoning, where evaluation standards depend on expert knowledge and cannot be fully captured by generic helpfulness or correctness criteria.

The key open challenge is therefore to train rubric-based reward models that can generalize across tasks and domains while remaining faithful to the intended evaluation criteria. This involves several intertwined questions. First, what kind of rubric supervision is needed for cross-domain transfer: broad generic criteria, domain-specialized criteria, or compositional mixtures of both? Second, how should rubric information be represented inside the reward model: as natural-language criteria, dimension-level vectors, structured graphs, or latent preference factors? Third, how can we evaluate the reliability of a rubric-based reward model when no gold rubric exists for a new task? Recent meta-evaluation efforts such as RubricEval (Pan et al., 2026), RubricBench (Zhang et al., 2026a), and JudgeBench (Tan et al., 2025) mark important first steps, but a principled understanding of rubric reward model generalization is still missing.

One promising direction is to enable efficient adaptation with limited new supervision. Instead of requiring large-scale rubric annotations for every new domain, a rubric-based reward model could be recalibrated using a small number of task-specific rubric annotations or expert-written criteria. Such adaptation may help the model adjust to domain-specific standards, output formats, and failure modes while retaining general evaluation capabilities learned from broad rubric supervision. This is particularly important for specialized domains such as medicine, law, finance, and scientific reasoning, where fully annotating fine-grained rubrics is expensive but small amounts of expert feedback may be available.

Another important direction is to design modular or compositional rubric representations. Many evaluation dimensions, such as factuality, completeness, safety, evidence grounding, instruction following, and practical utility, recur across tasks but appear in different combinations and with different weights. If reward models can learn reusable rubric modules, then new task-specific rubrics may be constructed by selecting, recombining, and recalibrating these shared dimensions rather than retraining the model from scratch. This would allow reward models to transfer partially across tasks: general dimensions can be reused, while domain-specific dimensions can be added or adjusted with limited supervision. Such compositionality may also improve interpretability, because the reward model's behavior can be analyzed in terms of which rubric modules are activated and how they are weighted in a given task.

6.3 Bias in Rubric-Based Evaluation

Although rubric-based evaluation improves the interpretability of LLM evaluation, it does not guarantee unbiased judgments. Bias may be introduced by the rubric content, by the judge models that apply the rubrics, and by the human experts who define or validate the criteria. Understanding these sources of bias is important for making rubric-based evaluation more reliable and comparable across tasks, models, and evaluators.

Bias from Rubric Design. Bias in rubric-based evaluation can arise from how rubrics are written or presented to the judge model. On the one hand, LLM judges can be sensitive to the phrasing, ordering, and formatting of rubric criteria. Even when different expressions look equivalent to human evaluators, they may lead LLM judges to generate different judgments (Rao and Callison-Burch, 2026). On the other hand, rubric-based scoring may also suffer from position bias when multiple score options are presented (Wang et al., 2024c). For example, presenting the score levels as "Poor–Fair–Good–Excellent" may produce a different score

distribution from presenting them in the reverse order, even though the available options are semantically equivalent. Therefore, rubric design should not rely only on intuitive writing, but should involve systematic prompt design and empirical validation.

Bias from Judge-Model. Bias can also come from the judge model that applies the rubrics. **First**, different LLM judges may assign different scores to the same response under the same rubric, due to differences in model priors, training data, and understanding of rubrics. This makes rubric-based evaluation dependent not only on the rubric itself, but also on the selected judge model. **Second**, LLM judges may exhibit **self-preference bias** (Pombal et al., 2026), where they tend to give higher scores to outputs generated by themselves or models from the same family. This bias can persist even under seemingly objective rubrics and may yield unreliable model comparisons. **Third**, LLM judges may differ from human evaluators in how they weight rubric dimensions. Even when humans and LLMs use the same rubrics, they may focus on different aspects of response quality (Mittal et al., 2026). For example, in code evaluation, LLM judges may emphasize functional correctness or detailed explanations, while human developers may care more about context-aware solutions, maintainability, or whether the code naturally fits into the existing workflow. These observations suggest that, beyond improving rubric design, developing more reliable evaluation protocols and reducing bias across different LLM judges are important directions for future work.

Human-Expert Disagreement and Pluralistic Bias. A third source of bias comes from the human side of rubric construction and validation. Many rubric-based evaluations implicitly assume the existence of a single "golden standard" or unified ground truth. However, in highly specialized downstream domains, such as medicine, law, education, moral reasoning, and policy analysis, experts may reasonably disagree about what constitutes a high-quality answer. Different clinical schools may prioritize different diagnostic considerations; different legal frameworks may lead to different interpretations; and different educational philosophies may value different forms of explanation or scaffolding. In such cases, rubrics written only by one human expert may encode a legitimate but partial perspective, while appearing to be an objective standard.

This suggests that future rubric-based evaluation should move beyond the single-standard assumption and consider multi-perspective or pluralistic rubrics. Drawing inspiration from benchmarks such as MoReBench (Chiu et al., 2025), which emphasizes logical completeness and process coherence in pluralistic moral dilemmas, future rubrics may evaluate whether an output is internally coherent within a given theoretical or professional framework, rather than forcing all responses into a single preference ordering. More broadly, rubric-based evaluation may need mechanisms for representing expert disagreement, reporting score uncertainty, and distinguishing factual errors from perspective-dependent judgments. Developing such pluralistic rubric architectures is essential for robust domain-specific evaluation, especially in settings where professional diversity and normative disagreement are unavoidable.

