Training Language Models to Critique With Multi-agent Feedback

Anonymous ACL submission

Abstract

Critique ability, a meta-cognitive capability of humans, presents significant challenges for LLMs to improve. While utilizing human annotation can enhance critique ability effectively, most recent works primarily rely on supervised fine-tuning (SFT) using critiques generated by a single LLM like GPT-4, which is more scalable and cost-effective. However, such model-generated critiques often suffer from inherent flaws due to the complexity of critique. Consequently, fine-tuning LLMs on these flawed critiques not only limits performance but also propagates errors into the learned model. To address this issue, we propose **MultiCritique**, a unified framework that leverages multi-agent feedback to improve critique ability in both the supervised fine-tuning (SFT) and reinforcement learning (RL) stages. In the SFT stage, MultiCritique aggregates high-quality multi-agent critiques through a fine-grained meta-critique mechanism. In the RL stage, preference critiques are constructed and refined by validating their contributions to revisions, thereby enhancing robustness of RL in improving critique ability. Based on MultiCritique, we construct SFT and RL datasets. Extensive experimental results on two benchmarks highlight the key benefits of our dataset, including superior quality, enhanced data efficiency, strong generalization on unseen tasks, and improvements in the general capability of LLMs. Notably, our fine-tuned 7B model significantly surpasses advanced 7B-13B models, approaching advanced 70B LLMs and GPT-4. Codes and datasets will be publicly available.

1 Introduction

The critique ability, *i.e.*, the capability to identify and refine flaws in responses, has been widely used to facilitate reliable automatic evaluation and self-improvement of LLMs (Lan et al., 2024; Wu et al., 2024). As a meta-cognitive capability (Toy et al., 2024; Wang and Zhao, 2024), critique ability requires LLMs to possess a deep understanding of user queries and evaluated responses beyond mere criticism (Kim et al., 2024; Zheng et al., 2023b). Therefore, it is challenging to improve the critique ability of LLMs (Lan et al., 2024; Lin et al., 2024).

While recent works demonstrate that utilizing human-annotated labels or answers could significantly improve the critique ability of LLMs (Wang et al., 2024a; Tang et al., 2025a; McAleese et al., 2024; Yu et al., 2024b; Liu et al., 2025), this approach faces scalability challenges due to the substantial demand for human annotation. In contrast, a more scalable and cost-effective way is to conduct the *i.e.,* Supervised Fine-Tuning (SFT) using critiques generated by a strong teacher model (GPT-4) (Li et al., 2024b; Kim et al., 2024). However, these model-generated critiques often suffer from inaccuracies stemming from the inherent biases of a single model and the complexity of the critique task (Lan et al., 2024; Liu et al., 2024e). As a result, LLMs fine-tuned on such datasets inherit these flaws, which are further propagated and potentially amplified during the SFT process.

To address this issue, we introduce **MultiCritique**, a unified framework designed to enhance the critique ability of LLMs by leveraging multi-agent feedback in both SFT and Reinforcement Learning (RL) stages, without any human annotation. First of all, to mitigate the limitations of critiques generated by a single LLM, we propose the **MultiCritique-SFT** pipeline (Figure 1 (Step 2)), which aggregates high-quality multi-agent critiques in a fine-grained manner. Specifically, multiple advanced LLMs first provide fine-grained critiques by critiquing responses at both sentence-by-sentence and cross-sentence levels. Then, meta-critique, as a specific communication mechanism, judges each critique unit by referencing multi-agent critiques. The meta-critique results are used to summarize a final critique by aggregating high-quality critique units while discarding flawed ones.

