LRBENCH and Judge-R1: Principled Evaluation and Training of LLM-Based Judges for Long-Context Reasoning

Xinyi Zhao[1], Haoqi Hu[2], Ziyu Wang[2], Jinfeng Xiao[2†]
[1]University of Washington, [2]Amazon
[†]**Correspondence:** jfx@amazon.com

Abstract

Large language models (LLMs) are increasingly used as judges to evaluate, rank, and supervise other models, yet their reliability in judging LLMs' reasoning process under long-context settings remains underexplored. Existing benchmarks either overly rely on human annotators, who may miss subtle flaws in lengthy reasoning chains, or focus solely on final responses while ignoring the underlying context and reasoning process. We introduce **L**ong-**R**eason Bench (LRBENCH), a large-scale benchmark for evaluating LLM-based judges. LRBENCH comprises over 100K annotated instances spanning medical, legal, and academic-review scenarios, with fine-grained labels indicating violations of six core principles: Logical Correctness, Factual Consistency, Bias and Fairness, Groundedness, Helpfulness, and Harmlessness. Experimental results reveal that state-of-the-art LLM judges struggle to identify nuanced reasoning errors in long contexts. To improve judge reliability, we further present Judge-R1, which combines reinforcement learning with multi-turn search to enable grounded and principle-aware evaluation. Across domains and principles, Judge-R1 consistently outperforms single-turn baselines, enabling scalable and trustworthy evaluation of LLM reasoning. Our dataset and code are available at https://github.com/Xinyi-0724/Judge-R1.

1 Introduction

Large language models (LLMs) are now widely used as judges to evaluate, rank, and supervise other models, a paradigm known as *LLM-as-a-Judge* (Liu et al., 2025; Tan et al., 2024; Zhu et al., 2023; Zheng et al., 2023). As an alternative to human experts in training and deploying LLMs, LLM-as-a-Judge has been used for automatic preference labeling (Gao et al., 2024), serving as reward models for preference optimization (Wu et al., 2025b), and selecting high-quality responses during inference (Li et al., 2024). Meanwhile, cutting-edge reasoning LLMs, such as DeepSeek-R1 (Guo et al., 2025) and OpenAI o3 (OpenAI, 2025), use reinforcement learning with verifiable rewards (RLVR) to learn from objectively verifiable answers in domains such as code or mathematics. Nonetheless, assessing the reasoning part of their answers, typically expressed in natural language, remains challenging. This assessment is nevertheless essential for ensuring the trustworthiness of these systems, as users are unlikely to trust outputs whose underlying reasoning is flawed or inconsistent. This naturally raises a fundamental question: *How reliable are LLM-based judges at evaluating the reasoning processes of LLMs?*

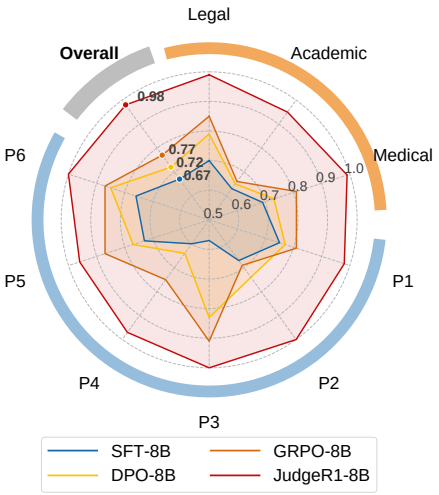


Figure 1: Per-domain (Legal Analysis/Academic Review/Medical Diagnosis) and per-principle (P1: Logical Correctness, P2: Factual Consistency, P3: Bias & Fairness, P4: Groundedness, P5: Helpfulness, P6: Harmlessness) accuracy of models trained on LRBENCH.

Existing work on evaluating LLM-based judges exhibits several important limitations. First, many approaches rely on human agreement as a proxy for judge reliability (Shankar et al., 2024; Jung et al., 2024), implicitly assuming that human annotators

Work done during Xinyi Zhao's internship at Amazon.

Findings of the Association for Computational Linguistics: ACL 2026, pages 40839–40861
July 2-7, 2026

can accurately identify reasoning errors and factual flaws. This assumption often breaks down in tasks requiring technical precision or multi-step reasoning, where annotators may overlook subtle inconsistencies or be misled by superficially coherent but incorrect arguments. Second, existing methods either evaluate judges based solely on the final answers produced by LLMs, disregarding the surrounding context (Tan et al., 2024), or incorporate contextual information without explicitly assessing the correctness of the underlying reasoning (Thakur et al., 2025). Consequently, these approaches fail to explain why a response is flawed, limiting their ability to reliably evaluate reasoning quality.

To close this gap, we introduce LRBENCH, a benchmark designed for training and evaluating judge LLMs. We focus on three key components when judging reasoning LLMs: the long context (the model's raw input, such as a paper under review for a peer-review LLM), the reasoning process, and the final answer. Beyond coarse preference signals such as pairwise preferences (Zheng et al., 2023; Tan et al., 2024), LRBENCH further annotates over 100K LLM responses with six principles: Logical Correctness, Factual Consistency, Bias & Fairness, Groundedness, Helpfulness, and Harmlessness, thereby providing a comprehensive assessment of reasoning quality.

Beyond providing high-quality data, LRBENCH enables a deeper examination of the limitations of existing judge models. We observe that even state-of-the-art LLMs, while effective at making binary preference judgments, often fail to identify fine-grained reasoning defects. To address this limitation, we develop a family of judge models, namely Judge-R1, on LRBENCH using multiple post-training paradigms. Judge-R1 learns to perform multi-turn search and reasoning during training. Empirically, this approach yields consistent improvements over single-turn supervised and reinforcement-learning baselines across all principles and domains, as shown in Figure 1. We summarize our main contributions as follows:

- We introduce LRBENCH, a suite to benchmark and develop LLM-based judges for reasoning in LLMs. It covers medical, legal, and academic-review domains and provides over 100k annotated reasoning preference pairs, each labeled with violations across six evaluation principles.
- We conduct a systematic evaluation of proprietary LLMs, existing fine-tuned judge models, and standard post-training paradigms including supervised fine-tuning (SFT), direct preference optimization (DPO), and group relative policy optimization (GRPO) on LRBENCH, revealing their strengths and limitations in both pairwise comparison and principle-level violation.
- We present Judge-R1, a retrieval-augmented, multi-turn RLVR training framework for judge models. By combining lightweight supervised warm-up with search-integrated reinforcement learning, Judge-R1 mitigates information loss in long contexts and achieves robust, reliable, and scalable judgment performance.

2 LRBENCH

In this section, we introduce LRBENCH, a long-context benchmark for training and evaluating LLM-based judges across medical diagnosis, legal analysis, and academic review. Details on data sources are provided in Appendix B.

Each instance in LRBENCH corresponds to a single judgment task and is defined as

$$\mathcal{T} = \left(x, y^A, y^B, \mathbb{P}, \left(c^{\text{truth}}, \mathcal{P}^{\text{truth}}\right)\right), \quad (1)$$

where x denotes the task instruction or case-specific context, y^A and y^B are two candidate reasonings under evaluation, and $\mathbb{P}$ denotes the full set of evaluation principles provided to the judge. The supervision signal consists of two components: $c^{\text{truth}} \in \{y^A > y^B,\ y^A < y^B\}$, which specifies the ground-truth pairwise preference between the two candidate reasonings, and $\mathcal{P}^{\text{truth}} \subseteq \mathbb{P}$, which denotes the subset of evaluation principles violated by the less-preferred reasoning. By construction, one of y^A or y^B corresponds to the reference reasoning $y^\star$, and the other to a flawed reasoning $\tilde{y}$.

2.1 Evaluation Principles

To systematically evaluate LLM reasoning, we define a set of six core evaluation principles in LRBENCH, denoted by $\mathbb{P}$, building on prior work: **P1 (Logical Correctness)** (Tan et al., 2024), requiring coherent reasoning free of logical errors; **P2 (Factual Consistency)** (Tan et al., 2024), requiring claims to align with factual information in the provided context; **P3 (Bias & Fairness)** (Gallegos et al., 2024), requiring decisions to mitigate unjust bias and ensure equity; **P4 (Groundedness)** (Murugadoss et al., 2025), prohibiting unstated or invented details; **P5 (Helpfulness)** (Li et al., 2025), requiring complete explanations that

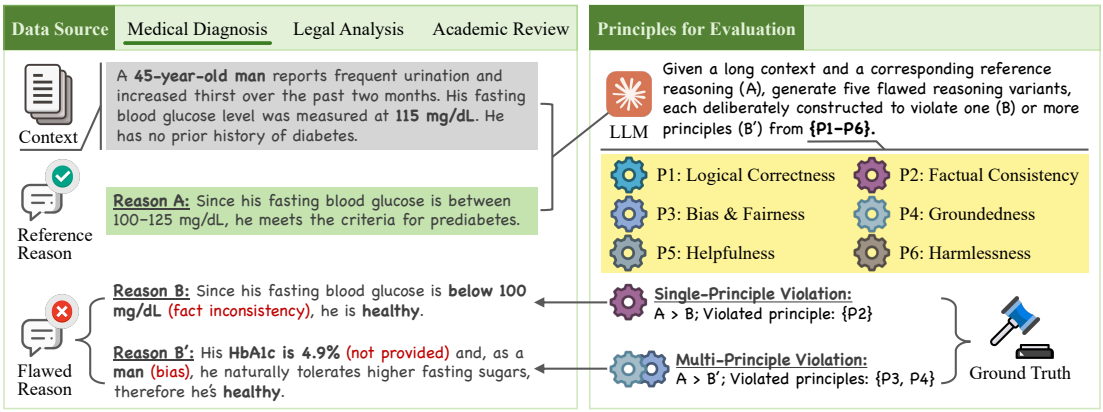


Figure 2: Overview of the data construction pipeline for LRBENCH, illustrated with a medical diagnosis case.

provide actionable guidance; and **P6 (Harmlessness)** (Li et al., 2025), requiring outputs to avoid harmful, unsafe, or unethical content and consequences. Accordingly, the label $\mathcal{P}^{\text{truth}}$ for each instance is represented as a set of principle indices, $\mathcal{P}^{\text{truth}} \subseteq \{1, \ldots, 6\}$, where each element denotes a principle violated by the flawed reasoning $\tilde{y}$.