6.4 Personalized Rubrics

Most existing rubric-based methods treat rubrics as general evaluation criteria shared across users. However, in many open-ended tasks, especially creative writing, recommendation, education, and assistant-style interaction, response quality is highly user-dependent. Different users may prefer different tones, levels of detail, writing styles, reasoning patterns, interaction goals, and even different trade-offs between accuracy, creativity, conciseness, and emotional resonance. Therefore, an important open question is how to construct **personalized rubrics** that explicitly capture individual preferences while remaining interpretable, reliable, and safe.

PREFINE (Ueda and Takayanagi, 2025) provides a useful example in personalized story generation. Instead of relying on direct user feedback or parameter updates, it constructs a pseudo-user agent from user history and generates user-specific rubrics to guide critique and refinement. This demonstrates that rubrics can serve not only as general evaluation standards, but also as an explicit interface for representing personalized preferences. In such a framework, user-specific rubrics can make personalization more transparent: rather than simply producing a personalized output, the system can expose what aspects of the user's preference it is optimizing for, such as preferred narrative style, emotional tone, character development, or thematic focus.

Despite this promise, personalized rubrics remain difficult to build and evaluate. User preferences are often

implicit, sparse, noisy, and context-dependent, making it unclear how much user history is needed to infer faithful rubrics. Moreover, preferences may change over time or vary across tasks, so a static user profile may quickly become outdated. Another risk is superficial personalization: a system may overfit to easily observable signals such as wording, genre, or length, while missing deeper preferences such as factual rigor, emotional nuance, pedagogical value, or practical usefulness. This is particularly problematic when personalized rubrics are used not only for evaluation but also for training, because models may learn to imitate shallow preference markers rather than genuinely adapt to user intent.

A further challenge lies in the tension between personalization and general quality or safety standards. Optimizing a user-specific rubric should not lead the model to reinforce biased, unsafe, manipulative, or low-quality preferences. Future work therefore needs mechanisms to validate whether personalized rubrics truly reflect user preferences, remain stable across contexts, and are compatible with broader alignment constraints. Promising directions include separating user-specific preferences from universal safety requirements, auditing personalized rubrics for harmful or biased criteria, modeling preference uncertainty, and allowing users to inspect or revise the rubrics that represent them.

6.5 Safety of Rubrics

As rubrics become widely used in LLM evaluation and training, their safety also becomes an important open question. Existing work often treats rubrics as transparent and trustworthy evaluation instructions, but recent evidence suggests that rubrics themselves can become an attack surface. Ding et al. (2026) identify *Rubric-Induced Preference Drift* (RIPD), where seemingly natural and benchmark-compliant rubric edits can systematically shift the preferences of LLM judges on target domains. This is especially concerning because such edits may not significantly reduce aggregate benchmark performance, yet they can still change the judge's preference direction and affect downstream alignment results.

This problem suggests that rubric safety should be studied beyond traditional evaluator robustness. A malicious or poorly designed rubric may subtly change criterion weights, introduce biased wording, emphasize some dimensions while downplaying others, or reshape the judge's decision boundary while still appearing reasonable to human inspection. For example, a rubric edit may preserve the same high-level topic but make the judge more tolerant of unsupported claims, more favorable toward a particular style, or less sensitive to safety-relevant omissions. Because natural-language rubrics are flexible and often manually refined, such preference shifts can be difficult to detect through standard benchmark validation alone.

The risk becomes more serious when rubric-based judgments are used as preference labels or reward signals. In this case, rubric-induced bias may propagate into policy model training and produce persistent behavioral drift. A judge that is biased by a rubric may generate biased preference data; a policy optimized on that data may then learn the shifted preference; and future evaluation may fail to detect the drift if it relies on similarly vulnerable rubric-based judges. Therefore, rubric safety should be understood as a pipeline-level problem rather than merely a prompt-level problem.

Future work needs mechanisms for detecting unsafe rubric edits, auditing preference shifts across domains, and validating rubric invariance under paraphrasing or refinement. One promising direction is to evaluate rubrics not only by aggregate benchmark scores but also by their directional effects on targeted preference axes. Another direction is to develop adversarial tests that perturb rubric wording, ordering, and emphasis to measure whether judge behavior remains stable. Ensuring the safety of rubrics is essential for making rubric-based evaluation and training reliable in high-stakes alignment pipelines.

7 Conclusion

As LLMs continue to move toward open-ended, high-stakes, and agentic applications, rubrics offer a practical way to express complex quality standards as explicit, interpretable, and operational criteria. This survey systematically reviewed rubric-based research from three perspectives: how rubrics are constructed, how they support the training of policy models and reward models, and how they are used for task evaluation across general and domain-specific tasks. Although recent studies have shown the value of rubrics in improving model training and evaluation, many challenges remain, such as reward hacking in rubric-based RL, generalization

of rubric-based reward models, and bias in rubric-based evaluation. Future work should therefore develop more reliable, adaptive, and trustworthy rubric systems that can better connect human expectations, task requirements, and model behavior.

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. *CoRR*, abs/2511.11562, 2025.