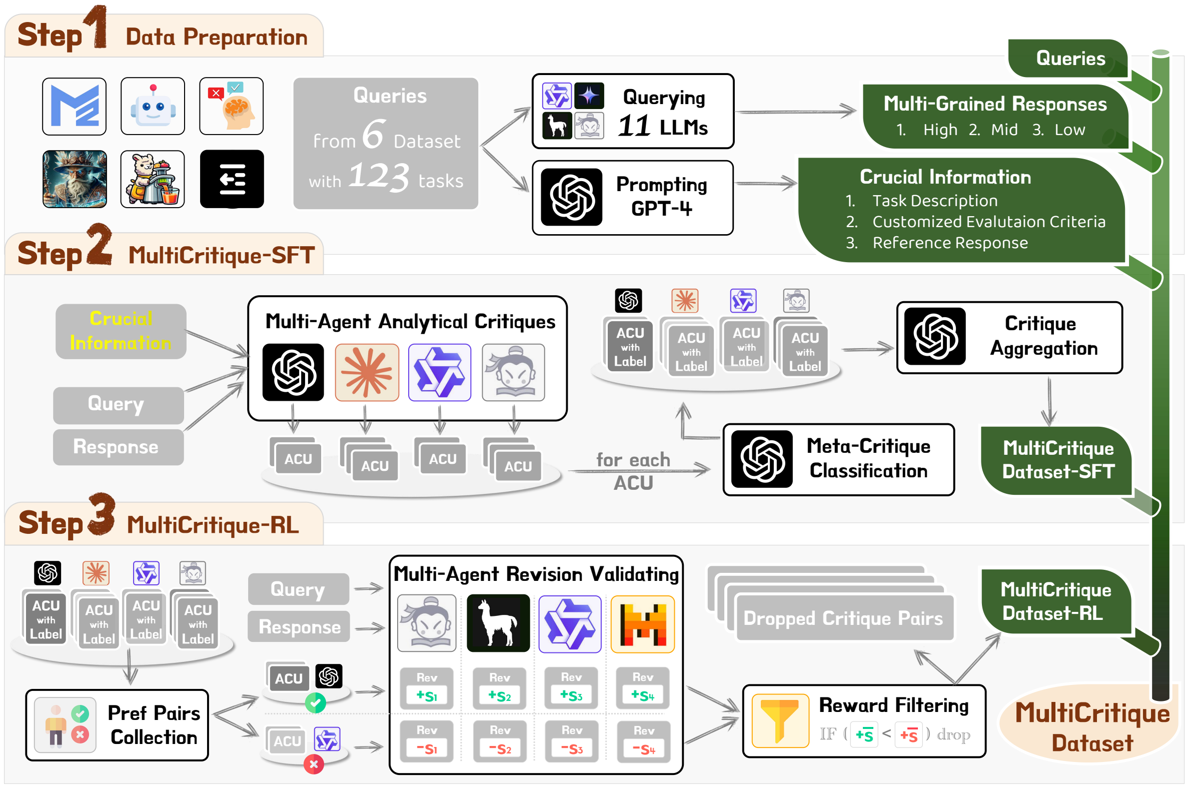

Figure 1: The overview of our proposed MultiCritique data generation pipeline. First, we prepare queries and evaluate responses and crucial information (Step 1). Then, MultiCritique-SFT pipeline aggregates high-quality multi-agent critiques (Step 2). Finally, MultiCritique-RL pipeline refines the preference critiques for the RL fine-tuning (Step 3). An ACU is a structured unit for identifying one specific flaw in responses (Section 3.2).

Second, to go beyond simple behavior cloning on model-generated critiques, we introduce the **MultiCritique-RL** pipeline, which constructs a high-quality preference critique dataset via multi-agent feedback, facilitating the effectiveness of RL in improving the critique ability of LLMs. Specifically, as shown in Figure 1 (Step 3), critiques are paired based on meta-critique evaluations obtained in SFT phase, with chosen critiques containing fewer and less severe flaws than rejected ones. The accuracy of preference critique is further improved by validating critiques' contributions to revisions across multiple models, retaining only pairs where the chosen critiques consistently lead to superior revisions. MultiCritique-RL is free from human annotations, demonstrating better scalability than recent works that rely on human-annotated labels or answers, like Critic-RM (Yu et al., 2024b).

Building on the MultiCritique framework, we construct the **MultiCritiqueDataset**. Extensive experimental results on CRITICEVAL (Lan et al., 2024) and CRITICBENCH (Lin et al., 2024) benchmarks demonstrate that several 7B-8B LLMs fine-tuned by SFT and RL stages on MultiCritique-Dataset significantly outperforms advanced 7B-13B baselines that trained on datasets 3-8x larger than ours. Notably, our model achieves performance close to advanced 70B LLMs and GPT-4. For instance, on CRITICBENCH, our model achieves 75.66% F1 score, compared to GPT-4's 78.75%. In addition, our proposed MultiCritique exhibits superior data efficiency during training, surpassing previous baselines by a factor of 2.15-4.22. Ablation studies further validate the positive contributions of our designs in the MultiCritique framework. Moreover, our SFT dataset exhibits strong generalization on unseen tasks and enhances the general ability of LLMs, underscoring its utility and robustness.

2 Related Work

Critique Ability of LLMs The critique ability of LLMs has been applied in three key areas: (1) Reliable Automatic Evaluation (Saunders et al., 2022; Zheng et al., 2023a); (2) Self-improvement of LLMs (Yuan et al., 2024; Wu et al., 2024); and (3) Robust Reward Modeling (Ye et al., 2024; Zhang et al., 2024b; DeepSeek-AI et al., 2024; Liu et al., 2024a; Vu et al., 2024; Zeng et al., 2024).

So far, two primary approaches have been employed to enhance the critique ability of

LLMs: (1) **Human Annotation:** This method has demonstrated effectiveness in improving critique ability by using human-annotated labels or critiques (Wang et al., 2023; Chen et al., 2025; Wang et al., 2024a; Yu et al., 2024b; Tang et al., 2025a), as exemplified by CriticGPT (McAleese et al., 2024) and DeepSeek-GRM (Liu et al., 2025); (2) **Distillation:** This method enhances the critique capability of LLMs using model-generated critiques (Kim et al., 2024; Wang et al., 2024b), such as UltraFeedback (Cui et al., 2023), Auto-J (Li et al., 2024b). However, these approaches face a dilemma: (1) human annotation incur prohibitively high costs that severely limit scalability; (2) model-generated critiques often suffer from quality issues. In contrast, we utilize multi-agent feedback to improve critique quality in a more scalable way.

Preference-based Reinforcement Learning Reinforcement learning (RL) algorithms are widely utilized to guide LLMs to generate responses that are more preferred by humans (Schulman et al., 2017; Yang et al., 2024b). It typically employs a reward model as a proxy for human judgment, learning through human-annotated pairwise comparison of responses, often called Reinforcement Learning from Human Feedback (RLHF) (Stiennon et al., 2022; Ouyang et al., 2022). Existing works utilize human-annotated or model-generated preference dataset to improve critique ability (Hu et al., 2024; Wu et al., 2024; Liu et al., 2025). These models highly rely on the high-quality human-annotated preference datasets, while our proposed MultiCritique-RL pipeline refines preference critiques with multi-agent feedback to enable robust RL fine-tuning in a more scalable way.

Multi-Agent Framework Current multi-agent frameworks are widely used in two applications: (1) **LLMs alignment:** extensive researches (Du et al., 2023; Khan et al., 2024) have proven the effectiveness of multi-agents in enhancing LLM's alignment through fostering more divergent thinking and aggregating the diverse opinions of multiple LLMs (Liang et al., 2024; Zhang et al., 2024a; Ji et al., 2024), such as Stable Alignment (Liu et al., 2024b) and Arena Learning (Luo et al., 2024). In contrast, our work focuses on improving the critique ability of LLMs, addressing a distinct challenge. (2) **LLM-based evaluation:** recent works employ multi-agent frameworks for reliable automatic evaluation, such as ChatEval (Chan et al., 2023), PoLL (Verga et al., 2024) and PRD (Li et al., 2024c). Unlike these works, which operate primarily during the inference stage, our work improve the critique quality by utilizing multi-agent feedback to enhance LLM critique ability during both the SFT and RL stages.

3 Method

3.1 Data Preparation

As shown in Figure 1 (Step 1), we first collect diverse queries, evaluated responses as well as crucial information for simplifying critique task.

Diverse Queries Collection We compile 10.7K queries with 123 diverse tasks from several well-established datasets, including alignment datasets (OpenHermes-2.5, DEITA (Liu et al., 2024c) and OpenAssistant (Köpf et al., 2023)), mathematical and coding datasets (MetaMathQA (Yu et al., 2024a) and CodeFeedback (Zheng et al., 2024)), and critique dataset Auto-J (Li et al., 2024b). More details are listed in Table 13.

