Crowd Comparative Reasoning: Unlocking Comprehensive Evaluations for LLM-as-a-Judge

Qiyuan Zhang[1], **Yufei Wang**[2], **Yuxin Jiang**[3], **Liangyou Li**[2], **Chuhan Wu**[2], **Yasheng Wang**[2], **Xin Jiang**[2], **Lifeng Shang**[2], **Ruiming Tang**[2], **Fuyuan Lyu**[4], **Chen Ma**[1*]
[1]City University of Hong Kong, [2]Huawei Noah's Ark Lab,
[3]The Hong Kong University of Science and Technology (Guangzhou),
[4]McGill University & Mila - Quebec AI Institute
qzhang732-c@my.cityu.edu.hk, wang.yufei1@huawei.com,

Abstract

LLM-as-a-Judge, which generates chain-of-thought (CoT) judgments, has become a widely adopted auto-evaluation method. However, its reliability is compromised by the CoT reasoning's inability to capture comprehensive and deeper details, often leading to incomplete outcomes. Existing methods mainly rely on majority voting or criteria expansion, which is insufficient to address the limitation in CoT. We propose Crowd-based Comparative Evaluation, which introduces additional crowd responses to compare with the candidate responses, thereby exposing deeper and more comprehensive details within the candidate responses. This process effectively guides LLM-as-a-Judge to provide a more detailed CoT judgment. Extensive experiments demonstrate that our approach enhances evaluation reliability, achieving an average accuracy gain of 6.7% across five benchmarks. Moreover, our method produces higher-quality CoTs that facilitate judge distillation and exhibit superior performance in rejection sampling for supervised fine-tuning (SFT), referred to as crowd rejection sampling, thereby enabling more efficient SFT. Our analysis confirms that CoTs generated by ours are more comprehensive and of higher quality, and evaluation accuracy improves as test-time computation scales. Our code is available at https://github.com/Don-Joey/CCE.git.

1 Introduction

With the prohibitive cost and limited scalability of human evaluation, LLM-as-a-Judge has emerged as a scalable framework for auto-evaluation (Chang et al., 2024; Li et al., 2024a, 2025). Given a *task instruction* and corresponding *candidate responses*, LLM-as-a-Judge (Zheng et al., 2023; Wang et al., 2024b; Wagner et al., 2024) employs CoT judgment to analyze granular quality details of the responses, ultimately deriving a final outcome.

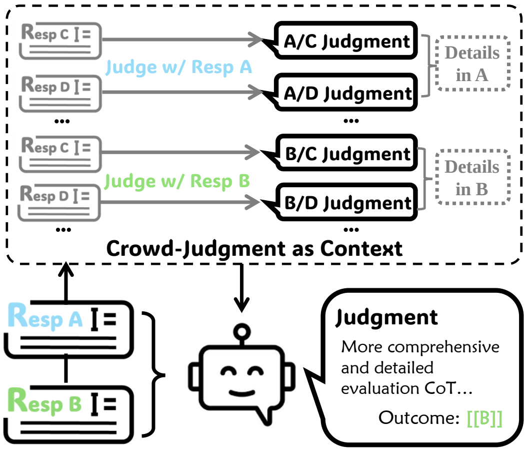

Figure 1: An overview of our method. By evaluating the candidate responses A/B alongside the crowd responses, the resulting crowd judgment can be used as context to enrich the evaluation of A/B responses, leading to a more comprehensive CoT judgment.

Despite advancements in techniques such as CoT reasoning (Saha et al., 2025; Zheng et al., 2023), specialized rubrics (Liu et al., 2023), and preference-aligned training datasets (Li et al., 2024b; Wang et al., 2024d), human evaluation remains the gold standard due to persistent limitations (Zeng et al., 2024a) in LLM-as-a-Judge. These limitations include biases (Park et al., 2024) in judgment and susceptibility to misleading context (Dubois et al., 2024a; Chen et al., 2024), which undermine the reliability of automated evaluation. One important yet overlooked reason is that the quality of CoT reasoning hinges on the model's ability to comprehensively compare nuanced details across responses. Our observation reveals high-quality judgments incorporate a thorough comparison of these details, while flawed reasoning tends to focus on limited details, leading to premature and incomplete outcomes. Therefore, enhancing the richness and comprehensiveness of CoT reasoning is essential to improve LLM-as-a-Judge.

Two commonly adopted strategies aim to address

[*]Corresponding Author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5059–5074
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

this issue: majority voting (Zhang et al., 2024; Mahan et al., 2024; DeepSeek-AI, 2024) and criteria expansion (Kim et al., 2024a; Liu et al., 2024; Hu et al., 2024a). The majority voting generates multiple judgments independently in parallel and aggregates these results through voting. It essentially leverages the randomness from temperature sampling to encourage detailed reasoning. However, this approach is passive and computationally expensive. In contrast, criteria expansion augments prompts with additional evaluation aspects, proactively guiding the model to consider more dimensions of quality. Yet, this strategy is response-unaware, failing to adapt the evaluation process to the unique details of each response. For instance, even if a response is rich with nuanced insights, incorporating a criterion like "accuracy" does little to prompt the LLM to identify the unique details of its reasoning. Consequently, neither approach effectively guides LLM-as-a-Judge to consistently produce nuanced, comprehensive CoT evaluations. This leads to a critical research question: *how can we guide LLMs to engage in deeper, more detail-rich CoT reasoning during judgment?*

In this work, we propose a novel **crowd-based comparative evaluation** (**CCE**) to address this challenge by enabling LLM-as-a-Judge to uncover valuable details, as depicted in Figure 1. Our approach is inspired by human evaluative behavior: humans merely compare candidates in isolation by also contrasting them against a broader crowd, thereby uncovering additional nuanced insights about each candidate. The crowd serves as cognitive scaffolding (Obukhova and and, 2009), it forces the evaluation to traverse candidates of varying quality levels, triggering a deeper understanding of candidates' features. Building on this principle, CCE first gathers a set of alternative responses to the task instruction, referred to as *crowd responses*, and then compares each candidate response against these crowd responses to derive multiple *crowd judgments*. Throughout this process, the diversity of crowd responses serves as multiple evaluation anchors, revealing different layers of detail within the candidate responses. Based on this, CCE prompts the LLM-as-a-Judge to perform a more comprehensive and deeper overall CoT judgment.

CCE achieves a remarkable average improvement of 6.7% across five judge benchmarks, including RewardBench, HelpSteer2, MTBench Human, JudgeBench and EvalBias. When applied to judge distillation, we find that the high-quality long CoT judgments generated by CCE enable a smaller judge model to achieve higher accuracy, yielding an average improvement of 4.5%-5.6% (in Qwen 2.5-7B), particularly enhancing bias robustness. Moreover, we extend CCE naturally to SFT rejection sampling, referred to as *crowd rejection sampling*, where our approach serves as a quality signal to identify training-efficient samples from the response pool. Our enhanced rejection strategy consistently outperforms both random sampling and vanilla rejection sampling on MTBench and AlpacaEval-v2, demonstrating the reliability and practical utility of CCE in LLM alignment. Finally, our analysis confirms that CCE scales test-time compute effectively and produced CoTs consistently yield more key points and capture finer-grained details within responses compared to Vanilla LLM-as-a-Judge, facilitating more comprehensive and deeper CoT reasoning.

2 Related Work

Human evaluation is typically regarded as the gold standard for evaluating LLM responses to intricate and open-ended instructions (Chiang and Lee, 2023; Elangovan et al., 2024). Nevertheless, due to its inherent limitations—being time-consuming, costly, and prone to variability (Karpinska et al., 2021)—automated evaluation methods leveraging LLMs have gained prominence as scalable and cost-efficient alternatives. Unlike reward models that provide only scalar scores (Wang et al., 2024a,b), LLM-as-a-Judge frameworks offer enhanced robustness and interpretability by producing detailed CoT rationales (Li et al., 2024c; Gao et al., 2024).