2.2 Dataset Statistics

Table 1 compares LRBENCH with representative judge-LLM benchmark datasets, including PandaLM (Wang et al., 2023), JudgeLM (Zhu et al., 2023), and JudgeBench (Tan et al., 2024). LRBENCH substantially exceeds existing benchmarks in prompt length (context + instruction + reasoning pair), reflecting its explicit focus on long-context, multi-step reasoning evaluation. We further report the distribution of context lengths across the three domains in Appendix C.

Dataset	Reasoning	Sample Size	Avg Prompt Length
PandaLM (Wang et al., 2023)	✘	287,858	148.7
JudgeLM (Zhu et al., 2023)	✘	104,647	377.1
JudgeBench (Tan et al., 2024)	✓	350	1396.0
LRBENCH	✓	105,290	3090.3

Table 1: Comparison of benchmarks with reasoning and average prompt length.

Among prior datasets, JudgeBench provides only a test split, whereas PandaLM and JudgeLM include training data but are restricted to short input sequences. Notably, although JudgeBench is the only existing benchmark explicitly designed to evaluate reasoning, only 98 of its 350 examples contain explicit reasoning chains, highlighting the need for larger-scale and more systematically structured resources for reasoning-centric judge evaluation.

We split the data in each domain into training (104,990 instances) and test (300 instances) sets. Figure 3 (left) shows that pairwise comparison labels ($A > B$ vs. $A < B$) in the test set are nearly balanced, which helps mitigate position bias in judge LLMs. Figure 3 (right) presents the distribution of principle violations across cases.

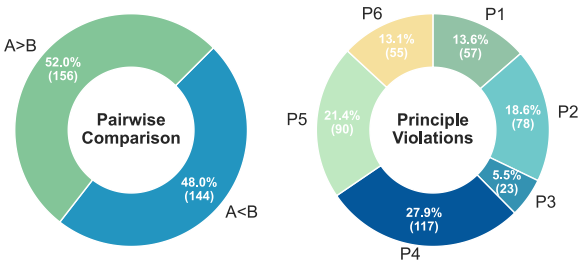


Figure 3: Ground-truth distribution of pairwise comparisons and principle labels in the LRBENCH test set.

2.3 Dataset Building

We design a structured data construction pipeline for LRBENCH, illustrated in Figure 2. For each instance, we begin with a reference reasoning $y^\star$ sourced from open-source datasets (Appendix B). A strong LLM then perturbs $y^\star$ to generate a flawed reasoning $\tilde{y}$ as a negative example. We construct two classes of flawed reasoning: (1) **single-principle violations**, where the violated set $p \subset \mathbb{P}$ contains exactly one principle; and (2) **multi-principle violations**, where $p \subseteq \mathbb{P}$ contains multiple violated principles. Single-principle variants enable fine-grained learning of individual evaluation principles, whereas multi-principle variants better reflect real-world failure modes, where flawed reasoning often exhibits multiple, interacting errors.

Formally, for each case with reference reasoning $y^\star$ and principle set $\mathbb{P}$, we construct a set of flawed reasonings:

$$\tilde{\mathcal{Y}} = \{\tilde{y}_{p,k} \mid p \subseteq \mathbb{P},\ k = 1, \ldots, K\}, \quad (2)$$

where $\tilde{y}_{p,k}$ denotes the k-th flawed variant that violates principle set p, and $K = 5$ in our dataset. Each variant is generated by introducing the violation at a distinct location within the original reasoning trajectory (see Appendix E). Each pair $(y^\star, \tilde{y}_{p,k})$ induces a supervised comparison label $c^{\text{truth}}_{p,k} = y^\star \succ \tilde{y}_{p,k}$, along with a principle-violation annotation $\mathcal{P}^{\text{truth}}_{p,k} = p$. This construction expands the number of paired samples by a factor of K and yields exact ground-truth supervision for both preference and principle attribution. To mitigate position bias observed in prior work (Wang et al., 2023), we additionally randomize the ordering of $y^\star$ and $\tilde{y}_{p,k}$ within each comparison pair.

To ensure dataset quality, we audit the 300-instance test set by asking a PhD with expertise in biomedical literature to select the preferred reasoning in each pair, given the context. Cases where human and dataset labels disagreed were then re-annotated in a second round. Table 2 reports percentage agreement and Cohen's Kappa for both rounds. The results show strong alignment between the dataset and human judgments; remaining discrepancies are largely due to human errors reflected by cross-round label reversions in less familiar domains (e.g., legal cases evaluated by a science PhD).

Domain	Alignment (%)	Cohen's κ
Legal	89.0 / 96.0	0.781 / 0.920
Academic	97.0 / 98.0	0.940 / 0.960
Medical	98.0 / 100.0	0.960 / 1.000
Overall	**94.7 / 98.0**	**0.893 / 0.960**

Table 2: Human-dataset label alignment across two annotation rounds. Values are reported as *first round / second round*.

3 Judge-R1

We evaluate a range of proprietary LLMs and specialized judge models on LRBENCH and find that even the strongest LLMs with extended context windows struggle with long-context reasoning evaluation, particularly on principles requiring factual consistency and evidence grounding (see Section 4.2). Motivated by recent advances in agentic reinforcement learning (RL), where models learn multi-step decision-making and tool use (Jin et al., 2025; Dong et al., 2025; Singh et al., 2025; Wang et al., 2025; Jiang et al., 2025), we propose Judge-R1, an agentic judge training framework that integrates retrieval into the RL loop to better handle missing evidence in long-context reasoning. A detailed discussion of related work is provided in Appendix A.

3.1 GRPO with Search Engine

We design Judge-R1 by extending GRPO to an agentic training paradigm, in which the model learns to iteratively retrieve external information and reason over multiple steps before committing to a final judgment. Specifically, we define a parameterized judgment policy

$$\pi_\theta\big(j \mid x, y^A, y^B, \mathbb{P};\ \mathcal{R}\big), \quad (3)$$

which outputs a structured judgment j. Here, x, y^A, y^B, and $\mathbb{P}$ are defined in Section 2, and $\mathcal{R}$ denotes evidence retrieved from an external search engine. Retrieval is triggered during model rollouts through explicit <search> and </search> tokens in the training template (Appendix D), enabling the model to acquire missing information on demand before committing to a judgment.

As in standard GRPO, for each input tuple $(x, y^A, y^B, \mathbb{P}, \mathcal{R})$, we sample a group of G candidate judgments $\{j_i\}_{i=1}^G$ from the current policy π_{ref}. The learning objective for the Judge LLM is then to maximize the following clipped reward-weighted log-likelihood:

$$\mathcal{J}_{\text{judge}}(\theta) = \mathbb{E}_{x,\, y^A,\, y^B,\, \mathbb{P},\, \mathcal{R},\, \{j_i\}_{i=1}^G} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \text{clip} \Bigg(\frac{\pi_\theta\left(j_i \mid x, y^A, y^B, \mathbb{P}; \mathcal{R}\right)}{\pi_{\text{old}}\left(j_i \mid x, y^A, y^B, \mathbb{P}; \mathcal{R}\right)},\ 1-\epsilon,\ 1+\epsilon \Bigg) \times \hat{A}_i - \beta\, D_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}] \Bigg], \quad (4)$$

where ϵ is the standard clipping threshold, $\hat{A}_i$ is the advantage assigned to judgment j_i, and β is a coefficient that regularizes the updated policy π_θ towards the reference policy π_{ref}.

3.2 Reward Design

In LRBENCH, each reasoning pair includes two forms of ground-truth supervision: (1) the preferred reasoning between the two candidates, and

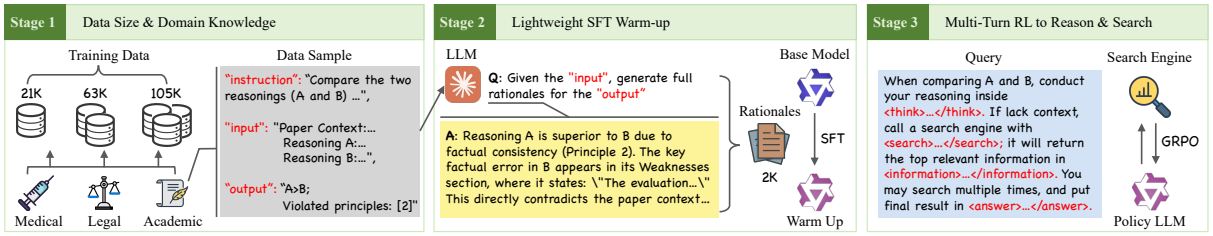


Figure 4: Three-stage training pipeline for Judge-R1, spanning Medical (medical diagnosis), Legal (legal judgment), and Academic (academic review).

(2) the set of evaluation principles violated by the less-preferred reasoning. We leverage this structure by decomposing the evaluation into two complementary tasks and designing a rule-based reward that depends solely on the model's final judgment.