Diverse Responses Collection Then, eleven LLMs with different capabilities are prompted to generate responses, which are coarsely evaluated using a robust reward model. We select low-, medium- and high-quality responses for each query, ensuring the uniform response quality distribution. In total, 10.7K×3=32.1K query-response pairs are collected. More implementation details are placed in Appendix C.1.

Crucial Information Collection Once the query-response pairs are collected, we sequentially elicit three crucial information to simplify the critique task, facilitating robust critique generation: **(1) Task Description**: Preliminary findings indicate that LLMs often misinterpret the query objectives. By prompting GPT-4 to describe the task, we can mitigate this issue to some extent; **(2) Customized Evaluation Criteria**: Once the task description is obtained, we propose generating customized two-tier structure evaluation criteria tailored to each query to guide effective critiques (Liu et al., 2024f). The first and second tiers outlines the fundamental and customized evaluation criteria. Each criterion is structured with a name, description and level of importance; **(3) Reference Response**: Finally, we generate reference responses that satisfy all customized evaluation criteria.

Following these three steps, we collect $\mathbb{N}$=32.1K samples $\{(q_i, r_i, \mathcal{CI}_i)\}_{i=1}^{\mathbb{N}}$, where $q_i, r_i, \mathcal{CI}_i$ represents the i-th query, evaluated response and corresponding crucial information in the dataset.

3.2 MultiCritique-SFT Pipeline

After collecting query-response and crucial information, we propose MultiCritique-SFT data generation pipeline (Figure 1 (Step 2)) to mitigate flawed critiques generated by a single LLM. This pipeline consists of three key stages: (1) collecting detailed analytical critiques from multiple LLMs (Multi-Agent Analytical Critique); (2) judging multi-agent critiques through a fine-grained meta-critique process (Meta-Critique Classification); and (3) aggregating multi-agent critiques into a final critique by summarizing accurate critique content while discarding flawed ones (Critique Aggregation). The details are described below.

Multi-agent Analytical Critiques To ensure diverse and robust critique generation, we employ four LLMs to simultaneously critique responses: GPT-4, Claude-1-instant, Qwen-1.5-72B-Chat and InternLM2-20B-Chat, all of which exhibits strong performance on the CriticEval benchmark (Lan et al., 2024). Each LLM structures analytical critiques by performing both sentence-by-sentence and cross-sentence critique. These critiques are organized into a list of **A**nalytical **C**ritique **U**nits (ACUs), which are designed to identify and address specific flaws in the evaluated responses. An ACU consists of five key components: (1) the location[1]; (2) the description; (3) suggestions for revision; (4) the criteria type; and (5) the severity. These structured ACUs not only enhance the transparency and interpretability of the critiques but also facilitate a robust meta-critique process (Sun et al., 2024).

Meta-Critique Classification This step can be seen as a specific multi-agent communication mechanism in critique generation task. Unlike previous multi-agent debate framework (Chan et al., 2023; Liu et al., 2024b), our preliminary study observes that critiques can influence each other and reduce diversity. Therefore, we maintains critique independence by using a meta-critique model judge the ACUs given all multi-agent critiques (Lan et al., 2024), thereby enhancing both diversity and comprehensiveness.

Specifically, GPT-4 evaluates each ACU within the context of multi-agent critiques, classifying it into one of seven quality categories, rather than directly judging the critique as a whole (Lan et al., 2024). These quality categories are determined by human annotators and are associated with severity scores ranging from 1 to 5 (Appendix J.5). For one model-generated analytical critique, the accumulated severity scores of its ACUs could indicate the overall quality of critiques, whereas a higher accumulated severity score indicates lower quality. While we have proven that other LLMs like Claude and Qwen2.5 are effective for meta-critique tasks in Appendix G, GPT-4 is selected due to its verified performance (Lan et al., 2024).

Critique Aggregation Finally, GPT-4 aggregates these ACUs into a comprehensive analytical critique by retaining and merging accurate ACUs from multi-agent while modifying or excluding those identified as flawed. We also prompt GPT-4 to generate an overall description and judgment score for the evaluated response. The final analytical critique, description and judgment score are concatenated as the final critique, denoted as $\mathcal{C}$.

By following previous steps of MultiCritique-SFT, we construct a supervised fine-tuning dataset MultiCritiqueDataset-SFT, consisting of $\mathbb{N}$=32.1K samples: $\{(q_i, r_i, \mathcal{CI}_i, \mathcal{C}_i)\}_{i=1}^{\mathbb{N}}$. Besides, as shown in Table 1, MultiCritique exhibits superior diversity in critiques, as evidenced by both higher ACUs number and broader coverage of evaluation criteria aspects.

Source of Critique	ACUs Number	Criteria Coverage
MultiCritique	**4.08**	**2.84**
GPT-4	4.02	2.76
Claude	3.40	2.70
Qwen	4.01	2.79
InternLM2	3.88	2.75

Table 1: Diversity of Generated Critiques.

3.3 MultiCritique-RL Pipeline

Beyond the behavior cloning on the supervised dataset, we also conduct the MultiCritique-RL data generation pipeline to construct the preference critiques, facilitating improvements by RL. Our solution automatically collects high-quality preference critique pairs by validating the multi-agent revision qualities given critiques, rather than using human-annotated judgment labels or answers in recent works, like SFR-Judge (Wang et al., 2024a), Critic-RM (Yu et al., 2024b) and SCRIT (Tang et al., 2025a), exhibiting better scalability. As shown in Figure 1 (Step 3), the MultiCritique-RL pipeline involves following two steps.

[1] We introduce a pre-processing step to label sentences in evaluated responses, as detailed in Appendix C.1.

Preference Pairs Collection In MultiCritique-SFT, the quality of ACUs in each analytical critique is measured by meta-critique process. Therefore, for each sample i, it is easy to identify a pair of critiques: the chosen critique C_{i,j_+} and the rejected critique C_{i,j_-}. The pairing is determined based on a significant performance gap between the two critiques, quantified by the difference in their accumulated severity scores.