Enhancing the performance of LLM-as-a-Judge has attracted significant attention, with many techniques proposed recently. One prominent approach involves fine-tuning pre-trained LLMs on task-specific datasets to better adapt them for judgment tasks (Vu et al., 2024; Li et al., 2024b; Wang et al., 2024d; Kim et al., 2024b). Another line of research focuses on step-by-step methodologies, such as G-EVAL (Liu et al., 2023), ICE-Score (Zhuo, 2024), and EvalPlanner (Saha et al., 2025), which decompose complex evaluation tasks into granular components, thereby harnessing the reasoning capabilities of LLMs to streamline the evaluation process. Additionally, recent advances explore using LLMs to generate reasoning traces by designing domain-specific prompts and meticulously crafting components of CoT reasoning. These include constructing fine-grained scoring rubrics (Zheng et al., 2023; Zeng et al., 2024b; Trivedi et al., 2024) and generating reference answers (Zhang et al., 2025). Building on this direction, recent studies have further explored expanding or restructuring evaluation criteria to enhance judgment quality (Kim et al., 2024a; Liu et al., 2024; Hu et al., 2024b). For instance, Pereira et al. (2024); Kim et al. (2024c) propose enriching coarse preference-based judgments with more fine-grained, interpretable, or multi-dimensional criteria, enabling LLM judges to produce more consistent and informative evaluations. Despite these efforts, the richness and comprehensiveness of CoT reasoning remain underexplored, leaving room for further advancements in improving LLM-as-a-Judge. While simple heuristics such as majority voting (Badshah and Sajjad, 2024; Verga et al., 2024) can mitigate this issue by improving the reliability and accuracy of evaluations, they often fall short in terms of efficacy and efficiency.

3 Methodology

As illustrated in Figure 2, we propose a crowd-based comparative evaluation that elicits and integrates multiple crowd judgments before producing a final outcome. It consists of three core components: (1) Crowd Response and Judgment Generation, (2) Crowd Judgment Selection and Processing, and (3) Context-augmented Inference, which we will discuss in the following subsections. Furthermore, we distill the CoT judgments generated by CCE to train a judge and expand its application to an enhanced rejection sampling technique for SFT.

3.1 Problem Formulation

Supposing $\{y^A, y^B\}$ denote two candidate responses generated by two assistants for a given task instruction x, Vanilla LLM-as-a-Judge $\mathcal{F}$ is prompted to provide a CoT-based judgment j of y^A and y^B, based on a specific set of evaluation criteria s (*e.g.*, *correctness*, *coherence*).

$$j = \mathcal{F}(y^A, y^B | x, s). \tag{1}$$

The objective is to ensure that the $\mathcal{F}$ preference aligns closely with human evaluation. In pairwise comparisons, this alignment is quantified by measuring the accuracy relative to human labels.

3.2 Crowd Response and Judgment Generation

Based on the task instruction x, we first prompt the LLM to generate a set of n synthetic crowd responses $\{y^i | i \in \{C, D, E, ...\}\}$. To enhance the diversity of these responses, we can leverage multiple LLMs ranging from smaller models (*e.g.*, *Qwen2.5-0.5B-Instruct*) to larger ones (*e.g.*, *Mistral-Nemo-Instruct-2407*), along with varying temperature settings. Theoretically, more diverse responses can cover a wider range of scenarios. When compared with y^A and y^B, these crowd responses emphasize different details of $\{y^A, y^B\}$, offering a more comprehensive perspective and facilitating deeper reasoning. As Figure 2 demonstrated, crowd judgment digs the importance of "he", where Response A subtly shifts the actor "he" onto the object "task" itself, thereby violating the instruction's requirement to rewrite while preserving the concise original meaning. Then, we use it as context to reinforce the following CoT. This advantage surpasses that of criteria expansion, which cannot anticipate such details through pre-prompting.

For each synthetic y^i, $\mathcal{F}$ independently produces two crowd judgments, j_i^A and j_i^B, by individually judging y^i with y^A and y^B, separately:

$$j_i^A = \mathcal{F}(y^A, y^i | x, s), \quad j_i^B = \mathcal{F}(y^B, y^i | x, s). \tag{2}$$

Formally, we collect a set of $2n$ crowd judgments:

$$\mathcal{J} = \{ j_i^A, j_i^B \mid i \in \{C, D, E, ...\} \}. \tag{3}$$

While each judgment may not fully capture all details of the candidate responses, they together provide a richer pool of evidence about how y^A and y^B differ in nuanced ways.

3.3 Crowd Judgment Selection and Processing

After obtaining $\mathcal{J}$, the key stage lies in selecting and processing these judgments effectively. Random Selection is neither stable nor optimal, so we need better strategies for using crowd judgments.

To this end, we propose a simple yet effective method called **Criticizing Selection**. Specifically, we choose judgments based on their outcomes: for j_i^A, we keep those where A loses, and for j_i^B, those where B loses. Notably, our observation reveals judgments with a critical outcome tend to provide detailed and informative reasoning for the criticized response. For instance, Judge might point out how the criticized response confuses key concepts by

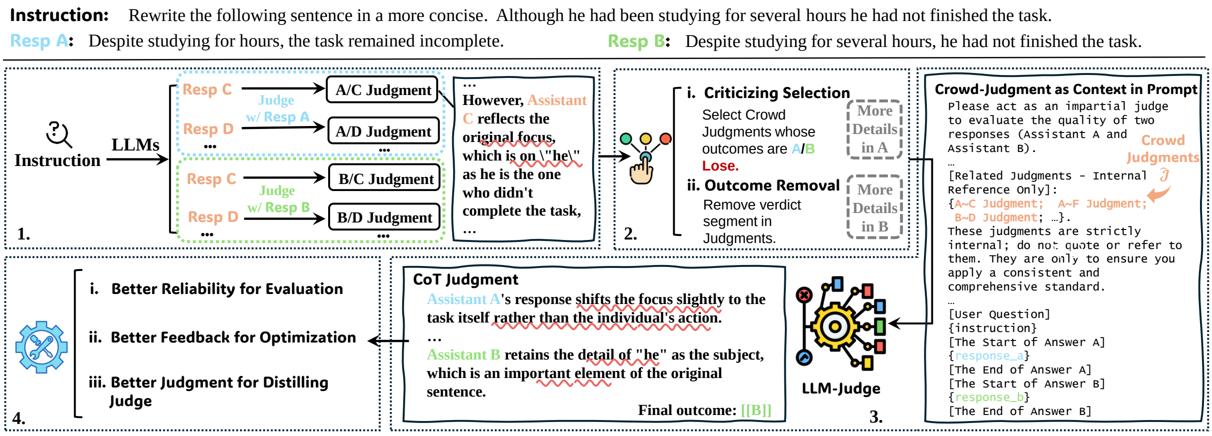

Figure 2: **Pipeline of our proposed crowd-based comparative evaluation.** For a given instance (x, y^A, y^B), we first use the LLM to generate crowd responses $\{y^i | i \in \{C, D, E, ...\}\}$ based on x. These responses are then compared with y^A and y^B to produce initial crowd judgments $\mathcal{J}$, which are subsequently refined into $\hat{\mathcal{J}}$ after selection and processing. Finally, $\hat{\mathcal{J}}$ are used as contextual input to evaluate the instance (x, y^A, y^B).

elaborating on specific errors in the definition and citing relevant theoretical principles. In contrast, judgments favoring the winning response tend to be brief, where the Judge might simply say, "this answer is correct" without further analysis. We also explore two alternative outcome-based strategies: **Praising Selection** (choosing only judgments where A/B wins) and **Balanced Selection** (maintaining an equal split between A/B wins and losses). However, as shown in our analysis (Table 4), both strategies perform worse than Criticizing Selection. Additionally, to mitigate bias from the outcome distribution from crowd judgments, we introduce **Outcome Removal**, where an LLM rewrites j_i to remove explicit outcome segments, ensuring a more neutral evaluation. After the selection and processing, we obtain $\hat{\mathcal{J}}$.