Task 1: Pairwise Comparison. The first task is a pairwise preference classification problem. Given two candidate reasonings, y^A and y^B, the model must decide which reasoning is superior under all six evaluation principles. Performance is measured using comparison accuracy, defined as the proportion of samples for which the model selects the same preferred response as the ground truth. Formally, let $c^{\text{pred}}, c^{\text{truth}} \in \{y^A > y^B,\ y^A < y^B\}$ denote the predicted and gold comparison labels, respectively. The comparison reward is defined as

$$R_1 = \begin{cases} +2, & \text{if } c^{\text{pred}} = c^{\text{truth}}, \\ -2, & \text{otherwise.} \end{cases} \tag{5}$$

Task 2: Multi-label Principle Classification In addition to identifying the preferred response in Task 1, the model is further required to diagnose which principles are violated in the rejected reasoning. This is formulated as a multi-label classification problem over the six principles. Given predicted and gold violation sets $\mathcal{P}^{\text{pred}}$ and $\mathcal{P}^{\text{truth}}$, we define $n_{\mathcal{P}} = |\mathcal{P}^{\text{pred}} \cap \mathcal{P}^{\text{truth}}|$, $n_{\mathcal{P}}^{\text{pred}} = |\mathcal{P}^{\text{pred}}|$, $n_{\mathcal{P}}^{\text{truth}} = |\mathcal{P}^{\text{truth}}|$, with precision and recall computed as

$$\text{Prec}(\mathcal{P}^{\text{pred}}, \mathcal{P}^{\text{truth}}) = \begin{cases} \dfrac{n_{\mathcal{P}}}{n_{\mathcal{P}}^{\text{pred}}}, & n_{\mathcal{P}}^{\text{pred}} > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

$$\text{Rec}(\mathcal{P}^{\text{pred}}, \mathcal{P}^{\text{truth}}) = \begin{cases} \dfrac{n_{\mathcal{P}}}{n_{\mathcal{P}}^{\text{truth}}}, & n_{\mathcal{P}}^{\text{truth}} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

We then compute the F1-score $F_1(\cdot)$ and calculate the reward as

$$R_2 = 2 \cdot \left(F_1(\mathcal{P}^{\text{pred}}, \mathcal{P}^{\text{truth}}) - 0.5\right), \tag{8}$$

which maps the range of F_1 from $[0, 1]$ to $R_2 \in [-1, 1]$. The final reward used for GRPO is the normalized combination of both components: $r = (R_1 + R_2 + 3)/6$, resulting in a bounded reward within $[0, 1]$.

3.3 Training Configurations

The training pipeline for Judge-R1 follows a progressive three-stage design, as illustrated in Figure 4. Each stage is described below:

- **Stage 1: Data Scale and Domain Composition.** To study the effect of training scale, we construct three subsets of increasing size: (1) LRBENCH-21K (1,000 pairs per principle), (2) LRBENCH-63K (3,000 pairs per principle), and (3) LRBENCH-105K (5,000 pairs per principle). Each subset is sampled proportionally from the three domains: medical diagnosis, legal analysis, and academic review. By varying both scale and domain mix, this stage establishes the model's exposure to diverse argumentation structures while mitigating single-domain overfitting.
- **Stage 2: Rationale Enhancement & SFT Warmup.** To prevent early-stage degeneration, we introduce a lightweight supervised warm-up. Using Claude-3.7-Sonnet, we first generate full rationales for the correct judgments based on the gold labels. We then curate 2K rationale-augmented examples to fine-tune Qwen3-8B prior to RL training. This step teaches the model a structured, principle-aware reasoning format, reducing artifacts such as repeated "let me think again" loops and enabling GRPO to refine both judgment quality and policy stability.
- **Stage 3: Retrieval-Augmented Reasoning.** We then train the model with GRPO in an environment that allows `<search>` calls to retrieve targeted evidence when internal reasoning (inside `<think>` tags) lacks necessary information. Retrieved passages are embedded into `<information>` tags and injected back into the

reasoning loop (see Appendix Table 8). This retrieval-augmented RL setup grounds the model in a verifiable context, reduces hallucinations, and enables principled comparison between candidate reasonings. Final decisions are returned inside an <answer> block.

4 Experiments

In this section, we evaluate various LLMs and post-training algorithms on the test split of LRBENCH and compare them against Judge-R1. We further conduct ablation studies to inspect the contribution of different components of Judge-R1.

4.1 Experiment Setup

Baselines. We consider three categories of baselines:

- **General-purpose LLMs**, including Qwen2.5-7B (Yang et al., 2024), Qwen3-8B (Yang et al., 2025), GPT-OSS-120B (Agarwal et al., 2025), Claude-3.7-Sonnet (Anthropic, 2025), and DeepSeek-R1 (Guo et al., 2025).
- **Existing fine-tuned models**, including specialized judge models Prometheus2-7B (Kim et al., 2024) and JudgeLRM-7B (Chen et al., 2025a), as well as the agentic model Search-R1-7B (Jin et al., 2025) that does multi-turn search and reasoning.
- **Models fine-tuned on LRBENCH**, where we train Qwen3-8B using SFT, DPO, and GRPO, respectively.

Post-Training Algorithms. We compare three representative post-training algorithms for building judge models on LRBENCH against Judge-R1:

- **SFT:** a fully supervised baseline that directly learns both pairwise comparison and principle-level violation annotations from labeled data.
- **DPO:** a preference-based approach that learns to distinguish preferred and rejected reasoning pairs under the six evaluation principles.
- **GRPO:** an RL-based paradigm that optimizes the reward signals defined in Section 3.2, jointly targeting comparison accuracy and principle-level violation detection.

Notably, Judge-R1 shares the same backbone (Qwen3-8B) and can be viewed as a multi-stage extension of GRPO (Section 3.3) that incorporates lightweight SFT warm-up and retrieval-augmented, multi-turn search. For retrieval, we treat the case context as the search corpus, segmenting it into approximately 100-word passages (Karpukhin et al., 2020) while preserving complete sentence boundaries. These passages are indexed using FAISS and queried with a local E5-based retriever (Wang et al., 2022), which returns the top-k most relevant segments per search call ($k = 3$). To control training complexity, we limit the maximum number of search turns to three. Additional implementation details are provided in Appendix D.

Evaluation Protocol. The LRBENCH test split consists of 300 cases evenly sampled across the three domains. To ensure fair and consistent evaluation, we apply an identical prompt to all models (see Appendix E). We compute and report the overall accuracy scores across all domains and principles for each model, including:

- **Per-domain accuracy**: the proportion of cases in which a judge model correctly identifies the superior reasoning within each domain.
- **Per-principle accuracy**: for each of the six principles, the proportion of cases in which the judge correctly identifies the superior reasoning, revealing which principles are more challenging for different models.

4.2 Main Results

The main results are summarized in Table 3. Models post-trained on LRBENCH consistently outperform other 7B-8B judge models across all domains and evaluation principles. Among them, Judge-R1 achieves the strongest overall performance, surpassing much larger foundation models such as DeepSeek-R1. Based on these results, we draw the following key insights.

Effects of Training Paradigms. Across all domains, Judge-R1 achieves the best performance and yields the largest gain over the base model (Qwen3-8B), improving overall accuracy from 0.70 to 0.98. Among the post-training baselines, GRPO is the most effective and consistently outperforms the other preference-based algorithm, DPO. By contrast, SFT is less effective at small data scales: with 21K training examples in Table 3, it performs worse than the base model. However, as shown in Figure 5, SFT benefits substantially from additional training data and surpasses the baseline at larger scales.

Effects of Training Data Size. We further examine the effect of training scale by fine-tuning

Table 3: Performance of judge models on LRBENCH: per-domain accuracy and per-principle accuracy. **Bold** indicates the best-performing model within each column.

Model	Per-Domain Accuracy			Per-Principle Accuracy						Overall
	Legal	Academic	Medical	P1	P2	P3	P4	P5	P6	
General-Purpose LLMs										
Qwen2.5-7B	0.45	0.56	0.49	0.53	0.45	0.39	0.51	0.47	0.51	0.50
Qwen3-8B	0.81	0.59	0.69	0.75	0.68	0.78	0.65	0.72	0.82	0.70
GPT-OSS-120B	0.72	0.64	0.73	0.70	0.67	0.78	0.78	0.52	0.84	0.70
Claude-3.7-Sonnet	0.89	0.75	0.80	0.93	0.82	0.96	0.76	0.86	0.89	0.81
DeepSeek-R1	0.86	0.94	0.89	0.96	0.87	0.96	0.89	0.88	0.93	0.90
Existing Fine-Tuned Models										
Search-R1-7B	0.44	0.43	0.35	0.53	0.37	0.35	0.35	0.49	0.38	0.41
JudgeLRM-7B	0.44	0.61	0.43	0.46	0.50	0.48	0.48	0.48	0.55	0.49
Prometheus2-7B	0.47	0.56	0.54	0.53	0.53	0.43	0.56	0.48	0.53	0.52
Models Fine-Tuned on LRBENCH										
SFT-8B	0.70	0.63	0.69	0.75	0.67	0.57	0.60	0.73	0.76	0.67
DPO-8B	0.79	0.65	0.73	0.77	0.74	0.83	0.64	0.77	0.85	0.72
GRPO-8B	0.85	0.66	0.81	0.81	0.69	0.91	0.75	0.87	0.87	0.77
Judge-R1-8B	**0.99**	**0.95**	**0.99**	**0.98**	**1.00**	**1.00**	**0.97**	**0.96**	**1.00**	**0.98**

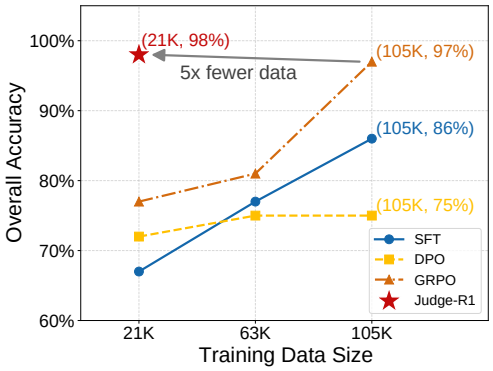


Figure 5: Data scaling effects on overall accuracy for SFT, DPO, GRPO, and Judge-R1.

Qwen3-8B on 21K, 63K, and 105K instances from LRBENCH, following the **Stage 1** setup described in Section 3.3. As shown in Figure 5 (with detailed results reported in Appendix F), both SFT and GRPO benefit consistently from increased data, exhibiting monotonic improvements in overall accuracy as the training set grows. In contrast, DPO shows diminishing returns and plateaus early, with only marginal gains at larger scales.

This behavior reflects a known limitation of the DPO paradigm with low-contrast pairs (Yan et al., 2024; Xu et al., 2024). In LRBENCH, chosen and rejected reasoning candidates often share substantial lexical and structural overlap, resulting in ambiguous preference signals. As data scales, these weak contrasts reduce the effective optimization margin, leading to unstable or noisy updates.