David Anugraha, Shou-Yi Hung, Zilu Tang, En-Shiun Annie Lee, Derry Tanti Wijaya, and Genta Indra Winata. mr3: Multilingual rubric-agnostic reward reasoning models. *CoRR*, abs/2510.01146, 2025a.

David Anugraha, Zilu Tang, Lester James V. Miranda, Hanyang Zhao, Mohammad Rifqi Farhansyah, Garry Kuwanto, Derry Wijaya, and Genta Indra Winata. R3: robust rubric-agnostic reward models. *CoRR*, abs/2505.13388, 2025b.

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. *CoRR*, abs/2505.08775, 2025.

Susan M Brookhart. Appropriate criteria: Key to effective rubrics. In *Frontiers in education*, volume 3, page 22. Frontiers Media SA, 2018.

Xin-Sheng Chen, Jiayu Zhu, Pei-lin Li, Hanzheng Wang, Shuojin Yang, and Menghao Guo. Presentbench: A fine-grained rubric-based benchmark for slide generation. *CoRR*, abs/2603.07244, 2026a.

Yukun Chen, Jiaming Li, Longze Chen, Ze Gong, Jingpeng Li, Zhen Qin, Hengyu Chang, Ancheng Xu, Zhihao Yang, Hamid Alinejad-Rokny, Qiang Qu, Bo Zheng, and Min Yang. Rucl: Stratified rubric-based curriculum learning for multimodal large language model reasoning. *CoRR*, abs/2602.21628, 2026b.

Yu Ying Chiu, Michael S. Lee, Rachel Calcott, Brandon Handoko, Paul de Font-Reaulx, Paula Rodriguez, Chen Bo Calvin Zhang, Ziwen Han, Udari Madhushani Sehwag, Yash Maurya, Christina Q. Knight, Harry R. Lloyd, Florence Bacus, Mantas Mazeika, Bing Liu, Yejin Choi, Mitchell L. Gordon, and Sydney Levine. Morebench: Evaluating procedural and pluralistic moral reasoning in language models, more than outcomes. *CoRR*, abs/2510.16380, 2025.

Yucheng Chu, Hang Li, Kaiqi Yang, Yasemin Copur-Gencturk, Joseph Krajcik, Namsoo Shin, and Jiliang Tang. Confusion-aware rubric optimization for llm-based automated grading. *CoRR*, abs/2603.00451, 2026.

Runpeng Dai, Linfeng Song, Haolin Liu, Zhenwen Liang, Dian Yu, Haitao Mi, Zhaopeng Tu, Rui Liu, Tong Zheng, Hongtu Zhu, and Dong Yu. CDE: curiosity-driven exploration for efficient reinforcement learning in large language models. *CoRR*, abs/2509.09675, 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

Kaustubh Deshpande, Ved Sirdeshmukh, Johannes Baptist Mols, Lifeng Jin, Ed-Yeremai Hernandez-Cardona, Dean Lee, Jeremy Kritz, Willow E. Primack, Summer Yue, and Chen Xing. Multichallenge: A realistic multi-turn conversation evaluation benchmark challenging to frontier llms. In *ACL (Findings)*, Findings of ACL, pages 18632–18702. Association for Computational Linguistics, 2025.

Kaustubh D. Dhole and Eugene Agichtein. RubricRAG: Towards interpretable and reliable llm evaluation via domain knowledge retrieval for rubric generation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2026.

Liang Ding. Adarubric: Task-adaptive rubrics for LLM agent evaluation. *CoRR*, abs/2603.21362, 2026.

Ruomeng Ding, Yifei Pang, He Sun, Yizhong Wang, Zhiwei Steven Wu, and Zhun Deng. Rubrics as an attack surface: Stealthy preference drift in LLM judges. *CoRR*, abs/2602.13576, 2026.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *CoRR*, abs/2505.16410, 2025a.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization. *CoRR*, abs/2507.19849, 2025b.

Guanting Dong, Junting Lu, Junjie Huang, Wanjun Zhong, Longxiang Liu, Shijue Huang, Zhenyu Li, Yang Zhao, Xiaoshuai Song, Xiaoxi Li, et al. Agent-world: Scaling real-world environment synthesis for evolving general agent intelligence. *arXiv preprint arXiv:2604.18292*, 2026.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents. *CoRR*, abs/2506.11763, 2025.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. In *ICML*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

Yuxuan Fan, Jing Hao, Hong Chen, Jiahao Bao, Yihua Shao, Yuci Liang, Kuo Feng Hung, and Hao Tang. Oralgpt-plus: Learning to use visual tools via reinforcement learning for panoramic x-ray analysis. *arXiv preprint arXiv:2603.06366*, 2026a.

Zhiting Fan, Ruizhe Chen, Tianxiang Hu, Ru Peng, Zenan Huang, Haokai Xu, Yixin Chen, Jian Wu, Junbo Zhao, and Zuozhu Liu. Optimsyn: Influence-guided rubrics optimization for synthetic data generation. *CoRR*, abs/2604.00536, 2026b.

Zhiyuan Fan, Weinong Wang, Xing Wu, and Debing Zhang. Sedareval: Automated evaluation using self-adaptive rubrics. In *EMNLP (Findings)*, Findings of ACL, pages 16916–16930. Association for Computational Linguistics, 2024.