Notably, j_i includes CoT judgments not only of the (y^A, y^B) but also of y^i. Our pilot study shows that removing the CoT segments about y^i does not improve performance; therefore, we retain them to keep our approach simple.

3.4 Context-augmented Inference

The final judgment is derived by evaluating responses y^A and y^B conditioned on the instruction x, the criteria s, and the post-processed crowd judgments $\hat{\mathcal{J}}$:

$$j^\star = \mathcal{F}(y^A, y^B \mid x, s, \hat{\mathcal{J}}), \tag{4}$$

where the prompt template is provided in Appendix A. Notably, we distill $\{j^\star\}$ for training a smaller judge, whose performance surpasses the judge distilled from $\{j\}$, as demonstrated in Table 2. It proves that higher-quality CoT judgment has better distillation efficiency.

3.5 Extensive Application–Crowd Rejection Sampling in SFT

This subsection demonstrates the practicality of CCE by showcasing its extensive application in SFT. Rejection sampling has been proven an effective augmentation technique for SFT (Yuan et al., 2023; Zhu et al., 2023b). In a typical rejection sampling framework, given the task instruction and k generated responses, low-quality responses are filtered out, and the remaining high-quality ones are then used for fine-tuning. Traditionally, the Vanilla LLM-as-a-Judge selects the best response by comparing responses in pairs and choosing the one that wins most often. In contrast, CCE naturally adapts to the scenario that rejection sampling involves more than two responses, and we refer to it as *crowd rejection sampling*. During pairwise comparing any two candidate responses, we effectively utilize the additional $k-2$ responses as crowd responses as introduced in Subsection 3.2. After producing crowd judgments, it ensures a more detailed and consistent judgment. We validate the crowd rejection sampling in our subsequent experiment (in Table 3), where the integration of crowd responses consistently leads to more reliable and interpretable sampling, ultimately improving the overall performance of the fine-tuned model.

4 Experiments

4.1 Experimental Setup

We conduct a comprehensive evaluation of CCE across three tasks: testing preference benchmarks, judge distillation, and SFT rejection sampling.

Preference Benchmarks and Baselines. We adopt 5 preference benchmarks to test LLM-as-a-Judge, including REWARDBENCH (Lambert et al., 2024), HELPSTEER2 (Wang et al., 2024e), MTBENCH-HUMAN (Zheng et al., 2023), JUDGEBENCH (Tan et al., 2025), and EVALBIAS (Park et al., 2024). These benchmarks provide general instructions across a wide range of tasks with diverse responses and use accuracy to measure their evaluation performance. They each focus on different aspects. For example, REWARDBENCH covers a wider range of scenarios, while EVALBIAS focuses on various bias scenarios. We verify the generality of CCE on 5 LLMs and compare it against multiple baselines. In particular, we consider **Vanilla**, which uses the general LLM-as-a-Judge prompt implemented by REWARDBENCH; **Maj@16**, where we independently judge a case 16 times and take a majority vote of the outcomes; **Agg@16**, where instead of majority voting, the 16 individual judgments are fed back into the LLM to aggregate a final decision; **16-Criteria**, which incorporates 16 criteria with corresponding descriptions in the prompt as designed in Hu et al. (2024b) and Wang et al. (2024e); **LongPrompt**, where the LLM is explicitly directed to produce a longer CoT; and **EvalPlan**, in which an unconstrained evaluation plan is first generated based on the target case and then executed to derive the final judgment (Saha et al., 2025). Additional details on the preference benchmarks and baselines can be found in Appendix B.

Distilling CoT for Training Judge. We start with a large preference dataset and evaluate it using the Vanilla LLM-as-a-Judge and CCE under *GPT-4o-as-a-Judge*, producing two CoTs. We then pair each CoT with the original preference data to form two separate training sets, which we use to fine-tune a smaller LLM as a judge. The resulting judges' performance clearly reflects the quality and effectiveness of each CoT. We use **TULU3-preference** data as the distillation query while the preference benchmarks for evaluating the judge remain the same as previously introduced. Details of the training implementation are provided in Appendix C.

SFT Rejection Sampling. Firstly, we generate a pool of 4 responses based on a given task instruction to serve as the rejection sampling base. We compare Crowd Rejection Sampling against Random Selection and a Vanilla Rejection Sampling method to select the best response for fine-tuning.

We select two datasets of different scales, **LIMA** (Zhou et al., 2023) (1K) and **TULU3-SFT** (Lambert et al., 2025) (sample 10K), as instruction query. *GPT-4o* served as the judge LLM, while *Llama-3.1-8B* and *Qwen-2.5-7B* are used as base models for SFT. We then evaluate the generative ability of finetuned models using MTBENCH and ALPACAEVAL-2 (Dubois et al., 2024b). Details of the implementation are provided in Appendix D.

4.2 Experiment Result

In this section, we present our main results. The preference benchmark results are shown in Table 1, the efficacy of distilling CoT for training smaller judges is summarized in Table 2, and the training efficiency of SFT rejection sampling is reported in Table 3. These three objectives are concluded across various judge LLMs and downstream tasks. Our findings for each task are as follows.

Performance on Preference Benchmarks. Table 1 highlights **CCE consistently achieves state-of-the-art performance across all preference benchmarks**. First, it outperforms the Vanilla LLM-as-a-Judge, which already demonstrates reasonable reliability on multiple LLMs and benchmarks. Notably, with *Qwen 2.5-72B-Instruct* as the judge, our method achieves an 8.5 increase on REWARDBENCH and an overall average gain of 8.7.

Second, **CCE proves considerably more effective than common scaling strategies such as *Maj@16* and 16-Criteria**. Even with random selection, *Maj@16* underperforms CCE by an average of 1.9. Although *EvalPlan* offers a more response-aware reasoning process than *16-Criteria*, its effectiveness remains lower 2.0-3.7 than CCE. Simply generating longer CoT also falls short, indicating that scaling inference-time computation calls for a more nuanced approach.

Finally, CCE not only excels on REWARDBENCH, the most general benchmark, but also **outperforms alternatives on more challenging tasks** like JUDGEBENCH and EVALBIAS. Strategic crowd judgment selection further enhances performance compared to random selection. We adopt a "Criticizing Selection + Outcome Removal" strategy for

Model	RewardBench	HelpSteer2	MTBench Human	JudgeBench	EvalBias	Avg.
GPT-4o						
Vanilla	85.2	66.1	82.1	66.3	68.5	73.6
LongPrompt	86.9	67.3	81.8	63.5	70.5	74.0
EvalPlan	88.7	65.5	81.4	62.9	74.4	74.6
16-Criteria	87.3	69.1	82.8	66.6	73.7	75.9
Maj@16	87.9	68.9	82.4	68.6	75.5	76.7
Agg@16	88.1	68.7	82.6	67.2	77.9	76.9
CCE-random@16	91.2	69.5	83.1	68.9	80.1	78.6
CCE@16	**91.8**	**70.6**	**83.6**	**70.4**	**85.0**	**80.3**
Qwen 2.5 7B-Instruct						
Vanilla	78.2	60.7	76.1	58.3	57.4	66.1
CCE@16	**80.4**	**64.2**	**76.7**	**64.0**	**79.4**	**72.9**
Qwen 2.5 32B-Instruct						
Vanilla	87.4	**72.3**	79.0	68.9	71.1	75.7
CCE@16	**90.8**	72.1	**82.1**	**70.6**	**80.5**	**79.2**
Qwen 2.5 72B-Instruct						
Vanilla	85.2	**69.5**	79.5	68.3	68.5	74.0
CCE@16	**93.7**	68.5	**88.9**	**75.7**	**85.9**	**82.7**
Llama 3.3 70B-Instruct						
Vanilla	86.4	70.4	81.1	67.1	70.6	75.1
CCE@16	**91.7**	**71.3**	**83.5**	**69.7**	**79.2**	**79.1**

Table 1: Accuracy of LLM-as-a-Judge on pair-wise comparison benchmarks. CCE can consistently enhance the LLM-as-a-Judge's performance across 5 benchmarks, especially considerably outperforming other scaling inference strategies, like maj@16. The highest values are **bolded**. Here, *CCE-random* refers to replacing the "Criticizing Selection+Outcome-Removal Processing" with "Random Selection".