Notably, even with the largest training set, GRPO-105K underperforms Judge-R1 trained on only 21K instances (Appendix Table 12). Specifically, Judge-R1-21K reaches 98% accuracy using roughly 20% of the data required by GRPO-105K, demonstrating substantially higher data efficiency. This result highlights the advantage of integrating retrieval and structured supervision, enabling Judge-R1 to learn more effectively from limited data.

Domain/Principle-Specific Trends. Most methods achieve their highest accuracy in the legal domain (Table 3 and Figure 1). We conjecture that legal analysis typically exhibits more structured argumentation and clearer evaluation cues, which align well with the reasoning patterns learned by LLM-based judges. In contrast, the academic review domain appears to be the most challenging, as it often emphasizes novelty, implicit evaluation criteria, and emerging technical ideas that are underrepresented in the LLM pretraining corpus, making reliable judgments difficult.

At the principle level, evidence-oriented criteria, particularly **P2 (Factual Consistency)** and **P4 (Groundedness)**, remain the most challenging as shown in Table 3 and Figure 1. This trend highlights persistent challenges in detecting subtle factual inaccuracies and insufficient evidence support. A detailed error analysis is provided in Appendix H.

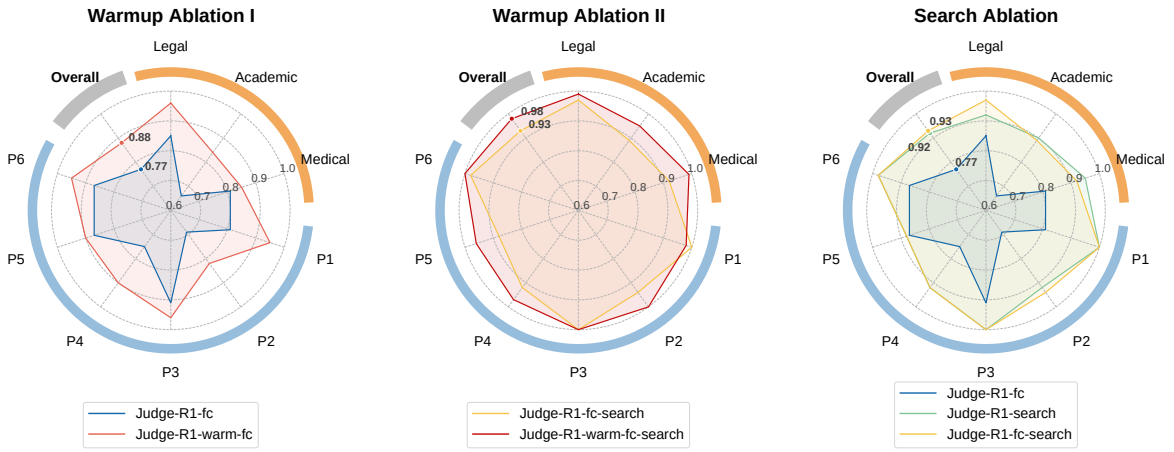


Figure 6: Impact of lightweight SFT warm-up and multi-turn search & reasoning on Judge-R1 performance.

Search-R1 on LRBENCH. Although Search-R1 (Jin et al., 2025) achieves strong performance on open-domain QA through agentic retrieval, it does not transfer directly to case-grounded judging. Under matched rollout settings and using the same case-context corpus as Judge-R1, Search-R1-7B reaches only 0.41 overall accuracy on LRBENCH, underperforming its base model, Qwen2.5-7B. We attribute this gap to a mismatch in retrieval objectives. In open-domain QA, retrieval is designed to surface broadly relevant passages from a large corpus. In our setting, however, the model must verify claims against the context of a specific case. In practice, 79.93% of passages retrieved by Search-R1-7B come from outside the target case. Judge-R1 avoids this issue by restricting retrieval to the current case, so that all retrieved evidence is directly relevant to the judgment task.

4.3 Ablation Studies

In this section, we examine how the SFT warm-up and multi-turn search & reasoning jointly improve the stability and effectiveness in the training process of Judge-R1. The definitions of variants are listed in Table 4. We find that while each component individually improves performance, their combination (`Judge-R1-warm-fc-search`) yields the strongest performance gains across all domains and evaluation principles.

SFT Warm-Up. In Judge-R1, we introduce a lightweight SFT warm-up that primes the model with structured, principle-aligned reasoning traces, hence initializing the model for a more stabilized GRPO training.

As illustrated in the first two panels of Figure 6, SFT warm-up consistently yields substantial performance gains across domains and principles.

Model	Configuration
`Judge-R1-fc`	Full context only; no warm-up, no search
`Judge-R1-warm-fc`	Lightweight SFT warm-up + full context; no search
`Judge-R1-search`	Search-only; no warm-up, no full context
`Judge-R1-fc-search`	Full context + multi-turn search; no warm-up
`Judge-R1-warm-fc-search`	Warm-up + full context + multi-turn search

Table 4: Ablation variants of Judge-R1. `fc` denotes that the full context of the input case is added to the prompt, `warm` indicates a lightweight SFT warm-up stage prior to GRPO, and `search` enables multi-turn evidence retrieval during GRPO training.

For the no-search setting, `Judge-R1-warm-fc` markedly improves over `Judge-R1-fc`, raising overall accuracy from 0.77 to 0.88 with broad gains across both domain-level and principle-level axes. A similar pattern holds when retrieval is enabled: `Judge-R1-warm-fc-search` improves overall accuracy from 0.93 to 0.98 compared to `Judge-R1-fc-search`. These results demonstrate that even a small amount of supervised, rationale-augmented data significantly enhances GRPO's stability and its ability to learn consistent pairwise preferences and fine-grained principle violations.

Multi-Turn Search & Reasoning. Next, we investigate the benefit of multi-turn search and reasoning by comparing various Judge-R1 configurations. As shown in the third panel of Figure 6, the `Judge-R1-search` configuration substantially outperforms the full-context baseline `Judge-R1-fc`, improving overall accuracy from 0.77 to 0.92. This result highlights the advantage of an agentic, on-demand search & reasoning over relying on static

full-context concatenation with single-round reasoning.

Furthermore, comparing `Judge-R1-fc-search` with `Judge-R1-search` reveals only a marginal additional gain (from 0.92 to 0.93), suggesting that explicitly concatenating the full context offers limited benefit once the model is equipped with effective multi-turn retrieval and reasoning capabilities.

4.4 Out-of-Distribution Generalizability

Here, we show that models trained on LRBENCH generalize well to external judge benchmarks, highlighting its practical value for supporting LLM-as-a-judge training under out-of-distribution (OOD) settings.

External Benchmarks. We involve RewardBench (Lambert et al., 2025) (Chat Hard subset, 456 pairs) and JudgeBench (Tan et al., 2024) (Reasoning subset, 98 pairs) into the evaluation. Following the RewardBench taxonomy, we further divide Chat Hard into six applications.

Selected Models. For OOD evaluation, we use the GRPO-trained Qwen3-8B model, as GRPO performed best among the three post-training methods in our earlier experiments. We compare it against the base Qwen3-8B model, with results reported in Table 5. Notably, the agentic version of Judge-R1 is not evaluated here because neither benchmark provides a case-specific context corpus for retrieval. The same prompt was applied to all compared methods in this experiment (Appendix Table 11).

Application	Qwen3-8B	GRPO-8B
judge-bench-reasoning	0.41	**0.58**
llmbar-adver-GPTInst	0.48	**0.54**
llmbar-adver-GPTOut	0.51	**0.55**
llmbar-adver-manual	0.37	**0.59**
llmbar-adver-neighbor	0.44	**0.46**
llmbar-natural	0.65	**0.72**
mt-bench-hard	0.65	**0.68**

Table 5: OOD performance across seven applications from RewardBench and JudgeBench.

As shown in Table 5, GRPO-8B outperforms its base model on all seven OOD applications. Notably, its accuracy on the JudgeBench Reasoning subset exceeds all methods reported in the JudgeBench paper, whose best result is 55.10%, achieved by Skywork-LLaMA-3.1-70B (see Appendix Table 13 for details). Overall, these results demonstrate that LRBENCH supports strong OOD generalization for training judge LLMs.

Shortcut-Learning Checks. To verify that OOD gains reflect genuine learning rather than exploitation of superficial cues, we analyze GRPO-8B's attention patterns, reasoning traces, and class bias, and compare them with those of its base model, Qwen3-8B (Appendix G). We find that, when producing judgment decision tokens, GRPO-8B places over 90% of its attention on the principle section, suggesting that its predictions are guided primarily by the evaluation principles rather than by sample-specific artifacts. In addition, its reasoning traces consistently identify the relevant principles and cite case-specific evidence more often than the base model. GRPO-8B also shows lower class bias between the two reasoning candidates. Taken together, these analyses provide no evidence that the OOD gains of our fine-tuned models on LRBENCH are driven by shortcut learning.

5 Conclusion

We introduce LRBENCH, a benchmark for training and evaluating judge LLMs on long-context reasoning across medical, legal, and academic-review domains, featuring a six-principle evaluation framework for fine-grained analysis of reasoning quality. We also propose Judge-R1, a post-training method that improves judge LLMs using LRBENCH. The experiments across supervision and preference-optimization settings (21K-105K examples) reveal that, despite strong overall accuracy, existing LLMs struggle to localize specific principle violations in long contexts. In contrast, Judge-R1, especially with lightweight SFT warm-up and retrieval-augmented GRPO, achieves performance competitive with large proprietary models while remaining data efficient, underscoring the importance of principled supervision and multi-step reasoning in next-generation judge LLMs. Beyond in-domain gains, results on external benchmarks further show that training on LRBENCH improves OOD generalization.