Naghmeh Farzi and Laura Dietz. Pencils down! automatic rubric-based evaluation of retrieve/generate systems. In *ICTIR*, pages 175–184. ACM, 2024.

Shanghua Gao, Yuchang Su, Pengwei Sui, Curtis Ginder, and Marinka Zitnik. Qworld: Question-specific evaluation criteria for llms. *CoRR*, abs/2603.23522, 2026.

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. *CoRR*, abs/2512.23707, 2025.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. A survey on llm-as-a-judge. *CoRR*, abs/2411.15594, 2024.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on LLM-as-a-judge. *The Innovation*, 2025.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *CoRR*, abs/2507.17746, 2025.

Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Reward reasoning model. *CoRR*, abs/2505.14674, 2025.

Taneesh Gupta, Shivam Shandilya, Xuchao Zhang, Rahul Madhavan, Supriyo Ghosh, Chetan Bansal, Huaxiu Yao, and Saravan Rajmohan. CARMO: dynamic criteria generation for context aware reward modelling. In *ACL (Findings)*, Findings of ACL, pages 2202–2261. Association for Computational Linguistics, 2025.

Janghoon Han, Heegyu Kim, Changho Lee, Dahm Lee, Min Hyung Park, Hosung Song, Stanley Jungkyu Choi, Moontae Lee, and Honglak Lee. DEER: A comprehensive and reliable benchmark for deep-research expert reports. *CoRR*, abs/2512.17776, 2025.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. Llm-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *ACL (1)*, pages 13806–13834. Association for Computational Linguistics, 2024.

Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, Yue Xing, Jiliang Tang, and Benoît Dumoulin. Traject-bench:a trajectory-aware benchmark for evaluating agentic tool use. *CoRR*, abs/2510.04550, 2025a.

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. *CoRR*, abs/2511.10507, 2025b.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *COLING*, pages 6609–6625. International Committee on Computational Linguistics, 2020.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, Jianguo Li, and Junbo Zhao. Reinforcement learning with rubric anchors. *CoRR*, abs/2508.12790, 2025.

Maliheh Izadi, Jonathan Katzy, Tim van Dam, Marc Otten, Razvan Mihai Popescu, and Arie van Deursen. Language models for code completion: A practical evaluation. In *ICSE*, pages 79:1–79:13. ACM, 2024.

Mengzhao Jia, Zhihan Zhang, Ignacio Cases, Zheyuan Liu, Meng Jiang, and Peng Qi. Autorubric-r1v: Rubric-based generative rewards for faithful multimodal reasoning. *arXiv preprint arXiv:2510.14738*, 2025.

Ruipeng Jia, Yunyi Yang, Yuxin Wu, Yongbo Gai, Siyuan Tao, Mengyu Zhou, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Open rubric system: Scaling reinforcement learning with pairwise adaptive rubric. *CoRR*, abs/2602.14069, 2026.

Yuxuan Jiang and Francis Ferraro. SCRIBE: structured mid-level supervision for tool-using language models. *CoRR*, abs/2601.03555, 2026.

Yongnan Jin, Xurui Li, Feng Cao, Liucun Gao, and Juanjuan Yao. Multidimensional rubric-oriented reward model learning via geometric projection reference constraints. *CoRR*, abs/2511.16139, 2025.

Akira Kawabata and Saku Sugawara. C2: scalable rubric-augmented reward modeling from binary preferences. *CoRR*, abs/2604.13618, 2026.

Seungone Kim, Jamin Shin, Yejin Choi, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *ICLR*. OpenReview.net, 2024.

Zicheng Kong, Dehua Ma, Zhenbo Xu, Alven Yang, Yiwei Ru, Haoran Wang, Zixuan Zhou, Fuqing Bie, Liuyu Xiang, Huijia Wu, Jian Zhao, and Zhaofeng He. Omni-rrm: Advancing omni reward modeling via automatic rubric-grounded preference synthesis. *CoRR*, abs/2602.00846, 2026.

Mohammed Irfan Kurpath, Jaseel Muhammad Kaithakkodan, Jinxing Zhou, Sahal Shaji Mullappilly, Mohammad Almansoori, Noor Ahsan, Beknur Kalmakhanbet, Sambal Shikhar, Rishabh Lalla, Jean Lahoud, Mariette Awad, Fahad Shahbaz Khan, Salman H. Khan, Rao Muhammad Anwer, and Hisham Cholakkal. A benchmark and agentic framework for omni-modal reasoning and tool use in long videos. *CoRR*, abs/2512.16978, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124, 2024.

Guangchen Lan. Alternating reinforcement learning with contextual rubric rewards. *CoRR*, abs/2603.15646, 2026.

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

Ruizhe Li, Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench II: diagnosing deep research agents via rubrics from expert report. *CoRR*, abs/2601.08536, 2026a.

Shiyang Li, Zijian Zhang, Winson Chen, Yuebo Luo, Mingyi Hong, and Caiwen Ding. Stitchcuda: An automated multi-agents end-to-end gpu programing framework with rubric-based agentic reinforcement learning. *arXiv preprint arXiv:2603.02637*, 2026b.