Multi-Agent-Revision-Validating (MARV) Previous works (Lan et al., 2024) demonstrate that meta-critique is much more challenging than critiquing responses, might leading to the noise in the preference dataset. To address this issue, we propose the Multi-Agent-Revision-Validating (MARV) pipeline to refine the preference dataset, which validates the critique's contributions to the revision quality, thereby circumventing the complex meta-critique task. Specifically, four independent 7B LLMs first revise the evaluated response based on each critique, each performing eight revisions, resulting in a total of 4×8=32 revisions. The use of multiple LLMs ensures both reliability and robustness by reducing potential biases introduced by a single model. These revisions are then evaluated using the advanced reward models. Finally, preference critiques are reserved if the chosen critique's average reward score is higher than the rejected critique's score. Please refer to Appendix C.1 for more implementation details.

In summary, we construct the MultiCritique Dataset-RL, consisting of $\mathbb{M}$=19.7K samples: $\{(q_i, r_i, \mathcal{CI}_i, C_{i,j_+}, C_{i,j_-})\}_{i=1}^{\mathbb{M}}$.

4 Experimental Setup

4.1 Implementation Details

This paper presents a systematic experiments for enhancing the critique capabilities of 7B-8B LLMs (InternLM2, Llama3, and Qwen2.5), with a focus on inference efficiency. Our fine-tuning consists of two sequential stages: (1) **SFT Stage:** To ensure a deep understanding of the critiques, LLMs are trained to predict the concatenation of the crucial information $\mathcal{CI}_i$ and final critiques $\mathcal{C}_i$ by minimizing Maximum Likelihood Estimation (MLE); (2) **RL Stage:** A reward model is first trained to classify chosen and rejected analytical critiques C_{i,j_+}, C_{i,j_-} by optimizing the focal ranking loss (Cai et al., 2024). Then, the SFT model is optimized by PPO (Schulman et al., 2017), guided by this reward model. For comprehensive implementation details, please refer to Appendix C.

4.2 Benchmarks and Evaluation Metrics

We utilize CRITICEVAL (Lan et al., 2024) and CRITICBENCH (Lin et al., 2024) benchmarks to evaluate the critique ability of LLMs.

CRITICEVAL evaluates critique ability across 9 tasks, covering alignment, common NLP and reasoning capabilities. We first evaluate the critique quality: **(1) The objective feedback evaluation** ($\boldsymbol{F}_{\text{obj.}}$) calculates the Spearman correlation between LLM and human judgments on response quality; **(2) The subjective feedback evaluation** ($\boldsymbol{F}_{\text{sub.}}$) involves GPT-4 assessing the textual critiques quality. These scores range from 1 to 10. Furthermore, we evaluate the quality of revisions generated by critiques as the indicator of critique quality: **(1) The objective revision evaluation** ($\boldsymbol{R}_{\text{obj.}}$) measures the average Pass Rate of five LLMs' revisions for mathematical and coding questions. CRITICEVAL evaluates the chain-of-thought (CoT) and program-of-thought (PoT) approaches for mathematics. For coding tasks, it compares two settings: with execution (CodeExec) and without execution results (CodeNE); **(2) The subjective revision evaluation** ($\boldsymbol{R}_{\text{sub.}}$) is assessed by GPT-4, with scores ranging from 1 to 10. Importantly, CRITICEVAL has proven a strong correlation between GPT-4 and humans in subjective evaluation, given the human-annotated critiques as references. Note that the reliability of subjective evaluation has been well proven (Lan et al., 2024) with the help of the human-annotated reference critiques.

CRITICBENCH consists of 3,825 queries and evaluated responses for five challenging reasoning tasks: (1) mathematical reasoning; (2) commonsense reasoning; (3) symbolic reasoning; (4) algorithm reasoning; and (5) code generation. The correctness of the evaluated responses is annotated based on the ground-truth responses. The F1 score is used to evaluate whether LLMs can accurately identify the correctness of evaluated responses. Since critique-tuned LLMs cannot utilize few-shot samples, all models are tested under the zero-shot setting to ensure a fair comparison.

4.3 Baseline Datasets and Models

Baseline Datasets Three critique datasets constructed by GPT-4 are compared : (1) Auto-J (Li et al., 2024b); (2) UltraFeedback (Cui et al., 2023) and (3) Feedback-Collection (Kim et al., 2024).

Models	CriticEval				CriticBench					
	$\boldsymbol{F}_{\text{obj.}}$	$\boldsymbol{F}_{\text{sub.}}$	$\boldsymbol{R}_{\text{obj.}}$	$\boldsymbol{R}_{\text{sub.}}$	**Math**	**Comm.**	**Symb.**	**Algo.**	**Code**	**Overall**
Closed-source LLM										
GPT-3.5-Turbo	61.47	5.06	15.54	6.20	62.01	50.22	64.49	46.15	73.13	51.44
GPT-4-Turbo	76.09	7.90	26.88	7.71	92.55	71.56	90.75	63.51	91.36	78.75
70B instruction-tuned LLMs										
Qwen2-72B-Instruct	75.44	7.83	23.89	7.21	82.15	59.64	78.22	51.35	85.81	75.86
Llama3-70B-Instruct	73.28	7.05	21.97	6.90	82.35	60.22	86.31	54.90	86.16	76.80
7B-13B instruction-tuned and critique-tuned LLMs										
Qwen2-7B-Instruct	50.49	5.47	16.21	5.42	52.25	26.20	29.55	9.35	70.23	45.66
Llama3-8B-Instruct	37.20	5.04	17.69	5.98	78.33	62.64	62.05	62.19	76.41	70.71
CritiqueLLM-6B	35.52	3.88	11.34	2.71	66.37	62.53	63.00	62.83	65.12	67.73
Themis-8B	38.07	4.07	14.43	2.63	53.34	27.35	33.16	35.64	44.33	42.57
Prometheus-7B (Ours)	38.06	2.54	18.78	4.57	59.43	54.28	31.98	22.82	67.07	54.25
TIGERScore-7B	0.64	3.24	12.89	4.36	66.62	38.21	44.52	27.34	52.49	52.83
TIGERScore-13B	-2.31	3.39	15.45	4.54	68.91	45.47	53.04	42.86	44.13	56.28
UltraCM-13B	21.51	4.12	16.19	4.85	76.54	35.59	50.51	25.17	54.73	59.39
Auto-J-13B	36.05	4.21	17.69	5.62	80.02	50.64	53.06	52.06	75.61	67.41
InternLM2-7B-Chat-SFT	38.78	3.73	14.48	3.32	27.08	17.48	18.82	14.29	36.13	24.71
+ MultiCritiqueDataset-SFT	58.15	5.71	**19.33**	5.78	**89.49**	**62.60**	57.04	51.85	**79.51**	75.15
+ MultiCritiqueDataset-RL	**63.28**	**6.07**	**19.26**	**6.33**	**89.36**	60.56	**61.51**	**57.76**	**79.32**	**75.66**