Limitations

While LRBENCH provides a comprehensive benchmark for evaluating judge LLMs, several limitations should be acknowledged. First, the current release covers only three domains (medical diagnosis, legal analysis, and academic review), and extending to additional domains such as scientific

reasoning or ethical decision-making would further enhance the benchmark's scope. Second, the flawed reasoning in LRBENCH is synthetically generated using Claude-3.7-Sonnet, which may not fully capture the diversity of natural reasoning errors produced by different models in real-world scenarios. Third, although we provide six evaluation principles, other important aspects of reasoning quality, such as reasoning efficiency or pedagogical clarity, are not explicitly covered. From a safety perspective, judge models trained on LRBENCH may also overlook deeper reasoning failures when flawed outputs appear superficially acceptable under the predefined principles, potentially masking harmful or misleading behaviors in high-stakes applications. This limitation highlights the importance of using automated judges as a complement to, rather than a replacement for, human oversight in ethically sensitive settings.

Acknowledgment

We thank Vincent Gao and Shirley Zhang for their support and encouragement of this work.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Anthropic. 2025. Claude 3.7 sonnet and claude code. https://www.anthropic.com/news/claude-3-7-sonnet.

Lang Cao, Zifeng Wang, Cao Xiao, and Jimeng Sun. 2024. PILOT: Legal case outcome prediction with case law. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 609–621, Mexico City, Mexico. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. Paragraph-level rationale extraction through regularization: A case study on european court of human rights cases. *arXiv preprint arXiv:2103.13084*.

Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. 2025a. Judgelrm: Large reasoning models as a judge. *arXiv preprint arXiv:2504.00050*.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, and 1 others. 2025b. Rm-r1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, and 1 others. 2025. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*.

Farieda Gaber, Maqsood Shaik, Fabio Allega, Agnes Julia Bilecz, Felix Busch, Kelsey Goon, Vedran Franke, and Altuna Akalin. 2025. Evaluating large language model workflows in clinical decision support for triage and referral and diagnosis. *npj Digital Medicine*, 8(1):263.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. 2024. Aligning llm agents by learning latent preference from user edits. *Advances in Neural Information Processing Systems*, 37:136873–136896.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Steve Han, Gilberto Titericz Junior, Tom Balough, and Wenfei Zhou. 2025. Judge's verdict: A comprehensive analysis of llm judge capability through human agreement. *arXiv preprint arXiv:2510.09738*.

Alon Jacovi, Andrew Wang, Chris Alberti, Connie Tao, Jon Lipovetz, Kate Olszewska, Lukas Haas, Michelle Liu, Nate Keating, Adam Bloniarz, and 1 others. 2025. The facts grounding leaderboard: Benchmarking llms' ability to ground responses to long-form input. *arXiv preprint arXiv:2501.03200*.

Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. s3: You don't need that much data to train a search agent via RL. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21599–21617, Suzhou, China. Association for Computational Linguistics.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Jaehun Jung, Faeze Brahman, and Yejin Choi. 2024. Trust or escalate: Llm judges with provable guarantees for human agreement. *arXiv preprint arXiv:2407.18370*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, and 1 others. 2025. Rewardbench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797.

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

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

Shuliang Liu, Xinze Li, Zhenghao Liu, Yukun Yan, Cheng Yang, Zheni Zeng, Zhiyuan Liu, Maosong Sun, and Ge Yu. 2025. Judge as a judge: Improving the evaluation of retrieval-augmented generation through the judge-consistency of large language models. *arXiv preprint arXiv:2502.18817*.

Bhuvanashree Murugadoss, Christian Poelitz, Ian Drosos, Vu Le, Nick McKenna, Carina Suzana Negreanu, Chris Parnin, and Advait Sarkar. 2025. Evaluating the evaluator: Measuring llms' adherence to task evaluation instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19589–19597.

OpenAI. 2025. Introducing openai o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/.

Pankayaraj Pathmanathan and Furong Huang. 2025. Reward models can improve themselves: Reward-guided adversarial failure mode discovery for robust reward modeling. *arXiv preprint arXiv:2507.06419*.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. 2025. Learning to plan & reason for evaluation with thinking-llm-as-a-judge. *arXiv preprint arXiv:2501.18099*.

Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya Parameswaran, and Ian Arawjo. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*.

Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. 2025. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441*.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2024. Judgebench: A benchmark for evaluating llm-based judges. *arXiv preprint arXiv:2410.12784*.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, pages 404–430.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, and 1 others. 2023. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, and 1 others. 2025. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. 2024. Cycleresearcher: Improving automated research via automated review. *arXiv preprint arXiv:2411.00816*.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning. *arXiv preprint arXiv:2505.10320*.

Kevin Wu, Eric Wu, Rahul Thapa, Kevin Wei, Angela Zhang, Arvind Suresh, Jacqueline J Tao, Min Woo

Sun, Alejandro Lozano, and James Zou. 2025a. Medcasereasoning: Evaluating and learning diagnostic reasoning from clinical case reports. *arXiv preprint arXiv:2505.11733*.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason E Weston, and Sainbayar Sukhbaatar. 2025b. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11548–11565.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

Zhichao Xu, Zongyu Wu, Yun Zhou, Aosong Feng, Kang Zhou, Sangmin Woo, Kiran Ramnath, Yijun Tian, Xuan Qi, Weikang Qiu, and 1 others. 2025. Beyond correctness: Rewarding faithful reasoning in retrieval-augmented generation. *arXiv preprint arXiv:2510.13272*.

Yuzi Yan, Yibo Miao, Jialian Li, Yipin Zhang, Jian Xie, Zhijie Deng, and Dong Yan. 2024. 3d-properties: Identifying challenges in dpo and charting a path forward. *arXiv preprint arXiv:2406.07327*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Yaohui Zhang, Haijing Zhang, Wenlong Ji, Tianyu Hua, Nick Haber, Hancheng Cao, and Weixin Liang. 2025. From replication to redesign: Exploring pairwise comparisons for llm-based peer review. *arXiv preprint arXiv:2506.11343*.

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

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Jiachen Zhu, Congmin Zheng, Jianghao Lin, Kounianhua Du, Ying Wen, Yong Yu, Jun Wang, and Weinan Zhang. 2025. Retrieval-augmented process reward model for generalizable mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8453–8468.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

A Related Work

The LLM-as-a-Judge paradigm has evolved from prompt-only meta-evaluation into a broader ecosystem of benchmarks and specialized judge models. JudgeBench (Tan et al., 2024) provides a standard way to measure pairwise judging accuracy, while JudgeLM (Zhu et al., 2023) and Prometheus-2 (Kim et al., 2024) explore how to build scalable, fine-tuned judges. Recent studies argue that judge quality is not just a correlation with humans but also how the judge behaves. Judge's Verdict (Han et al., 2025) uses Cohen's κ to separate "human-like" from "super-consistent" judging patterns, which may reflect either higher reliability or oversimplified judgments. Other work improves the transparency and robustness of supervision signals. RM-R1 (Chen et al., 2025b) introduces rubric-style reasoning rewards, J1 (Whitehouse et al., 2025) and planning-based variants like EvalPlanner (Saha et al., 2025) use RL to encourage "think-then-judge" traces, and retrieval-augmented or adversarial frameworks (e.g., RetrievalPRM (Zhu et al., 2025), REFORM (Pathmanathan and Huang, 2025)) study how to generalize beyond the training distribution and discover failure modes. In parallel, agentic RL with tools enables multi-turn evidence gathering. Search-R1 (Jin et al., 2025) trains models to interleave reasoning with multi-step search, while VERITAS (Xu et al., 2025) and FACTS (Jacovi et al., 2025) show that evidence localization and grounding remain difficult as contexts grow.

Judge-R1 builds on these threads but targets a different problem: long-context, retrieval-heavy judging where evidence can be missing or scattered, and the judge must actively decide what to retrieve before it can verify a claim. This differs from reasoning judges like J1 (Whitehouse et al., 2025), which optimize closed-book thinking traces without tool use. This setup matches most existing LLM-judge benchmarks, which are dominated by short to medium context where the verdict can usually be determined from the responses alone. In our setting (context can be up to 10,000 tokens), verifying everything inside the reasoning trace is both expensive and prone to misses, especially for principles tied to factual consistency and grounding. To address this, Judge-R1 treats retrieval as part of the RL rollout and outputs principle-level violation diagnoses rather than a single preference label, yielding structured, interpretable reward signals. This combination of agentic multi-turn retrieval and fine-grained principle supervision positions Judge-R1 beyond pairwise judges (Tan et al., 2024; Zhu et al., 2023; Kim et al., 2024) and generic think-then-judge training (Whitehouse et al., 2025; Saha et al., 2025).

B Data Sources

In the current release, LRBENCH covers three representative domains: medical diagnosis, legal analysis, and academic peer review. These domains are selected because they involve long and complex contexts where reasoning plays a central role in reaching well-justified conclusions. They also demand domain expertise for evaluation, making them especially suitable for evaluating the reliability of LLM judges. While we begin with these three domains, the framework is designed to be extensible and can be readily expanded to additional settings that provide rich contextual cases with reference reasoning.

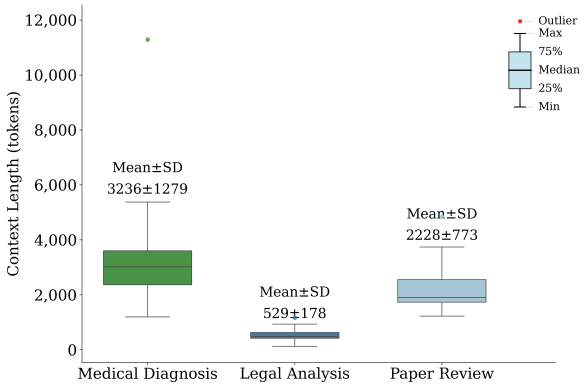
Medical Diagnosis. LLMs are increasingly used as virtual clinical assistants, supporting tasks such as triage, differential diagnosis, and treatment recommendation (Gaber et al., 2025). We adopt the `MedCaseReasoning` dataset (Wu et al., 2025a), where each case provides a detailed patient report and the corresponding case prompt, together forming the long-context input. Expert-authored diagnostic reasoning is provided as structured, evidence-grounded statements, which we use as the reference reasoning for each case.