Model	# of Training Samples	RewardBench	HelpSteer2	MTBench Human	JudgeBench	EvalBias	Avg.
JudgeLM-7B (Zhu et al., 2023a)	100,000	<u>46.4</u>	<u>60.1</u>	64.1	32.6	**42.4**	<u>49.1</u>
PandaLM-7B (Wang et al., 2024d)	300,000	45.7	57.6	<u>75.0</u>	36.0	27.0	48.3
Auto-J-13B (Li et al., 2024b)	4,396	**47.5**	**65.1**	**75.2**	**50.9**	16.5	**51.0**
Prometheus-7B (Kim et al., 2024a)	100,000	34.6	30.8	52.8	9.3	11.7	27.8
Prometheus-2-7B (Kim et al., 2024b)	300,000	43.7	37.6	55.0	<u>39.4</u>	<u>39.8</u>	43.1
Llama-3.1-8B-Tuned							
Synthetic Judgment from Vanilla	10,000	66.8	56.0	71.6	<u>60.1</u>	34.2	57.7
Synthetic Judgment from Vanilla	30,000	**72.5**	<u>58.6</u>	<u>73.9</u>	50.4	<u>46.2</u>	60.3
Synthetic Judgment from CCE	10,000	69.7	<u>58.6</u>	72.7	**66.4**	38.7	**61.2**
Synthetic Judgment from CCE	30,000	<u>70.0</u>	**60.1**	**74.3**	50.3	**50.7**	<u>61.1</u>
Qwen 2.5-7B-Tuned							
Synthetic Judgment from Vanilla	10,000	68.1	55.6	70.7	<u>50.2</u>	38.4	56.6
Synthetic Judgment from Vanilla	30,000	71.4	56.2	75.1	48.2	54.7	61.1
Synthetic Judgment from CCE	10,000	68.8	56.7	71.3	49.8	40.2	57.4
Synthetic Judgment from CCE	30,000	<u>73.3</u>	<u>59.5</u>	<u>74.9</u>	50.1	<u>57.1</u>	<u>63.0</u>
Mix Synthetic Judgment from CCE&Vanilla	60,000	**74.1**	**60.7**	**76.6**	**61.6**	**60.6**	**66.7**

Table 2: Accuracy of Trained small LLM-as-a-Judge on pair-wise comparison benchmarks. Under the same preference pairs data, the model trained with judgments synthesized using CCE achieves more reliable evaluation results. The highest values are **bolded**, and the second highest is <u>underlined</u>.

our SOTA selection & processing strategy, which we discuss in detail in the following analysis. Moreover, we verify our generality of CCE on more base LLMs in Table 9.

Distilling CoT for Training Smaller Judges. Distilling preference evaluation capabilities from powerful LLMs to train smaller LLMs is a promising direction. Table 2 demonstrates that higher-quality CoT leads to more effective distillation, resulting in improved performance for smaller judge models. Fine-tuning small models (*e.g.*, *Llama 3.1-8B* and *Qwen 2.5-7B*) on the CoTs generated by CCE yields higher accuracy on all five benchmarks than using *Vanilla* CoTs. For instance, *Qwen 2.5-7B* trained on CCE's synthetic CoT judgments achieves up to 73.3% on RewardBench, surpassing Vanilla baseline by a notable margin of 1.9. Moreover, combining both *Vanilla* and CCE synthetic judgments further boosts performance, reaching 74.1% on RewardBench and 60.6% on EvalBias. This

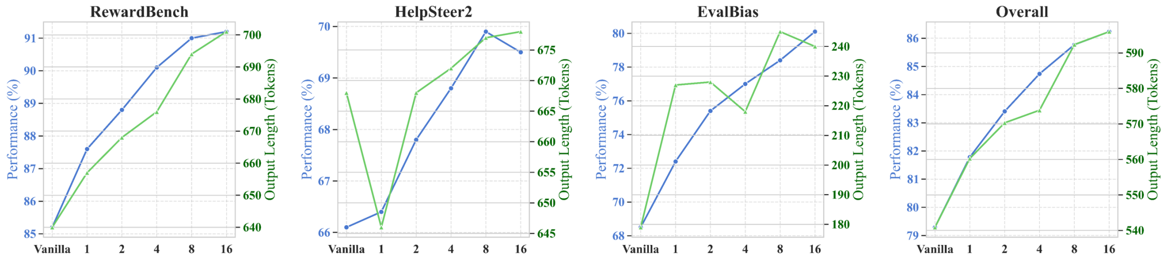

Figure 3: Evaluation performance under scaling crowd judgments in the context. As the number of crowd judgments grows, both accuracy and CoT length generally increase.

Rejection Sampling Method	MTBench	AlpacaEval-2
Llama 3.1 8B Base		
Instructions from LIMA # 1K		
Random Sampling	4.33	2.89/3.29
Vanilla Rejection Sampling	4.28	2.91/3.29
Crowd Rejection Sampling	**4.53**	**3.02/3.31**
Instructions from Tulu 3 # 10K		
Random Sampling	7.51	12.81/12.45
Vanilla Rejection Sampling	7.56	19.92/17.17
Crowd Rejection Sampling	**7.63**	**22.23/19.74**
Qwen 2.5 7B Base		
Instructions from LIMA # 1K		
Random Sampling	8.06	14.52/9.40
Vanilla Rejection Sampling	7.91	14.40/9.44
Crowd Rejection Sampling	**8.63**	**14.86/9.59**
Instructions from Tulu 3 # 10K		
Random Sampling	8.36	21.39/13.68
Vanilla Rejection Sampling	**8.46**	22.71/16.44
Crowd Rejection Sampling	8.41	**23.78/17.56**

Table 3: SFT Rejection Sampling Performance on the Instruction-Following Benchmark. The model fine-tuned with responses sampled using CCE demonstrates improved generative performance.

result suggests integrating diverse CoT can further enhance accuracy and generalization.

LLM-as-a-Judge can develop biases in various scenarios, such as favoring more verbose answers. This issue is particularly pronounced in smaller judge models. As shown in Table 2, even after fine-tuning on over 100K samples, many baseline models struggle to exceed 50% accuracy. This highlights the persistent challenge of evaluation bias. **Higher-quality and more comprehensive CoT distillation enhances the debiasing ability of smaller judge models**. These findings suggest that many biases stem from the model focusing on limited aspects of the responses rather than assessing them holistically.

Efficacy in SFT Rejection Sampling. As we can see in Table 3, Crowd Rejection Sampling proves effectiveness for both 1K and 10K data sizes, consistently **yielding better finetuning performances for two base LLMs**. CCE selects higher-quality responses compared to both Random Sampling and Vanilla Rejection Sampling, leading to consistent improvements in downstream instruction-following benchmarks on MTBench and AlpacaEval-2. For instance, with *Llama 3.1-8B* and the TULU3-SFT instructions, the fine-tuned model sees performance gains of up to 22.23/19.74 on AlpacaEval-2, compared to 19.92/17.17 under the Vanilla Rejection Sampling. This underscores the reliability of CCE in identifying higher-quality training examples.