Table 2: Overall experimental results. The best performance for 7B-13B critique-tuned models is highlighted in bold. Results comparable to the best performance (no more than 0.2% performance gap) are also highlighted.

Baseline Models We evaluate advanced closed-source and open-source LLMs, including GPT-3.5-Turbo and GPT-4, Llama3 and the Qwen2 (Yang et al., 2024a) series. We also assess critique-tuned LLMs: (1) Themis (Hu et al., 2024); (2) TIGERScore (Jiang et al., 2023); (3) Auto-J (Li et al., 2024b); (4) UltraCM (Cui et al., 2023); (5) CritiqueLLM (Ke et al., 2024); and (6) Prometheus (Kim et al., 2024).

More details about our evaluation setup can be found in Appendix C. Some baselines are excluded, and reasons are detailed in Appendix E.

5 Experimental Results

This section demonstrates the experimental results of our proposed datasets (Section 5.1), comparison with baseline datasets (Section 5.2), and its scaling phenomenon (Section 5.3).

5.1 Overall Experimental Results

Table 2 demonstrates that both SFT and RL fine-tuning stages on our proposed MultiCritique-Dataset significantly improves the critique ability of the InternLM2-7B-Chat-SFT model, outperforming other 7B-13B baselines and GPT-3.5-turbo. Besides, experimental results on other advanced LLMs, like Llama3 and Qwen2.5 models, also demonstrate significant improvements, which are placed in Appendix H due to the page limitation. Specifically, the SFT and RL stages yields absolute improvements of 19.8% and 6.3% on CriticEval subjective feedback evaluation ($\boldsymbol{F}_{\text{sub.}}$), and our fine-tuned model even approaches advanced 70B LLMs and GPT-4 on the CriticBench benchmark, highlighting its competitive performance. Note that improvement brought by MultiCritique-RL on CriticBench is modest, which can be primarily attributed to the inherent difficulty of the benchmark. CriticBench consists of highly challenging logical reasoning tasks, as evidenced by the fact that GPT-4 achieve only 78.75% accuracy, suggesting that current approaches are nearing a performance ceiling on this dataset.

Models	CriticEval ($\boldsymbol{F}_{\text{sub.}}$)			
	Math CoT	**Math PoT**	**Code Exec**	**Code NE**
SFT	4.64	5.21	4.72	**5.56**
RL	**5.70**	**6.21**	**4.87**	5.33

Table 3: Detailed results for mathematical and coding tasks in CriticEval.

Moreover, our analysis reveals an intriguing phenomenon: while the RL-fine-tuned model achieves a marginally lower objective revision score ($\boldsymbol{R_{obj.}}$) in CriticEval for mathematical and coding tasks compared to the SFT model (19.26 ≈ 19.33), it demonstrates substantial improvements in textual critique quality ($\boldsymbol{F}_{\text{sub.}}$), as evidenced by the experimental results in Table 3. This discrepancy highlights two insights: (1) RL optimization effectively

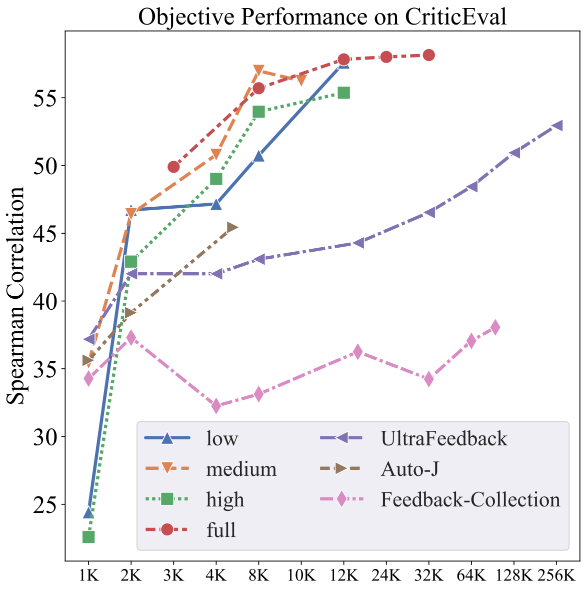

Figure 2: The correlation between the number of SFT training samples **(1K-256K)** and critique ability. *low, medium, high* and *full* represent the models that are trained on critiques in our SFT dataset for low-, medium-, high-quality, and all three response qualities (full). Please refer to Appendix B for complete results.

enhances critique quality across most mathematical and coding tasks, and (2) the evaluation of revisions in these domains exhibits inherent instability.

5.2 Comparison with Baseline Datasets

Table 4 demonstrates that the our SFT dataset significantly outperforms baseline datasets, with 21.48% and 22.50% average performance gain on CRITICEVAL and CRITICBENCH, respectively. Notably, despite the fact that Feedback-Collection and UltraFeedback are 3-8x larger than ours in scale, respectively. This performance suggests that our dataset exhibits superior quality compared to these larger-scale alternatives.