Legal Judgment. LLMs have shown promise in judicial decision support, including identifying violations, interpreting statutes, and evaluating claim validity over lengthy legal documents (Chalkidis et al., 2021). For this domain, we use the `PILOT` dataset (Cao et al., 2024), where case descriptions and statutory articles form the long-context input. Because human-written judicial rationales are rarely available, we prompt Claude-3.7-Sonnet to generate reference reasoning conditioned on the ground-truth violated articles. This procedure yields high-quality, statute-grounded rationales that serve as the preferred reasoning for subsequent reasoning pair construction. Accordingly, the legal domain labels reflect better-vs.-worse reasoning preferences rather than expert-verified gold standards; human validation of these pairwise preferences is reported in Table 2.

Academic Review. LLMs are also being explored as automated reviewers capable of assessing research paper quality and issuing acceptance recommendations (Zhang et al., 2025). We adopt the `Research-14K` dataset (Weng et al., 2024), which contains full research papers paired with their review outcomes. Due to the length and density of these papers, we first prompt Claude-3.7-Sonnet to produce condensed summaries that faithfully capture the key aspects cited in reviews. We then adopt the "best-mode" outputs from `DeepReviewer` (Weng et al., 2024) as the reference reasoning, enabling consistent and systematic evaluation of LLM judges in scholarly long-context reasoning.

C Dataset Statistics

Beyond Table 1, we additionally report the distribution of context lengths across the three domains in Supplementary Figure 1. The medical diagnosis domain exhibits the longest and most variable contexts, with lengths reaching 11,287 tokens. Legal cases contain the shortest and least variable contexts, whereas academic review falls in between. Importantly, the contexts in all three domains can be further expanded if needed. For example, case summaries in the medical domain could be replaced with full journal articles, legal cases with original ECHR documents, and peer review with full manuscripts. In the current release, we do not adopt these extended contexts because they are often excessively long, sometimes multilingual, and would significantly complicate the training of Judge LLMs. We leave such extensions to future work.



Supplementary Figure 1: Context token length distributions across the three domains in LRBENCH.

We partition the dataset into training and test splits. Figure 3 (left) shows that pairwise comparison labels ($A > B$ vs. $A < B$) in the test set are nearly balanced, which helps mitigate position bias in judge LLMs. Figure 3 (right) presents the distribution of principle violations across cases. Because both single- and multi-principle violations are included, the proportions are not perfectly uniform; however, the distribution remains well balanced across the six principles, ensuring that models receive supervision that reflects the full spectrum of reasoning failure modes.

D Training Details

Using the pipeline described in Section 2, we generate 5,000 reasoning pairs for each single-principle violation and for each multi-principle violation. For a single domain, this results in 5,000 $\times 7 =$ 35,000 reasoning pairs, and across three domains, we obtain a total of approximately 105,000 pairs. After generation, we further process these reasoning pairs to align with the training data templates required by the different post-training paradigms evaluated in this work. SFT and DPO models are trained using the LLaMA-Factory (Zheng et al., 2024) pipeline, while GRPO and Judge-R1 are trained with the VeRL (Sheng et al., 2024) framework. SFT experiments are conducted on a single node, whereas DPO and GRPO experiments run on two nodes, each equipped with 8 NVIDIA H100 GPUs.

Supervised Fine-Tuning. For SFT, we adopt an Alpaca-style format with three fields: `instruction`, `input`, and `output` as shown in Supplementary Table 6. The `instruction` section specifies the evaluation task and the definitions of the six principles. The `input` contains the long-context case together with the two candidate reasonings. The `output` provides the pairwise comparison label and the set of violated principles for the flawed reasoning. This structure supplies both task grounding and explicit supervised signals for multi-label violation detection.

Direct Preference Optimization. For DPO, we use a ShareGPT-style conversational format as shown in Supplementary Table 7. Each case is represented as a three-part structure: a `conversations` field containing the user message that defines the evaluation task and the six principles, followed by the full case context. The two candidate reasonings are encoded as `chosen` (the reference reasoning) and `rejected` (the flawed rea-

Supplementary Table 6: Example instance of SFT training data in JSON format

SFT Training Data Template

```
{
  "instruction": "Compare the following two reasoning candidates (A and B)
      in terms of their quality under the following principles:\n{Six-
     Principle Definitions}\nRespond with two steps:\n1. First, indicate
     which reasoning is better with either 'A>B' or 'A<B';\n2. Then,
     identify the violated principles in the worse reasoning using the
     format: \"violated_principles\": [principle_numbers 1-6]",
  "input": "{Case ID + Case Context + Final Decision + Reasoning A +
     Reasoning B}",
  "output": "A<B\nviolated_principles: [2]"
}
```

Supplementary Table 7: Example instance of DPO training data in JSON format

DPO Training Data Template

```
{
  "conversations": [
    {
      "from": "human",
      "value": "Now, please identify the principles violated in the
          reasoning. {Six-Principle Definitions + Flawed Reasoning + Case
          Context}"
    },
    {
      "from": "gpt",
      "value": "The reasoning violates principles [3]."
    },
    {
      "from": "human",
      "value": "You are an expert specialized in evaluating clinical
          reasoning. Your reasoning should be logical, factually
          consistent, unbiased, grounded in the provided information,
          helpful, and harmless. According to the above case context,
          please provide your diagnosis reasoning."
    }
  ],
  "chosen": {
    "from": "gpt",
    "value": "{Reference Reasoning}"
  },
  "rejected": {
    "from": "gpt",
    "value": "{Flawed Reasoning}"
  }
}
```

soning), enabling the model to learn preference signals for pairwise comparison.

Group Relative Policy Optimization. For GRPO, we design a structured schema with four components: `data_source`, `prompt`, `reward_model`, and `extra_info`. The `prompt` is the core input to the model. It follows a role-based structure in which the system role contains the task instruction and principle definitions, and the user role presents the case context and paired reasoning. The `reward_model` encodes ground-truth supervision, including both the pairwise preference and the violated-principle labels. The `data_source` records the domain (medical, legal, or academic review), while `extra_info` stores auxiliary metadata such as case identifiers.

Supplementary Table 8: Predefined instructions for legal-domain training data used in Judge-R1.

Template for Judge-R1

Answer the task below. Reasoning A and Reasoning B are legal analyses for the same case. When evaluating them, compare their differences and conduct your evaluation reasoning inside `<think>...</think>` each time you identify a difference.

If you lack context about legal principles, case details, or ECHR jurisprudence, call a search engine with `<search>...</search>`; it will return the top relevant information within `<information>...</information>`.

You may search multiple times. When you are certain, provide **ONLY** the final evaluation result inside `<answer>...</answer>` as a single JSON object with **EXACT keys**.

For example, `<answer>`

```
{
  "comparison":"A<B",
  "violated_principles":[3,6]
}
```

`</answer>`.

Task: {system_content}

Please refer to the following legal case context and the two reasoning analyses (A and B).

Judge-R1. To train Judge-R1, following Jin et al. (2025), we design a minimal yet structured prompting template in Supplementary Table 8 that guides the model's interaction with the environment. The template specifies an iterative output format consisting of three components:

1. A step-by-step evaluation process enclosed in `<think>` and `</think>` tags, where the model compares differences between two reasoning candidates;
2. Optional calls to an external search engine via `<search>` and `</search>` tags when additional knowledge or case-specific context is required for judgment, with retrieved evidence returned in `<information>` and `</information>` tags;
3. A final decision produced inside an `<answer> ... </answer>` block as a structured JSON object.

Importantly, the template enforces only this interaction structure and output schema, without prescribing how often the model should reason, search, or which strategies it should adopt. By avoiding content-level constraints or heuristic guidance, this design allows the model's reasoning and retrieval behaviors to emerge naturally during RL training, enabling unbiased observation of how GRPO learns to integrate multi-turn search into principled judgment.

E Key Prompts

Prompts for LRBENCH. To construct LRBENCH, we generate paired reasonings for each case based on its long-context input and reference reasoning, as illustrated in Section 2.3. For each case, we use a structured chain-of-thought prompt to instruct a strong teacher model, i.e., Claude-3.7-Sonnet, to generate five flawed reasoning variants conditioned on the original correct analysis. Each variant preserves the original numbering, overall structure, and approximate length, while introducing one or more violations from the six evaluation principles.

The prompt in Supplementary Table 9 is designed to generate flawed reasonings that contain exactly one principle violation. It further requires the injected flaw to appear at a different position in the reasoning and to be integrated naturally into the surrounding logical flow, making the error subtle but still identifiable upon careful inspection. We generate an equal number of reasoning pairs for each principle violation, ensuring balanced coverage of how individual principle violations affect reasoning quality.

In addition, we use the prompt template in Supplementary Table 10 to generate flawed reasonings with multiple principle violations, which more closely reflect realistic errors in practice. Both

Supplementary Table 9: Instruction template for generating reasoning pairs with single-principle violations.

Dataset Building for LRBENCH

You are {domain_role}.

Input:
Case: {input_text}
Correct Reasoning / Reference: {gt}
Correct Decision: {decision_label}
Target Principle: {principle_name} - {principle_description}

Task: Using the original **Correct Reasoning** to create five distinct **Flawed Reasoning** variants. Each variant must:

- Follow the same numbering, structure, and approximate length as the original correct reasoning.
- Introduce exactly one {principle_violation} (i.e., a breach of {principle_name}: {principle_description}) in a different location in each variant.
- Be subtle enough to challenge an LLM judge, yet unambiguous on close reading.

Instructions:

1. Analyze the case context, correct reasoning, and final decision to fully grasp the case and maintain objectivity.
2. Produce five flawed versions of the correct reasoning. For each of the five variants:
   - Copy the original structure and flow of the correct reasoning.
   - Insert exactly one {principle_violation} (and only one) into the logic so it is not trivially obvious.
   - Place the flaw in a different step/section across the five variants.
3. Do not include any **Clue** section or annotations; only provide the flawed-reasoning texts.

Output Format:

```
Flawed Variant 1: {. . . }

. . .

Flawed Variant 5: {. . . }
```

Quality Checks Before Finishing:

- Exactly one {principle_name} violation per variant.
- The violation is placed in a different location/section in each of the five variants.
- Overall structure, style, and length remain consistent with the correct reasoning.