Sunzhu Li, Jiale Zhao, Miteto Wei, Huimin Ren, Yang Zhou, Jingwen Yang, Shunyu Liu, Kaike Zhang, and Wei Chen. Rubrichub: A comprehensive and highly discriminative rubric dataset via automated coarse-to-fine generation. *CoRR*, abs/2601.08430, 2026c.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *CoRR*, abs/2504.21776, 2025.

Bill Yuchen Lin, Yuntian Deng, Khyathi Raghavi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *CoRR*, abs/2406.04770, 2024.

Dengcan Liu, Fengkai Yang, Xiaohan Wang, Shurui Yan, Jiajun Chai, Jiahao Li, Yikun Ban, Zhendong Mao, Wei Lin, and Guojun Yin. Cdrrm: Contrast-driven rubric generation for reliable and interpretable reward modeling. *arXiv preprint arXiv:2603.08035*, 2026a.

Rui Liu, Dian Yu, Zhenwen Liang, Yucheng Shi, Tong Zheng, Runpeng Dai, Haitao Mi, Pratap Tokekar, and Leoweiliang. Deltarubric: Generative multimodal reward modeling via joint planning and verification. *arXiv preprint arXiv:2605.09269*, 2026b.

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. *CoRR*, abs/2510.07743, 2025a.

Wenhan Liu, Xinyu Ma, Weiwei Sun, Yutao Zhu, Yuchen Li, Dawei Yin, and Zhicheng Dou. Reasonrank: Empowering passage ranking with strong reasoning ability. *CoRR*, abs/2508.07050, 2025b.

Wenhan Liu, Xinyu Ma, Yutao Zhu, Yuchen Li, Daiting Shi, Dawei Yin, and Zhicheng Dou. Agentic-r: Learning to retrieve for agentic search. *CoRR*, abs/2601.11888, 2026c.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2511–2522. Association for Computational Linguistics, 2023.

Yuan-Jay Lü, Chengyu Wang, Lei Shen, Jun Huang, and Tong Xu. Mock worlds, real skills: Building small agentic language models with synthetic tasks, simulated environments, and rubric-based rewards. *CoRR*, abs/2601.22511, 2026.

Ziyang Luo, Zhiqi Shen, Wenzhuo Yang, Zirui Zhao, Prathyusha Jwalapuram, Amrita Saha, Doyen Sahoo, Silvio Savarese, Caiming Xiong, and Junnan Li. Mcp-universe: Benchmarking large language models with real-world model context protocol servers. *CoRR*, abs/2508.14704, 2025.

Changze Lv, Jie Zhou, Wentao Zhao, Jingwen Xu, Zisu Huang, Muzhao Tian, Shihan Dou, Tao Gui, Le Tian, Xiao Zhou, Xiaoqing Zheng, Xuanjing Huang, and Jie Zhou. Learning query-specific rubrics from human preferences for deepresearch report generation. *CoRR*, abs/2602.03619, 2026.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn LLM agents. *CoRR*, abs/2401.13178, 2024.

Linyue Ma, Yilong Xu, Xiang Long, and Zhi Zheng. An efficient rubric-based generative verifier for search-augmented llms. *CoRR*, abs/2510.14660, 2025.

Hamed Mahdavi, Pouria Mahdavinia, Samira Malek, Pegah Mohammadipour, Alireza Hashemi, Majid Daliri, Alireza Farhadi, Amir Khasahmadi, Niloofar Mireshghallah, and Vasant G. Honavar. Refgrader: Automated grading of mathematical competition proofs using agentic workflows. *CoRR*, abs/2510.09021, 2025.

Anas Mahmoud, MohammadHossein Rezaei, Zihao Wang, Anisha Gunjal, Bing Liu, and Yunzhong He. Reward hacking in rubric-based reinforcement learning. *arXiv preprint arXiv:2605.12474*, 2026.

Aditya Mittal, Ryan Shar, Zichu Wu, Shyam Agarwal, Tongshuang Wu, Chris Donahue, Ameet Talwalkar, Wayne Chi, and Valerie Chen. Comparing developer and LLM biases in code evaluation. *CoRR*, abs/2603.24586, 2026.

Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. Rule based rewards for language model safety. In *NeurIPS*, 2024.

Minheng Ni, Zhengyuan Yang, Yaowen Zhang, Linjie Li, Chung-Ching Lin, Kevin Lin, Zhendong Wang, Xiaofei Wang, Shujie Liu, Lei Zhang, Wangmeng Zuo, and Lijuan Wang. Techimage-bench: Rubric-based evaluation for technical image generation, 2026.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155, 2022.

Tianjun Pan, Xuan Lin, Wenyan Yang, Qianyu He, Shisong Chen, Licai Qi, Wanqing Xu, Hongwei Feng, Bo Xu, and Yanghua Xiao. Rubriceval: A rubric-level meta-evaluation benchmark for LLM judges in instruction following. *CoRR*, abs/2603.25133, 2026.

Ernesto Panadero and Anders Jonsson. The use of scoring rubrics for formative assessment purposes revisited: A review. *Educational Research Review*, 9:129–144, 2013.

Aditya Kamlesh Parikh, Cristian Tejedor García, Catia Cucchiarini, and Helmer Strik. Rubric-guided fine-tuning of speechllms for multi-aspect, multi-rater L2 reading-speech assessment. *CoRR*, abs/2603.16889, 2026.

Shinwoo Park and Yo-Sub Han. From intuition to calibrated judgment: A rubric-based expert-panel study of human detection of llm-generated korean text, 2026.