Overall, the experiments confirm the flexibility and effectiveness of CCE in three key general scenarios. By **leveraging crowd-based context, scaling inference-time computation, and strategically guiding the CoT process**, CCE delivers consistent improvements over strong baselines.

4.3 Analysis Experiments

In this section, we conduct an in-depth analysis of the two core components of our method: crowd judgment selection & processing strategies, as well as inference scaling. We then directly examine whether the generated CoT is more comprehensive and provides a more detailed analysis of the responses under evaluation.

Selection & Processing Strategy. We compare Random Selection, Criticizing Selection, Praising Selection, and Balanced Selection. As shown in Table 4, Criticizing Selection yields the best results, followed by Balanced Selection, while Praising Selection performs even worse than Random Selection. This suggests that **lose-based judgments provide deeper insights into A/B comparisons, making criticism more informative**. Additionally, the **Outcome-Removal post-processing strategy substantially improves evaluation reliability**, likely because final verdicts lack valuable details while introducing biases into LLM decision-making.

Strategy	# of Selection Samples	RewardBench	HelpSteer2	MTBench Human	JudgeBench	EvalBias	Avg.
Random-Selection	8	91.0	69.9	82.6	68.7	78.4	78.1
Praising-Selection	8	86.6	64.2	81.5	67.1	77.7	75.4
Criticizing-Selection	8	91.2	69.2	83.0	68.9	79.1	78.3
Balanced-Selection	8	90.7	68.6	82.8	67.4	78.7	77.6
Outcome-Removal Random-Selection	8	**91.5**	69.9	83.0	69.4	79.5	78.7
Outcome-Removal Criticizing-Selection (Sota)	8	**91.5**	**70.1**	**83.2**	**69.5**	**79.9**	**78.8**
Random-Selection	16	91.2	69.5	83.1	68.9	80.1	78.6
Praising-Selection	16	87.0	68.4	82.0	67.1	77.9	76.5
Criticizing-Selection	16	90.8	69.7	83.0	69.6	82.9	79.2
Balanced-Selection	16	90.6	69.3	82.9	68.0	79.6	78.1
Outcome-Removal Random-Selection	16	91.7	69.7	83.2	70.0	81.5	79.2
Outcome-Removal Criticizing-Selection(Sota)	16	**91.8**	**70.6**	**83.6**	**70.4**	**85.0**	**80.3**

Table 4: Accuracy of CCE using different selection strategies on LLM-as-a-Judge benchmarks. Our proposed *Outcome-Removal Criticizing-Selection* consistently surpasses performances using other selection strategies during the test-time inference phase.

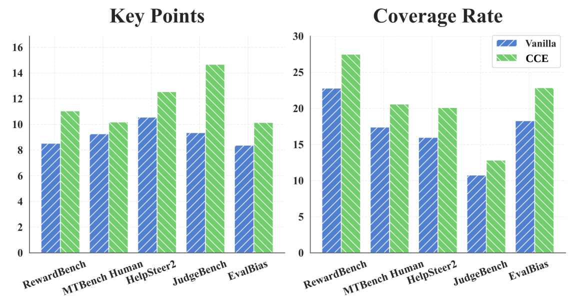

Figure 4: CoT Comparison. CCE's CoT consistently yields a higher average number of key points and a higher coverage rate across all benchmarks.

Inference Scaling. Figure 3 illustrates our analysis of how scaling crowd judgments influence evaluation outcomes. Measuring accuracy and the average token length of the CoT, three preference benchmarks are tested across different judgment counts and then averaged for an overall assessment. The implementation details are in Appendix E.

As shown in Figure 3, **both performance and output length generally increase as crowd judgments rise from 0 to 16**. RewardBench displays a clear upward trend, while HelpSteer2 dips briefly at 2 judgments before recovering. Averaging across benchmarks (rightmost panel) confirms that more crowd judgments lead to higher accuracy and longer CoT, consistent with the inference scaling observed in studies (Brown et al., 2024; Snell et al., 2025). Furthermore, we reexamine the Table 1 and find that **scaling test-time inference is a promising strategy for LLM-as-a-Judge**, as demonstrated by *GPT-4o-as-a-Judge*. This is especially evident in bias scenarios, where the Vanilla struggles, while scaling-inference-based baselines, including CCE, show substantial gains.

CoT Comparison. To more directly assess whether the CoTs generated by CCE are more comprehensive than those of the Vanilla approach, we perform two analyses: Key Points Counting and Coverage Rate. First, we use *GPT-4o* to parse and summarize each CoT, counting the key points to measure how thoroughly the CoT is. Second, we leverage the cross-attention mechanism from *Bart-base* to quantify the coverage rate–how thoroughly CoT covers details in the candidate responses. We introduce the details in the Appendix F.2.

As shown in Figure 4, CCE outperforms Vanilla across all benchmarks in key point counting and coverage rate. More key points indicate that our CoT examines the text from multiple angles, while a higher coverage rate reflects a more detailed analysis. These results demonstrate that **CCE offers a deeper and wider evaluation than Vanilla**.

Case Study. Table 5 presents a representative case. The vanilla is misled by fake information in Response A, causing it to overlook the Instruction and mistakenly rate Response A as more helpful. In contrast, the crowd judgment correctly identifies the error in Response A and informs subsequent evaluations. Additionally, our method produces a more detailed CoT thereby enriching the overall evaluation process, as evidenced by statements like "Assistant A does provide a brief explanation".

5 Conclusion

In this work, we tackle the shortcomings of LLM-as-a-Judge, which stem from CoT reasoning lacking comprehensiveness and detail, by drawing inspiration from human evaluative behavior. We introduce a novel crowd-based comparative evaluation framework that enriches the CoT process to unlock more comprehensive and reliable evaluations. By scaling inference more effectively, our method serves as an efficient alternative to traditional majority voting and criteria expansion. Importantly, we demonstrate that high-quality CoT judgments boost evaluation reliability and distilling efficiency across

Instruction: Classify this tweet from the categories of positive, negative, or neutral. Tweet: I'm so thankful for all the wonderful people in my life
Response A The sentiment of this social media comment is positive. The use of the word "grateful" implies appreciation and thankfulness, while "amazing experiences" suggest that these experiences have been very fulfilling or enjoyable. **Response B** Positive
Vanilla LLM-as-a-Judge: Assistant A's . . . . . . It highlights specific words and phrases in the tweet, such as "thankful" and "wonderful people," to justify the positive sentiment classification. This explanation adds depth and clarity to the response, making it more helpful and informative. Assistant B, ... the response less informative and helpful . . . . - Final Verdict: [[A]]
CCE: ***AC Judgment***: However, Assistant A makes a mistake by referencing words not present in the tweet, such as "grateful" and "amazing experiences," . . . . . . Assistant C also classifies the tweet as positive and provides a detailed explanation . . . . Assistant A . . . . . , but it inaccurately references words not present in the tweet, such as "grateful" and "amazing experiences." This detracts from the accuracy of the response and could potentially confuse the user. . . . . . . Assistant B is concise and correctly classifies the tweet as positive. However, it lacks any explanation or reasoning, which limits its helpfulness and depth. . . . . . . In comparing the two, Given the importance of accuracy and explanation in sentiment analysis, ... - Final Verdict: [[B]]

Table 5: A pairwise comparison case evaluated by different methods. Preference refers to right result and Preference refers to wrong result. We emphasize the noisy evaluation elements in orange, while highlighting the useful elements of the evaluation in limongreen.

multiple benchmarks, while broadening the scope of crowd-based evaluation applications.

Limitations

Progressive Self-Iteration Paradigm. A limitation of our work is that we do not explore self-iteration in this study, despite its potential for enhancing the evaluation process. Our method inherently allows for iterative refinement, which could be further extended into a progressive paradigm. We leave this direction for future work, aiming to investigate how iterative self-improvement can further enhance evaluation quality and robustness.