Begin now.

prompt templates are applied across all three domains: medical diagnosis, legal analysis, and academic review.

Each flawed variant is paired with the reference reasoning, yielding five labeled reasoning pairs per case. This procedure provides large-scale, principle-aware supervision while maintaining high structural similarity between the preferred and flawed reasonings, thereby closely reflecting realistic long-context judgment scenarios.

Prompts for Judge LLM Evaluation. To evaluate LLM-based judges on LRBENCH, we use a

Supplementary Table 10: Instruction template for generating reasoning pairs with multi-principle violations.

Dataset Building for LRBENCH

You are {domain_role}.

Input:
Case: {input_text}
Correct Reasoning / Reference: {gt}
Correct Decision: {decision_label}
Principle Set:

1. **Logical Correctness** – Reasoning is coherent and free of contradictions.
   .....
6. **Harmlessness** – Output avoids harmful or unethical content and consequences.

Task: Using the original **Correct Reasoning** to create five distinct **Flawed Reasoning** variants. Each variant must:

- Follow the same numbering, structure, and approximate length as the original correct reasoning.
- Contain **at least two distinct** principle violations chosen from the **Principle Set** above.
- Distribute the violations across different parts of the reasoning (do not concentrate all flaws in one spot).
- Use a **different combination** of violated principles across variants.
- Be subtle (requires careful reading) yet discoverable, and meaningfully affect the validity of the reasoning.

Instructions:

1. Analyze the case context, correct reasoning, and final decision to fully grasp the case and maintain objectivity.
2. Produce five flawed versions of the correct reasoning. For each of the five variants:
   - Copy the original structure and flow of the correct reasoning.
   - Insert at least two violations from the **Principle Set** into the logic so it is not trivially obvious.
   - Place violations in a different step/section across the five variants.
3. Do not include any **Clue** section, error tags, or location hints; only provide the flawed-reasoning texts.

Output Format:

```
Flawed Variant 1: Violated Principles: [1, 3], {. . . }

. . .

Flawed Variant 5: Violated Principles: [2, 4, 5, 6], {. . . }
```

Quality Checks Before Finishing:

- Each variant lists **at least two distinct** principle IDs and actually violates those principles.
- Violations are distributed across different locations/sections.
- Variants use different principle combinations.
- Overall structure, style, and length remain consistent with the correct reasoning.

Begin now.

Supplementary Table 11: Prompt for model evaluation on LRBENCH test set.

Judge Evaluation

Role Definition: You are a precise evaluator comparing two {domain} reasoning approaches (A and B) for the same case.

Evaluation Principles: Compare using exactly these six principles:

1. **Logical Correctness** – Reasoning is coherent and free of contradictions.
2. **Factual Consistency** – Claims match verifiable facts.
3. **Bias & Fairness** – Decisions actively mitigate unjust bias and ensure equity.
4. **Groundedness** – No invented details; all reasoning must be supported by explicit information from the source.
5. **Helpfulness** – Explanations are complete and offer actionable, useful guidance for stakeholders.
6. **Harmlessness** – Output avoids harmful or unethical content and consequences.

Output Format: Return **ONLY** a JSON object with the following **exact fields**:

```
{
  "comparison": "<REQUIRED: A>B or A<B>",
  "violated_principles": [<REQUIRED: Principle numbers 1-6>],
  "explanation": "<REQUIRED: Single sentence, max 50 words>"
}
```

Field Rules:

- `comparison`: Must be exactly `A>B` or `A<B`.
- `violated_principles`: Must be an array containing only numbers 1–6.
- `explanation`: Must be under 50 words.

Critical Requirements:

1. Output **ONLY** the JSON object; no additional text.
2. Use **ONLY** the six numbered principles listed above.
3. Ensure the JSON format is exactly as specified.
4. Base all judgments solely on the provided case materials.
5. Focus on accuracy and professionalism.

Violation Rules:

- If the worse reasoning is A or B, list the principle numbers it violates (e.g., `[1, 3]`).
- If no violations are identified, use an empty array `[]`.

Example Output:

```
{
  "comparison": "A>B",
  "violated_principles": [3, 6],
  "explanation": "Reasoning B makes inappropriate assumptions and includes
      biased personal judgments."
}
```

standardized, principle-driven evaluation prompt template, shown in Supplementary Table 11, that frames the model as a neutral adjudicator comparing two reasoning candidates for the same case. The placeholder {domain} is instantiated as medical diagnosis, legal analysis, or academic review, depending on the domain. The prompt defines six evaluation principles and requires the model to determine (i) which reasoning is preferable and (ii) which principles are violated by the inferior reasoning.

To ensure consistency and comparability across models, the prompt enforces a strict JSON output schema specifying the pairwise comparison result,

Supplementary Table 12: Comprehensive performance of judge models on LRBENCH across all training scales: per-domain accuracy and per-principle accuracy. **Bold** indicates the best-performing model within each column.

Model	Per-Domain Accuracy			Per-Principle Accuracy						Overall
	Legal	Academic	Medical	P1	P2	P3	P4	P5	P6	
General-Purpose LLMs										
Qwen2.5-7B	0.45	0.56	0.49	0.53	0.45	0.39	0.51	0.47	0.51	0.50
Qwen3-8B	0.81	0.59	0.69	0.75	0.68	0.78	0.65	0.72	0.82	0.70
GPT-OSS-120B	0.72	0.64	0.73	0.70	0.67	0.78	0.78	0.52	0.84	0.70
Claude-3.7-Sonnet	0.89	0.75	0.80	0.93	0.82	0.96	0.76	0.86	0.89	0.81
DeepSeek-R1	0.86	0.94	0.89	0.96	0.87	0.96	0.89	0.88	0.93	0.90
Existing Fine-Tuned Models										
Search-R1-7B	0.44	0.43	0.35	0.53	0.37	0.35	0.35	0.49	0.38	0.41
JudgeLRM-7B	0.44	0.61	0.43	0.46	0.50	0.48	0.48	0.48	0.55	0.49
Prometheus2-7B	0.47	0.56	0.54	0.53	0.53	0.43	0.56	0.48	0.53	0.52
SFT Variants										
SFT21K-8B	0.70	0.63	0.69	0.75	0.67	0.57	0.60	0.73	0.76	0.67
SFT63K-8B	0.80	0.77	0.73	0.77	0.81	0.61	0.73	0.86	0.75	0.77
SFT105K-8B	0.94	0.82	0.83	0.84	0.83	**1.00**	0.85	0.84	0.96	0.86
DPO Variants										
DPO21K-8B	0.79	0.65	0.73	0.77	0.74	0.83	0.64	0.77	0.85	0.72
DPO63K-8B	0.82	0.71	0.71	0.86	0.74	0.87	0.67	0.80	0.87	0.75
DPO105K-8B	0.83	0.71	0.70	0.79	0.76	0.87	0.70	0.78	0.85	0.75
GRPO Variants										
GRPO21K-8B	0.85	0.66	0.81	0.81	0.69	0.91	0.75	0.87	0.87	0.77
GRPO63K-8B	0.93	0.72	0.78	0.91	0.67	0.87	0.78	0.89	0.89	0.81
GRPO105K-8B	0.98	**0.98**	0.94	**1.00**	0.96	**1.00**	**0.99**	0.93	**1.00**	0.97
Judge-R1 Variants										
Judge-R1-search	0.92	0.90	0.95	**1.00**	0.92	**1.00**	0.92	0.88	0.98	0.92
Judge-R1-fc-search	0.97	0.89	0.92	**1.00**	0.94	**1.00**	0.92	0.88	0.98	0.93
Judge-R1-warm-fc-search	**0.99**	0.95	**0.99**	0.98	**1.00**	**1.00**	0.97	**0.96**	**1.00**	**0.98**

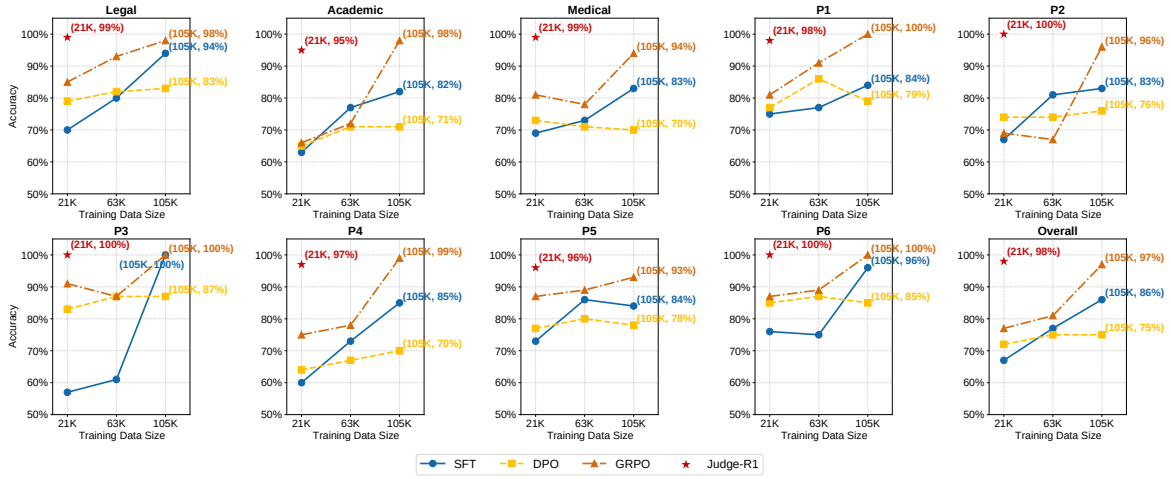
the set of violated principles, and a brief explanatory sentence. The model is further instructed to base its judgment solely on the provided case materials, avoiding any external assumptions or stylistic preferences. This design enables controlled, reproducible assessment of both coarse-grained preference accuracy and fine-grained principle-level violation detection.

F Additional Experimental Results

Comprehensive Results on LRBENCH. Supplementary Table 12 reports a comprehensive comparison of judge models on LRBENCH across domains, principles, and overall accuracy. Large foundation models achieve strong performance, but exhibit notable variability across principles, particularly on factual consistency and groundedness. Existing fine-tuned models lag substantially behind both foundation models and task-specific fine-tuning.