Aditya Pathak, Rachit Gandhi, Vaibhav Uttam, Arnav Ramamoorthy, Pratyush Ghosh, Aaryan Raj Jindal, Shreyash Verma, Aditya Mittal, Aashna Ased, Chirag Khatri, Yashwanth Nakka, Devansh, Jagat Sesh Challa, and Dhruv Kumar. Rubric is all you need: Improving llm-based code evaluation with question-specific rubrics. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V.1*, ICER '25, pages 181–195. ACM, 2025.

José Pombal, Ricardo Rei, and André F. T. Martins. Self-preference bias in rubric-based evaluation of large language models. *CoRR*, abs/2604.06996, 2026.

Zhengyang Qi, Charles Dickens, Derek Pham, Amanda Dsouza, Armin Parchami, Frederic Sala, and Paroma Varma. RIFT: A rubric failure mode taxonomy and automated diagnostics. *CoRR*, abs/2604.01375, 2026.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. Infobench: Evaluating instruction following ability in large language models. In *ACL (Findings)*, Findings of ACL, pages 13025–13048. Association for Computational Linguistics, 2024.

Weijie Qiu, Dai Guan, Junxin Wang, Zhihang Li, Yongbo Gai, Mengyu Zhou, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Rationale matters: Learning transferable rubrics via proxy-guided critique for VLM reward models. *CoRR*, abs/2603.16600, 2026.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, Junran Peng, Zhaoxiang Zhang, Songyang Zhang, and Kai Chen. Hellobench: Evaluating long text generation capabilities of large language models. *CoRR*, abs/2409.16191, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Delip Rao and Chris Callison-Burch. Autorubric: A unified framework for rubric-based llm evaluation. *arXiv preprint arXiv:2603.00077*, 2026.

MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Bing Liu, Yunzhong He, and Afra Feyza Akyürek. Online rubrics elicitation from pairwise comparisons. *CoRR*, abs/2510.07284, 2025.

Kate Sanders, Nathaniel Weir, Sapana Chaudhary, Kaj Bostrom, and Huzefa Rangwala. Generating data-driven reasoning rubrics for domain-adaptive reward modeling. *CoRR*, abs/2602.06795, 2026.

Tim Schopf and Michael Färber. Is this idea novel? an automated benchmark for judgment of research ideas. *CoRR*, abs/2603.10303, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G. Finlayson, David A. Sontag, Tyler Murray, Sewon Min, Pradeep Dasigi, Luca Soldaini, Faeze Brahman, Wen-tau Yih, Tongshuang Wu, Luke Zettlemoyer, Yoon Kim, Hannaneh Hajishirzi, and Pang Wei Koh. DR tulu: Reinforcement learning with evolving rubrics for deep research. *CoRR*, abs/2511.19399, 2025.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. Assisting in writing Wikipedia-like articles from scratch with large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6252–6278, Mexico City, Mexico, June 2024a. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024b.

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. *CoRR*, abs/2511.07685, 2025.

William F. Shen, Xinchi Qiu, Chenxi Whitehouse, Lisa Alazraki, Shashwat Goel, Francesco Barbieri, Timon Willi, Akhil Mathur, and Ilias Leontiadis. Rethinking rubric generation for improving LLM judge and reward modeling for open-ended tasks. *CoRR*, abs/2602.05125, 2026.

Leheng Sheng, Wenchang Ma, Ruixin Hong, Xiang Wang, An Zhang, and Tat-Seng Chua. Reinforcing chain-of-thought reasoning with self-evolving rubrics. *CoRR*, abs/2602.10885, 2026.

Zhengliang Shi, Yiqun Chen, Haitao Li, Weiwei Sun, Shiyu Ni, Yougang Lyu, Run-Ze Fan, Bowen Jin, Yixuan Weng, Minjun Zhu, Qiujie Xie, Xinyu Guo, Qu Yang, Jiayi Wu, Jujia Zhao, Xiaqiang Tang, Xinbei Ma, Cunxiang Wang, Jiaxin Mao, Qingyao Ai, Jen-tse Huang, Wenxuan Wang, Yue Zhang, Yiming Yang, Zhaopeng Tu, and Zhaochun Ren. Deep research: A systematic survey. *CoRR*, abs/2512.02038, 2025.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A strongreject for empty jailbreaks. In *NeurIPS*, 2024.

Pragya Srivastava, Harman Singh, Rahul Madhavan, Gandharv Patil, Sravanti Addepalli, Arun Sai Suggala, Rengarajan Aravamudhan, Soumya Sharma, Anirban Laha, Aravindan Raghuveer, Karthikeyan Shanmugam, and doina Precup. Robust reward modeling via causal rubrics. *CoRR*, abs/2506.16507, 2025.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. Paperbench: Evaluating ai's ability to replicate AI research. In *ICML*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Yuan Tang, Alejandro Cuadron, Chenguang Wang, Raluca A. Popa, and Ion Stoica. Judgebench: A benchmark for evaluating llm-based judges. In *ICLR*. OpenReview.net, 2025.

Zelin Tan, Zhouliang Yu, Bohan Lin, Zijie Geng, Hejia Geng, Yudong Zhang, Mulei Zhang, Yang Chen, Shuyue Hu, Zhenfei Yin, Chen Zhang, and Lei Bai. Stabilizing rubric integration training via decoupled advantage normalization. *CoRR*, abs/2603.26535, 2026.