Selection based on LLMs. We identify that the quality of crowd judgments influences the CoT and explore a simple yet efficient selection strategy. We generate crowd responses using many LLMs, but we do not explore which LLM's crowd response has a greater influence on crowd judgment.

References

Sher Badshah and Hassan Sajjad. 2024. Reference-guided verdict: Llms-as-judges in automatic evaluation of free-form text. In arXiv.

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. 2022. Is attention explanation? an introduction to the debate. In Annual Meeting of the Association for Computational Linguistics, pages 3889–3900.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. In arXiv.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology, pages 1–45.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or llms as the judge? a study on judgement biases. In arXiv.

David Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In Annual Meeting of the Association for Computational Linguistics, pages 15607–15631.

DeepSeek-AI. 2024. Deepseek-v3 technical report. In arXiv.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024a. Length-controlled alpacaeval: A simple way to debias automatic evaluators. In arXiv.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. 2024b. Length-controlled alpacaeval: A simple debiasing of automatic evaluators. In Conference on Language Modeling.

Aparna Elangovan, Ling Liu, Lei Xu, Sravan Babu Bodapati, and Dan Roth. 2024. Considers-the-human evaluation framework: Rethinking human evaluation for generative large language models. In Annual Meeting of the Association for Computational Linguistics, pages 1137–1160.

Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024. Llm-based nlg evaluation: Current status and challenges. In arXiv.

Xinyu Hu, Mingqi Gao, Sen Hu, Yang Zhang, Yicheng Chen, Teng Xu, and Xiaojun Wan. 2024a. Are LLM-based evaluators confusing NLG quality criteria? In Annual Meeting of the Association for Computational Linguistics, pages 9530–9570.

Xinyu Hu, Mingqi Gao, Sen Hu, Yang Zhang, Yicheng Chen, Teng Xu, and Xiaojun Wan. 2024b. Are

LLM-based evaluators confusing NLG quality criteria? In Annual Meeting of the Association for Computational Linguistics, pages 9530–9570.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using mechanical turk to evaluate open-ended text generation. In Conference on Empirical Methods in Natural Language Processing, pages 1265–1285.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. In International Conference on Learning Representations.

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

Tae Soo Kim, Yoonjoo Lee, Jamin Shin, Young-Ho Kim, and Juho Kim. 2024c. Evallm: Interactive evaluation of large language model prompts on user-defined criteria. In Conference on Human Factors in Computing Systems.

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. 2025. Tulu 3: Pushing frontiers in open language model post-training. In arXiv.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. RewardBench: Evaluating reward models for language modeling. In arXiv.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. From generation to judgment: Opportunities and challenges of llm-as-a-judge. In arXiv.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024a. LLMs-as-Judges: A comprehensive survey on llm-based evaluation methods. In arXiv.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, hai zhao, and Pengfei Liu. 2024b. Generative judge for evaluating alignment. In International Conference on Learning Representations.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, Yuxuan Lai, Chongyang Tao, and Shuai Ma. 2024c. Leveraging large language models for nlg evaluation: Advances and challenges. In arXiv.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In Conference on Empirical Methods in Natural Language Processing, pages 2511–2522.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024. HD-eval: Aligning large language model evaluators through hierarchical criteria decomposition. In Annual Meeting of the Association for Computational Linguistics, pages 7641–7660.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. 2024. Generative reward models. In arXiv.

L. F. Obukhova and I. A. Korepanova and. 2009. The zone of proximal development. Journal of Russian & East European Psychology, 47:25–47.

Junsoo Park, Seungyeon Jwa, Ren Meiying, Daeyoung Kim, and Sanghyuk Choi. 2024. OffsetBias: Leveraging debiased data for tuning evaluators. In Findings of EMNLP, pages 1043–1067.

Jayr Pereira, Andre Assumpcao, and Roberto Lotufo. 2024. Check-eval: A checklist-based approach for evaluating text quality. In arXiv.

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

Chenglei Shen, Guofu Xie, Xiao Zhang, and Jun Xu. 2024. On the decision-making abilities in role-playing using large language models. In arXiv.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling test-time compute optimally can be more effective than scaling LLM parameters. In International Conference on Learning Representations.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Yuan Tang, Alejandro Cuadron, Chenguang Wang, Raluca Popa, and Ion Stoica. 2025. JudgeBench: A benchmark for evaluating LLM-based judges. In International Conference on Learning Representations.

Prapti Trivedi, Aditya Gulati, Oliver Molenschot, Meghana Arakkal Rajeev, Rajkumar Ramamurthy, Keith Stevens, Tanveesh Singh Chaudhery, Jahnavi Jambholkar, James Zou, and Nazneen Rajani. 2024. Self-rationalization improves llm as a fine-grained judge. In arXiv.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. In arXiv.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 37–42.

Tu Vu, Kalpesh Krishna, Salaheddin Alzubi, Chris Tar, Manaal Faruqui, and Yun-Hsuan Sung. 2024. Foundational autoraters: Taming large language models for better automatic evaluation. In arXiv.

Nico Wagner, Michael Desmond, Rahul Nair, Zahra Ashktorab, Elizabeth M. Daly, Qian Pan, Martín Santillán Cooper, James M. Johnson, and Werner Geyer. 2024. Black-box uncertainty quantification method for llm-as-a-judge.

Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2024a. Direct judgement preference optimization. In arXiv.

Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024b. Self-taught evaluators. In arXiv.

Xintao Wang, Yunze Xiao, Jen-tse Huang, Siyu Yuan, Rui Xu, Haoran Guo, Quan Tu, Yaying Fei, Ziang Leng, Wei Wang, Jiangjie Chen, Cheng Li, and Yanghua Xiao. 2024c. InCharacter: Evaluating personality fidelity in role-playing agents through psychological interviews. In Annual Meeting of the Association for Computational Linguistics, pages 1840–1873.

Yidong Wang, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2024d. PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization. In International Conference on Learning Representations.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024e. HelpSteer 2: Open-source dataset for training top-performing reward models. In Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. In arXiv.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024a. Evaluating large language models at evaluating instruction following. In International Conference on Learning Representations.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024b. Evaluating large language models at evaluating instruction following. In International Conference on Learning Representations.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. Generative verifiers: Reward modeling as next-token prediction. In Workshop on Mathematical Reasoning and AI at NeurIPS.

Qiyuan Zhang, Yufei Wang, Tiezheng YU, Yuxin Jiang, Chuhan Wu, Liangyou Li, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, Fuyuan Lyu, and Chen Ma. 2025. RevisEval: Improving LLM-as-a-judge via response-adapted references. In International Conference on Learning Representations.

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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In Conference on Neural Information Processing Systems.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In Conference on Neural Information Processing Systems.

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

Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Yongfeng Huang, Ruyi Gan, Jiaxing Zhang, and Yujiu Yang. 2023b. Solving math word problems via cooperative reasoning induced language models. In Annual Meeting of the Association for Computational Linguistics, pages 4471–4485.

Terry Yue Zhuo. 2024. ICE-Score: Instructing large language models to evaluate code. In Findings of the Association for Computational Linguistics: EACL 2024, pages 2232–2242.

A Prompt Template

We provide the prompt we used in this work for the experiment, as depicted in Figure 5. For Vanilla LLM-as-a-Judge (Figure 6), we deployed the prompt designed in MTBench, which is widely deployed in many works, *e.g.*, RewardBench. Notably, HelpSteer2 specializes in 5 aspects, so we replace the MTBench's aspects with these aspects when we test the method in HelpSteer2. Furthermore, we also present the prompts of baselines: *LongPrompt*(Figure 8) forces the CoT as long as possible; *16-Criteria* (Figure 7) incorporates 16

Benchmarks	Size	Focus
RewardBench	2,985	It covers multiple scenarios, including Chat, Chat-Hard, Safety, and Reasoning.
HelpSteer2	519	It provides multiple fine-grained dimensions for evaluation, like Helpfulness, Coherence, Correctness, Complexity, Verbosity.
MTBench Human	2,665	It provides multi-turn conversation for evaluation, and we filter the samples whose outcome is "Tie".
JudgeBench	350	It focuses on challenging response pairs spanning knowledge, reasoning, math, and coding
EvalBias	1,000	It tests the robustness of judges on various scenarios containing evaluation biases.