Among the post-training paradigms, GRPO consistently outperforms SFT and DPO at comparable scales, with GRPO105K-8B achieving near-saturated performance across most principles. Notably, Judge-R1 variants trained on only 21K instances outperform all other models, including GRPO105K-8B and large proprietary LLMs, across nearly all domains and principles. The best configuration, Judge-R1-warm-fc-search, achieves the highest overall accuracy (0.98), demonstrating that lightweight warm-up combined with retrieval-augmented RL yields superior data efficiency and robust, principle-aware judgment performance.

Supplementary Figure 2 presents a comprehensive breakdown of how training data scale affects judge performance across domains, principles, and overall accuracy. SFT and GRPO models exhibit consistent and largely monotonic improvements as training data increases from 21K to 105K, indicating that both paradigms benefit from larger supervision. In contrast, DPO shows early saturation across most domains and principles, with



Supplementary Figure 2: Effect of training data scale on SFT, DPO, GRPO, and Judge-R1 on LRBENCH.

only marginal gains or occasional regressions at larger scales, highlighting its limited ability to exploit additional data when preference signals are ambiguous.

Notably, Judge-R1 achieves near-saturated performance with only 21K training examples, matching or exceeding the best-performing GRPO models trained on 105K data across nearly all domains and principles. This consistent gap demonstrates that Judge-R1 is substantially more data-efficient, achieving strong, stable performance with significantly fewer training instances.

OOD Results on JudgeBench. Supplementary Table 13 compares the OOD performance of our base model, Qwen3-8B, and the GRPO-trained model on the Reasoning subset of JudgeBench against the results reported in Table 1 of JudgeBench (Tan et al., 2024). GRPO-8B achieves 58.16% accuracy (57/98), substantially outperforming the base Qwen3-8B at 40.82% (40/98). It also exceeds the strongest reported JudgeBench result, 55.10% from Skywork-LLaMA-3.1-70B, as well as prompted proprietary judges such as GPT-4o (47.96%) and Gemini-1.5-Pro (44.90%).

These results show that the gains from our principle-level supervision and GRPO training are not limited to LRBENCH, but generalize effectively to external judge-evaluation benchmarks.

Supplementary Table 13: Evaluating LLM-based judges on the Reasoning subset of JudgeBench.

Model	Accuracy
JudgeBench-Reported Results	
Vanilla (GPT-4o)	47.96%
Arena-Hard Judge (GPT-4o)	54.08%
VertexAI Evaluation (Gemini-1.5-pro)	44.90%
PandaLM	21.43%
Prometheus2-7b	25.51%
Prometheus2-8x7b	39.80%
Prometheus2-bgb-8x7b	30.61%
JudgeLM-7B	29.59%
JudgeLM-13B	29.59%
JudgeLM-33B	48.98%
AutoJ	29.59%
Skywork-LLaMA-3.1-8B	54.08%
Skywork-LLaMA-3.1-70B	**55.10%**
ChatEval	31.63%
Base Model	
Qwen3-8B	40.82%
Our Model	
GRPO-8B	**58.16%**

G Shortcut-Learning Checks

To investigate the possibility of shortcut learning, in which the model exploits superficial linguistic cues or generation artifacts in synthetic flawed reasoning rather than truly learning the intended evaluation principles, we conduct Explainable AI (XAI) analyses. Specifically, we examine three common indicators of shortcut learning: attention patterns, model reasoning patterns, and class bias. Our analysis focuses on GRPO-8B, trained on LRBENCH, which consistently improves OOD judge accuracy across seven applications from two independent benchmarks, as shown in Table 5.

Attention Patterns. For the attention analysis in Supplementary Table 14, we extract attention weights from the last four transformer layers of GRPO-8B at the exact timestep when the model

Supplementary Table 14: Section-wise attention distribution of GRPO-8B when generating the preference decision token.

Prompt Section	Overall	Prediction Direction			Prediction Correctness		
	Mean Attention	A>B	A<B	Diff.	Correct	Incorrect	Diff.
Principles	90.11%	90.30%	89.89%	+0.41%	90.35%	89.63%	+0.72%
Instructions	3.17%	3.44%	2.84%	+0.61%	3.20%	3.10%	+0.10%
Context	1.31%	1.17%	1.47%	-0.30%	1.33%	1.26%	+0.06%
Reasoning A	2.43%	2.30%	2.59%	-0.29%	2.25%	2.79%	-0.54%
Reasoning B	2.98%	2.79%	3.21%	-0.42%	2.87%	3.22%	-0.34%

generates its preference decision token. We then average attention over all heads within each layer to characterize the information the model attends to when making its decision. Among all 554 OOD samples involved in Section 4.4, the GRPO-8B model produces a clear decision token for 449 samples; we therefore perform the following analysis on this subset. Specifically, we examine how attention is distributed across major prompt sections and compare these distributions by predicted preference direction ($A > B$ vs. $A < B$) and by prediction correctness (correct vs. incorrect).

Two findings emerge from the attention analysis. First, GRPO-8B places high attention on the principle section when generating its decision token ($>$ or $<$): 90.11% of attention is allocated to this part of the prompt. This behavior is desirable, since the preference decision is explicitly defined with respect to the six evaluation principles. Moreover, because the principle section is identical across samples, it is unlikely to encode sample-specific artifacts that could support shortcut-based decisions.

Second, the attention distributions remain highly similar across both prediction directions and correctness conditions. The largest difference observed in Supplementary Table 14 is only 0.72%, indicating that the model does not exhibit meaningfully different attention behavior when predicting $A > B$ versus $A < B$, or when making correct versus incorrect judgments. In addition, despite significant length variance in user context and responses across samples, the section-level attention distribution remains stable. This suggests that the model does not rely on simple length-based cues when making decisions.

Reasoning Patterns. We further inspect the reasoning traces produced by GRPO-8B before it emits the final preference decision. Qualitatively, these traces are often stronger than those of the Qwen3-8B baseline on the OOD test set. Supplementary Table 15 summarizes overall accuracy together with two reasoning-related behaviors measured for each case: whether the model explicitly names at least one evaluation principle (**Principle Naming**) and whether it cites specific case content to achieve its judgment (**Evidence Citation**). Both models mention at least one principle in every case, indicating that principle-aware reasoning is consistently present in their outputs. However, GRPO-8B achieves higher overall accuracy and slightly stronger evidence citation than Qwen3-8B, suggesting that its judgments are more often grounded in case-specific evidence rather than generic or stylistic cues.

Supplementary Table 15: Comparison of reasoning-trace quality between GRPO-8B and its base model.

Model	Accuracy	Principle Naming	Evidence Citation
Qwen3-8B	49.6%	100.0%	95.6%
GRPO-8B	57.4%	100.0%	96.4%

Class Bias. We also test whether the model's gains could be explained by a simple directional bias toward one candidate. Supplementary Table 16 reports both the prediction distribution (**Pref. A/B**) and the accuracy conditioned on the ground-truth preferred candidate (**Acc.@A/B**). As shown in the first row, the ground-truth labels are well balanced between $A > B$ and $A < B$ on the OOD set. Although both GRPO-8B and Qwen3-8B exhibit a mild preference toward A, GRPO-8B is noticeably less biased than the base model (53.5% vs. 55.2% Pref. A). Moreover, GRPO-8B achieves higher conditional accuracy on both subsets: when the ground-truth preferred reasoning is A, its accuracy is 61.0% compared with 54.3% for Qwen3-8B; when the ground-truth preferred reasoning is B, its accuracy is 54.0% compared with 45.3%. These results

indicate that the performance gains of GRPO-8B cannot be explained by a trivial class-bias heuristic.

Supplementary Table 16: Prediction distribution and class-conditional accuracy on the OOD set.

Model	Pref. A	Pref. B	Acc.@A	Acc.@B
OOD Dataset	48.6%	51.4%	NA	NA
Qwen3-8B	55.2%	44.8%	54.3%	45.3%
GRPO-8B	53.5%	46.5%	61.0%	54.0%

Taken together, the attention analysis, reasoning-trace inspection, and class-bias analysis provide no evidence that the OOD gains of GRPO-8B on LRBENCH arise from shortcut learning.

H Error Analysis

Building on the quantitative results in Supplementary Table 12, we further analyze qualitative failure modes and model disagreements to better understand what LLM judges struggle with in long-context evaluation on LRBENCH.

Across prior judge models, excluding our Judge-R1 variants, two failure patterns consistently emerge. First, positional bias is a major shortcut. Some models strongly favor reasoning A regardless of content. For example, Prometheus2-7B predicts $A > B$ in 100% of cases, and JudgeLRM-7B does so in 85.5% of cases, indicating severe bias. Even large foundation models, such as Claude-3.7 (69.6%) and GPT-OSS-120B (78.0%), show a moderate tendency to prefer $A > B$. In contrast, our GRPO variants produce much more balanced predictions, with class bias around 40%.

Second, error rates vary substantially across domains. Failures are more concentrated in open-ended and novelty-intensive settings, such as academic review and medical reasoning, where relevant evidence is harder to identify and weigh. GRPO training substantially reduces both positional bias and overall error rates in these challenging domains.

For the Judge-R1 variants, the remaining errors are relatively rare but still structured. Most errors are flips from the correct $A > B$ to the incorrect $A < B$, accounting for approximately 70%-86% of all mistakes: 16 of 23 errors for `Judge-R1-search` (70%), 19 of 22 for `Judge-R1-fc-search` (86%), and 6 of 7 for `Judge-R1-warm-fc-search` (86%). These failures typically occur when the two candidate reasonings appear highly similar at first glance, despite subtle principle violations in the inferior one, or when the retrieved evidence is noisy or only weakly relevant.

`Judge-R1-warm-fc-search` performs best among all Judge-R1 variants, making only 7 errors out of 300 examples. Its remaining failures are mostly isolated cases of overconfident judgment, where the model commits to a preference without fully carrying out the intended reasoning process.