Kentaro Ueda and Takehiro Takayanagi. PREFINE: personalized story generation via simulated user critics and user-specific rubric generation. *CoRR*, abs/2510.21721, 2025.

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. *CoRR*, abs/2507.18624, 2025.

Binghai Wang, Yantao Liu, Yuxuan Liu, Tianyi Tang, Shenzhi Wang, Chang Gao, Chujie Zheng, Yichang Zhang, Le Yu, Shixuan Liu, Tao Gui, Qi Zhang, Xuanjing Huang, Bowen Yu, Fei Huang, and Junyang Lin. Outcome accuracy is not enough: Aligning the reasoning process of reward models. *CoRR*, abs/2602.04649, 2026a.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *EMNLP (Findings)*, Findings of ACL, pages 10582–10592. Association for Computational Linguistics, 2024a.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *ACL (1)*, pages 9440–9450. Association for Computational Linguistics, 2024b.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *ACL (1)*, pages 9440–9450. Association for Computational Linguistics, 2024c.

Pengkai Wang, Qi Zuo, Pengwei Liu, Zhijie Sang, Congkai Xie, and Hongxia Yang. Infimed-orbit: Aligning llms on open-ended complex tasks via rubric-based incremental training. *CoRR*, abs/2510.15859, 2025a.

Xingyao Wang, Valerie Chen, Heng Ji, and Graham Neubig. A rubric-supervised critic from sparse real-world outcomes. *CoRR*, abs/2603.03800, 2026b.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. *CoRR*, abs/2504.20571, 2025b.

Zhilin Wang, Jaehun Jung, Ximing Lu, Shizhe Diao, Ellie Evans, Jiaqi Zeng, Pavlo Molchanov, Yejin Choi, Jan Kautz, and Yi Dong. Profbench: Multi-domain rubrics requiring professional knowledge to answer and judge. *CoRR*, abs/2510.18941, 2025c.

Marcus Williams. Multi-objective reinforcement learning from ai feedback. *arXiv preprint arXiv:2406.07295*, 2024.

Genta Indra Winata, David Anugraha, Emmy Liu, Alham Fikri Aji, Shou-Yi Hung, Aditya Parashar, Patrick Amadeus Irawan, Ruochen Zhang, Zheng-Xin Yong, Jan Christian Blaise Cruz, Niklas Muennighoff, Seungone Kim, Hanyang Zhao, Sudipta Kar, Kezia Erina Suryoraharjo, Muhammad Farid Adilazuarda, En-Shiun Annie Lee, Ayu Purwarianti, Derry Tanti Wijaya, and Monojit Choudhury. Datasheets aren't enough: Datarubrics for automated quality metrics and accountability. *CoRR*, abs/2506.01789, 2025.

Fang Wu, Weihao Xuan, Ximing Lu, Zaïd Harchaoui, and Yejin Choi. The invisible leash: Why RLVR may not escape its origin. *CoRR*, abs/2507.14843, 2025a.

Mian Wu, Gavin Zhang, Sewon Min, Sergey Levine, and Aviral Kumar. RLAC: reinforcement learning with adversarial critic for free-form generation tasks. *CoRR*, abs/2511.01758, 2025b.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. Writingbench: A comprehensive benchmark for generative writing. *CoRR*, abs/2503.05244, 2025c.

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. *CoRR*, abs/2510.17314, 2025.

Zidi Xiu, David Q. Sun, Kevin Cheng, Maitrik Patel, Josh Date, Yizhe Zhang, Jiarui Lu, Omar Attia, Raviteja Vemulapalli, Oncel Tuzel, Meng Cao, and Samy Bengio. Astra-bench: Evaluating tool-use agent reasoning and action planning with personal user context. *CoRR*, abs/2603.01357, 2026.

Ran Xu, Tianci Liu, Zihan Dong, Tony Yu, Ilgee Hong, Carl Yang, Linjun Zhang, Tao Zhao, and Haoyu Wang. Alternating reinforcement learning for rubric-based reward modeling in non-verifiable LLM post-training. *CoRR*, abs/2602.01511, 2026a.

Tianze Xu, Yanzhao Zheng, Pengrui Lu, Lyumanshan Ye, Yong Wu, ZhenTao Zhang, YuanQiang Yu, Chao Ma, JiHuai Zhu, Pengfei Liu, Baohua Dong, Hangcheng Zhu, Ruohui Huang, and Gang Yu. Rubrics to tokens: Bridging response-level rubrics and token-level rewards in instruction following tasks. *CoRR*, abs/2604.02795, 2026b.

Yifei Xu, Tusher Chakraborty, Srinagesh Sharma, Leonardo Nunes, Swati Sharma, Kate Drakos Demopulos, Emre Kıcıman, Songwu Lu, and Ranveer Chandra. Direct reasoning optimization: Token-level reasoning reflectivity meets rubric gates for unverifiable tasks, 2026c.

Yifei Xu, Guilherme Potje, Shivam Shandilya, Tiancheng Yuan, Leonardo O. Nunes, Rakshanda Agarwal, Saeid Asgari, Adam Atkinson, Emre Kiciman, Songwu Lu, Ranveer Chandra, and Tusher Chakraborty. Sibylsense: Adaptive rubric learning via memory tuning and adversarial probing. *CoRR*, abs/2602.20751, 2026d.