Table 6: The brief description of Preference Benchmarks for testing.

criteria and corresponding descriptions, which are designed in Hu et al. (2024b) and Wang et al. (2024e).

B Testing Preference Benchmark

B.1 Preference Benchmarks

As shown in Table 6, we give a brief introduction to preference benchmarks. Each of these benchmarks has its own strengths; thoroughly testing all of them and averaging the results is a reliable way to evaluate the method. Notably, we randomly sampled 1K cases from the training split of EvalBias since the size of the test split is 80 items, which is too small.

B.2 The Implementation of Generating Crowd Judgments

To generate crowd judgments, we produce a wide range of diverse responses. We employed several API-accessible and open-source LLMs to generate these responses based on the given instructions. Since diversity is crucial, we did not limit ourselves to only the most powerful models. Specifically, we used the following LLMs: *Qwen-2.5-0.5B-Instruct*, *Qwen-2.5-1.5B-Instruct*, *Qwen-2.5-3B-Instruct*, *Qwen-2.5-7B-Instruct*, *Llama-3.2-1B-Instruct*, *Llama-3.2-3B-Instruct*, *Llama-3.1-8B-Instruct*, *Mistral-Nemo10-Instruct-2407*, *Mistral-7B-Instruct*, *GPT-4o-mini*, and *GPT-4o*. Additionally, we applied two temperature settings (0.7 and 1.0) for each model. In principle, greater diversity in models and temperature configurations leads to improved performance.

Based on these crowd responses, we deployed the vanilla LLM-as-a-Judge to judge each crowd response with candidate response A/B separately using the judge LLM.

B.3 The Implementation of Baselines

For maj@16 and agg@16, we modify the temperature setting to 1.0 to promote more diversified responses. For other inferences in baselines, we set a unified temperature as 0.1.

B.4 The Implementation of Selection and Processing

For the selection strategy, we adopted "Criticize Selection" by choosing the crowd judgment where the outcome indicates that response A/B loses. For "Outcome Removal Processing," we used *GPT-4o-mini* to eliminate the outcome segment from the judgment with a temperature of 0. The prompt is:

> "*You are a helpful assistant. Specifically, I will provide you with the text quality judgment from an LLM-as-a-Judge evaluation of the responses from two AI assistants to an instruction. I need you to remove the final conclusion segments and only remain the evaluation analysis segments as soon as possible. ONLY OUTPUT the processed judgment.* "
>
> "**Judgment:* {judgment}*"

B.5 The Implementation of Inference

We tested our method on multiple LLMs-as-Judge, including *GPT-4o* (2024-08-06), *Qwen 2.5-7B-Instruct*, *Qwen 2.5-32B-Instruct*, *Qwen 2.5-72B-Instruct*, and *Llama 3.3-70B-Instruct*. We found that reliability and consistency of evaluation can be balanced when temperature= 0.1.

C Distilling CoT for Training Judge

C.1 Distilling Preference Source

We chose the TULU3-Preference-Mixture [1] as the preference data source. Specifically, we prompt the LLM-as-a-Judge to generate a CoT using the given instruction along with the chosen-rejected response pairs as input. Additionally, we experiment with two training sizes: random samples of 10K and 30K examples.

Distilling Inference. We use the *GPT-4o* as the Judge to produce the CoT, and the temperature setting is 0.1.

[1] https://huggingface.co/datasets/allenai/llama-3.1-tulu-3-8b-preference-mixture

```
Please act as an impartial and comprehensive judge evaluating the quality of two AI-generated responses (Assistant A and Assistant B) to the user's question. Your role is to determine which assistant better fulfills the user's needs according to multiple criteria, without focusing narrowly on any single aspect.

[Related Judgments - Internal Reference Only]: {crowd_judgments}.
These judgments are strictly internal; do not quote or refer to them. They are only to ensure you apply a consistent and comprehensive standard.

Evaluation Principles:
1. Holistic Assessment: Consider a wide range of factors—helpfulness, relevance, accuracy, completeness, clarity, depth, reasoning quality, creativity, level of detail—and maintain balance among these factors.
2. Comparative Contextualization: Although your final evaluation should be based solely on the two given responses, you may imagine or recall characteristics of other responses you've encountered in similar situations as background context. Implicitly inspire from the specific details in these related judgments for your current evaluation and refine the evaluation by identifying any missing dimensions or subpoints that are relevant for comparing Assistant A and Assistant B. Use this mental benchmark to ensure you apply consistent standards. Do not, however, introduce new information from outside sources into the final judgment. The goal is to remain aware of what an ideal response might look like, to ensure neither given answer is judged too narrowly.
3. Fairness and Impartiality: Do not be influenced by length, style, order of appearance, or assistant names. Evaluate purely on content quality.
4. Avoid External Biases: Rely only on the two given responses. Do not bring in outside data.

Instructions:
1. Please read the above related judgments carefully. Do not include the content under [Related Judgments] in your final output, nor should you explicitly reference them. These related judgments are for your internal reasoning only, serving as a mental benchmark to guide a comprehensive and fair evaluation. Use them as reference points to identify helpful aspects, potential weaknesses, or key criteria missed in prior assessments. Do not mention or hint at their existence in the final answer.
2. Evaluate both Assistant A's and Assistant B's answers thoroughly using the criteria above.
3. Compare their strengths and weaknesses, guided by your internal standard. Again, do not mention or allude to the internal references.
4. Begin your evaluation by comparing the two responses and provide a explanation.
5. End with your final verdict on which assistant performed better using the exact format:
  - [[A]] if Assistant A is better.
  - [[B]] if Assistant B is better.

[User Question]
{question}

[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

Figure 5: Prompt of Our Method.

```
Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses. Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better.

[User Question]
{question}

[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

Figure 6: Prompt of Vanilla LLM-as-a-Judge.

C.2 The Implementation of Training Judge

Base Models. To verify the generality of our method in Distilling CoT, we fine-tuned the preference data and corresponding CoT judgment in base LLMs: *Qwen 2.5-7B-Base* and *Llama 3.1-8B-Base*.

Training Setting. We trained the Base LLM with a *context length*$= 4,096$, *epochs*$= 3$, *batch size*$=$ 128,and *learning rate*$= 2e^{-5}$.

D SFT Data Selection

D.1 Synthetic Response Pool for Selection

To enhance the challenge and realism of the SFT Data Selection, we chose four LLMs with similar general generation capabilities as the base models for synthesizing responses. These are: *GPT-4o*, *DeepSeek-v3*, *Claude-3.5-Sonnet*, and *Qwen 2.5-72B-Instruct*. For inference, we set the temperature parameter to 0.7. We generate four responses for each instruction to serve as the basis for subsequent selection. The base instruction queries we used are two pools: LIMA and TULU3-SFT. LIMA [2] contains 1,000 instructions, which are regarded as high-quality; TULU3-SFT [3] contains 93.9K instruction-response pairs, and we randomly sampled 10K instructions as the query. The latter is the latest released multilingual dataset.