Models	CRITICEVAL				CRITIC BENCH
	$F_{\text{obj.}}$	$F_{\text{sub.}}$	$R_{\text{obj.}}$	$R_{\text{sub.}}$	**Overall**
Base Model	38.78	3.73	14.48	3.32	24.71
+ Auto-J	45.44	3.56	14.63	3.47	67.76
+ UF	52.95	4.42	15.81	3.54	58.67
+ FC	33.00	2.54	18.78	4.57	49.76
+ Ours	**58.15**	**5.71**	**19.33**	**5.78**	**75.15**

Table 4: Comparison between our MultiCritiqueDataset-SFT and baseline datasets. **FC** and **UF** indicates the Feedback-Collection and UltraFeedback datasets.

5.3 Scaling Phenomenon on Datasets

Figure 2 illustrates three findings: (1) **Scaling Behavior:** Critique ability improves steadily with more samples, leveling off beyond 12K samples; (2) **Superior Data Efficiency:** The model trained on our SFT dataset consistently outperforms those trained on other datasets across most data scales. Besids, models trained on 3K samples in our dataset (≈ \$890) surpasses baselines that require 100K-257K samples (\$1,915-\$3,758 for UltraFeedback and Prometheus)[2], demonstrating 2.15-4.22x improvement in data efficiency. Therefore, our approach is well-suited for resource-constrained organizations, as it significantly reduces the overall computational budget required to achieve competitive critique capabilities; (3) **Generalization Advantage:** In most cases, models trained on full response quality generalize better than those trained on individual types, indicating better generalization brought by diverse response qualities.

6 Analyze

This section conducts comprehensive studies on our dataset: (1) Effectiveness of MultiCritique-SFT pipeline; (2) Crucial Information for critique simplification; (3) MARV in refining preference critiques; (4) Improvements on general capability; and (5) Generalization to unseen tasks.

Ablation Study on MultiCritique-SFT We evaluate the MultiCritique-SFT pipeline by fine-tuning InternLM2-7B-Chat-SFT with analytical critiques generated by individual models within the MultiCritique-SFT pipeline.[3] Table 5 demonstrates that models fine-tuned with critiques generated by MultiCritique-SFT outperforms those optimized with critiques from individual models. Furthermore, there is a notable performance gap in the critiques generated by different models. For example, GPT-4 generates higher-quality critiques than Qwen-1.5-72B-Instruct and InternLM2-20B-Chat, while all three models surpass Claude-1-instant. These findings align with the evaluation results from CRITICEVAL (Lan et al., 2024).

SFT Models	CRITICEVAL			
	$F_{\text{obj.}}$	$F_{\text{sub.}}$	$R_{\text{obj.}}$	$R_{\text{sub.}}$
MultiCritique-SFT	**59.74**	**5.17**	**20.92**	**6.05**
GPT-4-Turbo	58.53	5.07	18.39	5.87
Claude-1-instant	56.77	5.01	19.00	5.79
Qwen-1.5-72B	57.30	4.89	17.74	5.81
InternLM2-20B	54.73	4.84	17.52	5.82

Table 5: Ablation study on MultiCritique-SFT.

[2] The costs are calculated based on the total number of input and output tokens of GPT-4 API.

[3] Please refer to Appendix C.2 (**Ablation Study in SFT**) for more details about this experimental setup.

Ablation Study on Crucial Information We assess the impact of three crucial information components on critique simplification by systematically removing each during training and evaluating their effects. Table 6 shows that removing each crucial information leads to a significant performance drop on most metrics in CriticEval. This observation suggests that crucial information plays a vital role in simplifying the critiques. Interestingly, training without evaluation criteria (w/o Criteria) leads to the best performance on the subjective revision evaluation in CriticEval ($\boldsymbol{R}_{\text{sub.}}$). This observation suggests that while criteria benefit critiques, they might have side effects for revisions. We plan to investigate the underlying mechanisms of this phenomenon in future research.

SFT Models	CriticEval			
	$\boldsymbol{F}_{\text{obj.}}$	$\boldsymbol{F}_{\text{sub.}}$	$\boldsymbol{R}_{\text{obj.}}$	$\boldsymbol{R}_{\text{sub.}}$
Full	**58.15**	**5.71**	**19.33**	5.78
- w/o Task	55.01	5.12	18.72	5.73
- w/o Criteria	57.28	5.46	19.12	**6.17**
- w/o Ref.	57.72	5.21	16.42	5.74
- w/o All	57.11	5.12	13.86	5.73

Table 6: Ablation study on crucial information.

Ablation Study on MARV We evaluate MARV's contribution to the MultiCritique-RL pipeline by fine-tuning the SFT model using RL with a reward model trained without MARV. Table 7 illustrates that exclusion of MARV results in a notable decline in performance. For example, the model (w/o MARV) falls short of the SFT baseline on the subjective feedback evaluation ($4.84 < 5.71$). These results demonstrate that MARV is essential for stabilizing RL fine-tuning, as it excludes the noise preference samples.

Models	CriticEval			
	$\boldsymbol{F}_{\text{obj.}}$	$\boldsymbol{F}_{\text{sub.}}$	$\boldsymbol{R}_{\text{obj.}}$	$\boldsymbol{R}_{\text{sub.}}$
SFT Stage	58.15	5.71	**19.33**	5.78
RL Stage	**63.28**	**6.07**	19.26	**6.33**
- w/o MARV	63.05	4.84	18.79	5.99

Table 7: Ablation study on MARV.

MultiCritique Improves General Capability We investigate whether MultiCritique enhances LLM's general capabilities by integrating our proposed MultiCritiqueDataset-SFT into open-source instruction-tuning datasets. Our evaluation framework includes two dimensions: (1) Objective evaluation computes the average performance on 18 famous benchmarks, like MMLU (Hendrycks et al., 2021b) and GSM8K (Cobbe et al., 2021); (2) Subjective evaluation uses CompassJudger toolkit (Cao et al., 2024) to evaluate performance on four general benchmarks: AlignBench (Liu et al., 2024d), AlpacaEval, Alpaca Hard (Li et al., 2023) and MTBench-101 (Bai et al., 2024). Table 8 shows that our datasets significantly improves critique and general capabilities, *e.g.*, average 9.43% gain on AlpacaEval and Alpaca-Hard.[4]

Models	CriticBench	General Benchmarks				
	Overall	Avg. Obj.	Align-Bench	Alpaca Eval	Alpaca Hard	MTBench101
w/o Ours	38.80	55.23	5.07	23.51	22.78	7.75
w/ Ours	**71.60**	**55.43**	**5.10**	**27.43**	**23.28**	**7.81**

Table 8: Experiments on InternLM2-7B-Chat-SFT base model. **Avg. Obj.** indicates the average objective scores of LLMs over 18 benchmarks.