Lin Yang, Yuancheng Yang, Xu Wang, Changkun Liu, and Haihua Yang. Medmt-bench: Can llms memorize and understand long multi-turn conversations in medical scenarios? *CoRR*, abs/2603.23519, 2026a.

Qianyu Yang, Yang Liu, Jiaqi Li, Jun Bai, Hao Chen, Kaiyuan Chen, Tiliang Duan, Jiayun Dong, Xiaobo Hu, Zixia Jia, Yang Liu, Tao Peng, Yixin Ren, Ran Tian, Zaiyuan Wang, Yanglihong Xiao, Gang Yao, Lingyue Yin, Ge Zhang, Chun Zhang, Jianpeng Jiao, Zilong Zheng, and Yuan Gong. $onemillion-bench: How far are language agents from human experts? *CoRR*, abs/2603.07980, 2026b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380. Association for Computational Linguistics, 2018.

Fangda Ye, Yuxin Hu, Pengxiang Zhu, Yibo Li, Ziqi Jin, Yao Xiao, Yibo Wang, Lei Wang, Zhen Zhang, Lu Wang, Yue Deng, Bin Wang, Yifan Zhang, Liangcai Su, Xinyu Wang, He Zhao, Chen Wei, Qiang Ren, Bryan Hooi, An Bo, Shuicheng Yan, and Lidong Bing. Miroeval: Benchmarking multimodal deep research agents in process and outcome. *CoRR*, abs/2603.28407, 2026.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. FLASK: fine-grained language model evaluation based on alignment skill sets. In *ICLR*. OpenReview.net, 2024.

Haneul Yoo, Jieun Han, So-Yeon Ahn, and Alice Oh. Dress: Dataset for rubric-based essay scoring on EFL writing. In *ACL (1)*, pages 13439–13454. Association for Computational Linguistics, 2025.

Jiachen Yu, Zhihao Xu, Junjie Wang, and Yujiu Yang. Think-with-rubrics: From external evaluator to internal reasoning guidance. *arXiv preprint arXiv:2605.07461*, 2026a.

Ya-Qi Yu, Fangyu Hong, Xiangyang Qu, Hao Wang, Gaojie Wu, Qiaoyu Luo, Nuo Xu, Huixin Wang, Wuheng Xu, Yongxin Liao, Zihao Chen, Haonan Li, Ziming Li, Dezhi Peng, Minghui Liao, Jihao Wu, Haoyu Ren, and Dandan Tu. Visual preference optimization with rubric rewards. *CoRR*, abs/2604.13029, 2026b.

Youliang Yuan, Qiuyang Mang, Jingbang Chen, Hong Wan, Xiaoyuan Liu, Junjielong Xu, Jen-tse Huang, Wenxuan Wang, Wenxiang Jiao, and Pinjia He. Curing miracle steps in LLM mathematical reasoning with rubric rewards. *CoRR*, abs/2510.07774, 2025.

Junkai Zhang, Zihao Wang, Lin Gui, Swarnashree Mysore Sathyendra, Jaehwan Jeong, Victor Veitch, Wei Wang, Yunzhong He, Bing Liu, and Lifeng Jin. Chasing the tail: Effective rubric-based reward modeling for large language model post-training. *CoRR*, abs/2509.21500, 2025.

Qiyuan Zhang, Junyi Zhou, Yufei Wang, Fuyuan Lyu, Yidong Ming, Can Xu, Qingfeng Sun, Kai Zheng, Peng Kang, Xue Liu, and Chen Ma. Rubricbench: Aligning model-generated rubrics with human standards. *CoRR*, abs/2603.01562, 2026a.

Wenjian Zhang, Kongcheng Zhang, Jiaxin Qi, Baisheng Lai, and Jianqiang Huang. Experience is the best teacher: Motivating effective exploration in reinforcement learning for llms. *CoRR*, abs/2603.20046, 2026b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS*, 2023.

Jialun Zhong, Wei Shen, Yanzeng Li, Songyang Gao, Hua Lu, Yicheng Chen, Yang Zhang, Wei Zhou, Jinjie Gu, and Lei Zou. A comprehensive survey of reward models: Taxonomy, applications, challenges, and future. *CoRR*, abs/2504.12328, 2025.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Jiale Zhao, Jingwen Yang, Jianwei Lv, Kongcheng Zhang, Yihe Zhou, Hengtong Lu, Wei Chen, Yan Xie, and Mingli Song. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general LLM reasoning. *CoRR*, abs/2508.16949, 2025.

Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek F. Wong, Xiaowei Huang, Qiufeng Wang, and Kaizhu Huang. Is your model really A good math reasoner? evaluating mathematical reasoning with checklist. *CoRR*, abs/2407.08733, 2024.

Yutao Zhu, Xingshuo Zhang, Maosen Zhang, Jiajie Jin, Liancheng Zhang, Xiaoshuai Song, Kangzhi Zhao, Wencong Zeng, Ruiming Tang, Han Li, Ji-Rong Wen, and Zhicheng Dou. GISA: A benchmark for general information-seeking assistant. *CoRR*, abs/2602.08543, 2026.