D.2 The Implementation of Rejection Sampling

Under the vanilla LLM-as-a-Judge approach, we perform pairwise comparisons among four responses, awarding a score of $+1$ to the winner of each matchup. After all comparisons, the response

[2] https://huggingface.co/datasets/GAIR/lima
[3] https://huggingface.co/datasets/allenai/tulu-3-sft-mixture

```
Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. our evaluation should consider factors such as helpfulness, relevance, accuracy, depth, creativity, level of detail of their responses, overall quality, readability, coherence, fluency, grammaticality, simplicity,  adequacy, faithfulness, non-hallucination, complexity, verbosity.
Helpfulness: How useful and helpful the response is.
Relevance: How relevant the response is.
Coherence: The response is self consistent in terms of content, style of writing, and does not contradict itself.
Simplicity: The response uses simple, easy to understand vocabulary and sentence structure that children can understand.
Complexity:  the model uses sophisticated language with elevated vocabulary that adults with advanced education or experts on the topic would use.
Verbosity: the response is wordy, giving a long winded and/or detailed reply.
Overall Qaulity: : It measures whether the target text is well-written and logical, and matches the required points of the source content.
Readability: It measures whether the target text is well-written, logical and clear.
Fluency: It measures whether individual sentences are grammatically correct and well-written.
Grammaticality: It measures whether the target text has no grammatical errors.
Adequacy: It measures how well the target text matches the required information of the source content.
Faithfulness: It measures whether the target text can be supported by the source content.
Non-hallucination: It measures whether the target text is verifiable according to the source content.
Accuracy: The response is based on facts, no hallucinations, no mistakes.
Depth: The response gives deep explanation to the query.
Creativity: How creative the response is.
Detailed: The response shows the detailed steps.

Begin your evaluation comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better.'

[User Question]
{question}

[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

Figure 7: Prompt of 16-Criteria LLM-as-a-Judge.

```
Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses. Begin your evaluation by comparing the two responses and provide a long explanation. This explanation should be as long as possible, to cover as many details as possible. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better.

[User Question]
{question}

[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

Figure 8: Prompt of LongPrompt LLM-as-a-Judge.

with the highest total score is selected. Building on this, our method incorporates the remaining two responses as "crowd responses" during each evaluation, allowing us to gather additional crowd judgments.

Base Judge Model. The base judge model is *GPT-4o*, and the temperature is set as 0.1.

D.3 The Implementation of Training SFT

Base Models. To verify the generality of our method in SFT data selection, we fine-tuned the instruction and selected response in base LLMs: *Qwen 2.5-7B-Base* and *Llama 3.1-8B-Base*.

Training Setting. We followed the common setup for SFT, with a *context length*$= 2048$, *epochs*$= 3$, *batch size*$= 128$,and *learning rate*$= 2e^{-5}$.

E Inference Scaling

The "Vanilla" setup has no crowd judgments, "1" includes a single judgment, and even-numbered settings split judgments evenly between A and B. We use *GPT-4o* as the judge and sample three times per setting to obtain the average result.

F CoT Comparison

F.1 Key Points Extraction

We use the Key points statistic to measure the richness of the CoT. Firstly, we use the *GPT-4o-mini* to summarize the CoT to aspects and corresponding sub-points. The summarization prompt is

> *"Extract the key evaluation aspects and detailed points mentioned in the text below. List the aspects and points in a strictly structured format:"*

"Example Input: 'The response is accurate but lacks creativity. It includes factual details but misses key arguments.' "

"Example Dictionary Output:" "- Aspect: Accuracy " " - Sub-point: Includes factual details " " - Sub-point: Misses key arguments " "- Aspect: Creativity " " - Sub-point: Lacks originality"

*"**Input**:"*

When we generate the summarized dictionary parsed output, we can get the total number of key points of each CoT.

F.2 Coverage Rate Compuataion

An attention-based approach computes mapping weights linking output tokens to input tokens. Interpretability research (Bibal et al., 2022; Vig, 2019) uses these weights to assess which input tokens influence the output. Our goal is to quantify how thoroughly CoT evaluates details in the target text, and attention-based computation provides a precise method for doing so.

Naturally, we used the *bart-base* [4] to compute the cross-attention between the target text and the generated CoT. We extract the cross-attention weights from the last layer of the decoder. By averaging these weights across attention heads and applying a threshold$=0.3$, it calculates a coverage rate—the fraction of the target text's tokens whose attention is above the threshold from the CoT.

G Study on Input Diversity

Diversity is a key characteristic of crowd responses, and in this section, we further investigate how different techniques influence the reliability of our proposed framework with respect to diversity. Specifically, in addition to our previously deployed setting, we incorporate an alternative setting: role-playing—a widely adopted prompting strategy for eliciting diverse responses from LLMs. In particular, we implement role-play using the 16 MBTI personality types as roles (Wang et al., 2024c). Following the setup described in Shen et al. (2024), we generate crowd responses under these role-playing conditions, while the overall LLM-as-a-Judge pipeline adheres to our CCE framework. For all experiments, we use *GPT-4o* as the base model.

As Table 7 shows, if we replace the original techniques with varying prompts, our *CCE* also works (compared to Vanilla) but has a relatively lower effectiveness than *CCE@16*. In conclusion, our method provides a simple yet effective means of enhancing response diversity, and we remain open to integrating additional diversity-enhancing techniques. In fact, this openness illustrates the flexibility and extensibility of our overall framework.

To this end, we design one metric to evaluate whether the diversity (or coverage) of crowd judgments expands as the number of n-shot examples increases: **PCA Coverage Volume**. This is an indirect measure of distributional complexity. It calculates how much variance is explained by the top principal components. A decline in the proportion of variance explained by the top two components indicates that the distribution is expanding into higher dimensions, suggesting that the samples are no longer constrained to a low-dimensional subspace—another signal of increased diversity.

As seen in Table 8, as the input samples increase, the coverage derived from crowd judgments expands.

H Generality of Our Method on Different Base LLMs

We supplement our paper with all the baselines on Qwen 2.5 7B and 72B to verify the generality of our method on different base LLMs.

[4] https://huggingface.co/facebook/bart-base

Methods	Reward Bench	HelpSteer2	MTBench Human	Judge Bench	EvalBias	Avg.
Vanilla	85.2	66.1	82.1	66.3	68.5	73.6
varying Prompts@16MBTIs	89.1	69.7	82.2	67.7	81.5	78.0
CCE@16 (Ours)	91.8	70.6	83.6	70.4	85.0	80.3

Table 7: Comparison of different varying prompts strategies on *CCE*.

Metrics	4 shots	8 shots	16 shots
PCA Coverage Volume(RewardBench)	12.2	43.6	54.5
PCA Coverage Volume(HelpSteer2)	15.7	40.6	50.0
PCA Coverage Volume(MTBench)	15.5	41.0	52.2

Table 8: PCA Coverage Volume across Shot Settings.

Model	Reward Bench	HelpSteer2	MTBench Human	Judge Bench	EvalBias	Avg.
Qwen 2.5 7B-Instruct						
Vanilla	78.2	60.7	76.1	58.3	57.4	66.1
LongPrompt	77.6	58.6	76.2	57.4	60.5	66.1
EvalPlan	79.5	61.7	75.7	60.9	60.2	67.6
16-Criteria	78.5	63.4	**76.8**	57.7	58.3	66.9
Maj@16	79.0	62.4	76.2	61.4	68.3	69.5
Agg@16	79.7	62.8	76.7	62.9	71.3	70.7
CCE@16	**80.4**	**64.2**	76.7	**64.0**	**79.4**	**72.9**
Qwen 2.5 32B-Instruct						
Vanilla	87.4	72.3	79.0	68.9	71.1	75.7
LongPrompt	87.7	71.8	79.4	69.1	70.7	75.7
EvalPlan	89.1	72.5	80.9	70.2	74.4	77.4
16-Criteria	87.5	71.2	79.4	69.4	71.5	75.8
Maj@16	88.3	72.4	79.7	69.1	73.0	76.5
Agg@16	88.9	**72.7**	80.2	69.5	76.2	77.5
CCE@16	**90.8**	72.1	**82.1**	**70.6**	**80.5**	**79.2**

Table 9: Accuracy of LLM-as-a-Judge on pair-wise comparison benchmarks.