Generalization to Unseen Tasks We further investigate the generalization of MultiCritique by removing mathematical and coding samples from our training dataset.Table 9 reveals that the model fine-tuned on our SFT dataset without math and code critiques achieves substantial performance gains on these tasks, closely matching the results of models trained on the full dataset (88.44% $\approx$ 88.56%, 77.63% $\approx$ 78.37%).

Models	CriticBench					
	Math	Comm.	Symb.	Algo.	Code	Overall
Baseline	59.46	48.26	42.63	37.21	63.97	53.98
+ Ours	**88.56**	**62.13**	**57.02**	**57.35**	**78.37**	**73.72**
- w/o MC	88.44	60.42	55.02	45.91	77.63	71.68

Table 9: Generalization evaluation of critique ability to unseen **M**ath and **C**ode reasoning tasks, denoted as MC.

7 Conclusions and Future Works

In this paper, we propose a novel data generation pipeline, MultiCritique, to automatically construct the dataset to improve the critique ability of LLMs through SFT and RL fine-tuning stages. Extensive experiments demonstrate that MultiCritique significantly surpasses existing datasets. Additionally, the RL fine-tuning stage on MultiCritique further improves the critique abilities of LLMs. In the future, we plan to expand MultiCritique to the pairwise response comparison (Lan et al., 2024), enhancing LLMs' ability to evaluate paired responses. Moreover, we also plan to enhance the quality of MultiCritique further and tackle challenging reasoning tasks, like mathematical and coding reasoning.

[4] Please refer to the Table 11 in Appendix C.2 for the complete results in Objective Evaluation.

Limitations

Limitations in MultiCritique-SFT Our MultiCritique-SFT pipeline utilizes four LLMs selected for their strong critique capabilities as of April 2024 (Lan et al., 2024). While these models may not reflect the latest advancements, future iterations will integrate more advanced models (e.g., Llama-3.1, OpenAI o1 series). Expanding the number of models could enhance critique diversity, but computational constraints limited this study to four.

Limitations in MultiCritique-RL The MultiCritique-RL pipeline integrates Multi-Agent-Revision-Validating (MARV) to refine preference-critique pairs, using revision quality as a proxy for critique quality. Currently, we use InternLM2-20B-reward (Cai et al., 2024), a leading model on RewardBench (Lambert et al., 2024), for quality assessment. However, its performance may vary across tasks. While human annotation offers the most reliable evaluation, its high cost and limited scalability necessitated this practical approach. Future work will aim to enhance reward modeling accuracy to improve the MARV component.

Insufficient Investigation on Larger Models As detailed in Section 4.1, our work focuses on improving critique capabilities in efficient 7B-8B models. While extensive experiments were conducted on these models, our preliminary tests with larger models like Qwen2.5-72B-Instruct (Team, 2024) show limited gains. This is likely due to the strong and even better performance of these 70B models (e.g., Qwen2.5-72B-Instruct surpasses GPT-4 on CRITICBENCH and CRITICEVAL). In future work, we aim to explore the data-mixing strategies to unlock the full potential of our dataset for these models.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 7421–7454. Association for Computational Linguistics.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, and 1 others. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

Maosong Cao, Alexander Lam, Haodong Duan, Hongwei Liu, Songyang Zhang, and Kai Chen. 2024. Compassjudger-1: All-in-one judge model helps model evaluation and evolution. *Preprint*, arXiv:2410.16256.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023. Theoremqa: A theorem-driven question answering dataset. *Preprint*, arXiv:2305.12524.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

LMDeploy Contributors. 2023a. Lmdeploy: A toolkit for compressing, deploying, and serving llm. `https://github.com/InternLM/lmdeploy`.

XTuner Contributors. 2023b. Xtuner: A toolkit for efficiently fine-tuning llm. `https://github.com/InternLM/xtuner`.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *Preprint*, arXiv:2310.01377.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, and 1 others. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *Preprint*, arXiv:2305.14325.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Xinyu Hu, Li Lin, Mingqi Gao, Xunjian Yin, and Xiaojun Wan. 2024. Themis: Towards flexible and interpretable nlg evaluation. *arXiv preprint arXiv:2406.18365*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Tianyi Qiu, Juntao Dai, and Yaodong Yang. 2024. Aligner: Efficient alignment by learning to correct. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2023. Tigerscore: Towards building explainable metric for all text generation tasks. *ArXiv*, abs/2310.00752.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Pei Ke, Bosi Wen, Zhuoer Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, Jie Tang, and Minlie Huang. 2024. Critiquellm: Towards an informative critique generation model for evaluation of large language model generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. *Preprint*, arXiv:2402.06782.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. *Preprint*, arXiv:2310.08491.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations – democratizing large language model alignment. *Preprint*, arXiv:2304.07327.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

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. *Preprint*, arXiv:2403.13787.

Tian Lan, Wenwei Zhang, Chen Xu, Heyan Huang, Dahua Lin, Kai Chen, and Xian ling Mao. 2024. Criticbench: Evaluating large language models as critic. *Preprint*, arXiv:2402.13764.

Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, and Pilsung Kang. 2024. Checkeval: Robust evaluation framework using large language model via checklist. *Preprint*, arXiv:2403.18771.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024a. Cmmlu: Measuring massive multitask language understanding in chinese. *Preprint*, arXiv:2306.09212.

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

Ruosen Li, Teerth Patel, and Xinya Du. 2024c. Prd: Peer rank and discussion improve large language model based evaluations. *Preprint*, arXiv:2307.02762.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2024. Encouraging divergent thinking in large language models through multi-agent debate. *Preprint*, arXiv:2305.19118.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection. *Preprint*, arXiv:1708.02002.

Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. 2024. Criticbench: Benchmarking llms for critique-correct reasoning. *Preprint*, arXiv:2402.14809.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024a. Skywork-reward: Bag of tricks for reward modeling in llms. *Preprint*, arXiv:2410.18451.

Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Diyi Yang, and Soroush Vosoughi. 2024b. Training socially aligned language models on simulated social interactions. In *The Twelfth International Conference on Learning Representations*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024c. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, Xiaohan Zhang, Lichao Sun, Xiaotao Gu, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024d. Alignbench: Benchmarking chinese alignment of large language models. *Preprint*, arXiv:2311.18743.

Xinyi Liu, Pinxin Liu, and Hangfeng He. 2024e. An empirical analysis on large language models in debate evaluation. *Preprint*, arXiv:2406.00050.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024f. HD-eval: Aligning large language model evaluators through hierarchical criteria decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7641–7660, Bangkok, Thailand. Association for Computational Linguistics.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025. Inference-time scaling for generalist reward modeling. *Preprint*, arXiv:2504.02495.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Qingwei Lin, Jianguang Lou, Shifeng Chen, Yansong Tang, and Weizhu Chen. 2024. Arena learning: Build data flywheel for llms post-training via simulated chatbot arena. *Preprint*, arXiv:2407.10627.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. 2024. Llm critics help catch llm bugs. *Preprint*, arXiv:2407.00215.

Meta. 2024. Llama 3 model card.

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

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *Preprint*, arXiv:2206.05802.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. *Preprint*, arXiv:2009.01325.

Shichao Sun, Junlong Li, Weizhe Yuan, Ruifeng Yuan, Wenjie Li, and Pengfei Liu. 2024. The critique of critique. *Preprint*, arXiv:2401.04518.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

Zhengyang Tang, Ziniu Li, Zhenyang Xiao, Tian Ding, Ruoyu Sun, Benyou Wang, Dayiheng Liu, Fei Huang, Tianyu Liu, Bowen Yu, and Junyang Lin. 2025a. Enabling scalable oversight via self-evolving critic. *Preprint*, arXiv:2501.05727.

Zhengyang Tang, Ziniu Li, Zhenyang Xiao, Tian Ding, Ruoyu Sun, Benyou Wang, Dayiheng Liu, Fei Huang, Tianyu Liu, Bowen Yu, and Junyang Lin. 2025b. Realcritic: Towards effectiveness-driven evaluation of language model critiques. *Preprint*, arXiv:2501.14492.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Jason Toy, Josh MacAdam, and Phil Tabor. 2024. Metacognition is all you need? using introspection in generative agents to improve goal-directed behavior. *arXiv preprint arXiv:2401.10910*.

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. *Preprint*, arXiv:2404.18796.

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. *Preprint*, arXiv:2407.10817.

Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2024a. Direct judgement preference optimization. *Preprint*, arXiv:2409.14664.

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. *Preprint*, arXiv:2408.02666.

Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. Shepherd: A critic for language model generation. *Preprint*, arXiv:2308.04592.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2024c. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization.

Yuqing Wang and Yun Zhao. 2024. Metacognitive prompting improves understanding in large language models. *Preprint*, arXiv:2308.05342.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *Preprint*, arXiv:2407.19594.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994, Singapore. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, and 1 others. 2024a. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. 2024b. Rlcd: Reinforcement learning from contrastive distillation for language model alignment. *Preprint*, arXiv:2307.12950.

Zihuiwen Ye, Fraser Greenlee-Scott, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gallé. 2024. Improving reward models with synthetic critiques. *Preprint*, arXiv:2405.20850.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024a. Metamath: Bootstrap your own mathematical questions for large language models. *Preprint*, arXiv:2309.12284.

Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, Melanie Kambadur, Dhruv Mahajan, and Rui Hou. 2024b. Self-generated critiques boost reward modeling for language models. *Preprint*, arXiv:2411.16646.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *Preprint*, arXiv:2401.10020.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *Preprint*, arXiv:1905.07830.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Bo Wang, Shimin Li, Yunhua Zhou, Qipeng Guo, Xuanjing Huang, and Xipeng Qiu. 2024. Scaling of search and learning: A roadmap to reproduce o1 from reinforcement learning perspective. *Preprint*, arXiv:2412.14135.

Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. 2024a. Exploring collaboration mechanisms for llm agents: A social psychology view. *Preprint*, arXiv:2310.02124.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024b. Generative verifiers: Reward modeling as next-token prediction. *Preprint*, arXiv:2408.15240.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2024c. Evaluating the performance of large language models on gaokao benchmark. *Preprint*, arXiv:2305.12474.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. 2024. OpenCodeInterpreter: Integrating code generation with execution and refinement. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 12834–12859, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

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

A Differences between Recent Works

This section will discuss the primary differences between our designed data generation pipeline and existing works, like Prometheus (Kim et al., 2024).

Difference in Data Preparation Although Prometheus collects five responses with quality scores ranging from 1 to 5 (Kim et al., 2024), these responses are synthesized using a GPT-4 reference response, leading to responses that are very similar to the reference, which is a significant deviation from real-world scenarios.

Difference in Crucial Information Although Prometheus also employs customized criteria for better critiques (Kim et al., 2024), our work differs significantly. Our evaluation criteria are organized into a hierarchical two-tier structure, providing clear definitions for diverse evaluation dimensions—a method proven effective in automatic evaluation (Lee et al., 2024; Liu et al., 2024f). In contrast, Prometheus synthesizes one criterion using GPT-4, lacking sufficient guidelines for high-quality reference response and critique generation.

B Complete Scaling Experimental Results

The complete scaling experimental results on CRITICEVAL and CRITICBENCH benchmarks are shown in Figure 3.

C Implementation Details

C.1 MultiCritiqueDataset Construction

Query Preparation All the queries in Auto-J (Li et al., 2024b) and DEITA (Liu et al., 2024c) are collected. For OpenHermes-2.5[5], we sample 1K queries for its 28 categories, leading to 28K queries. Following previous work (Yuan et al., 2024), we use 3.2K examples from the OpenAssistant dataset by sampling only the first conversation turns in the English language that achieves the highest human-annotated scores. Besides, we also sample 2K mathematical and coding questions from

[5] https://huggingface.co/datasets/teknium/OpenHermes-2.5

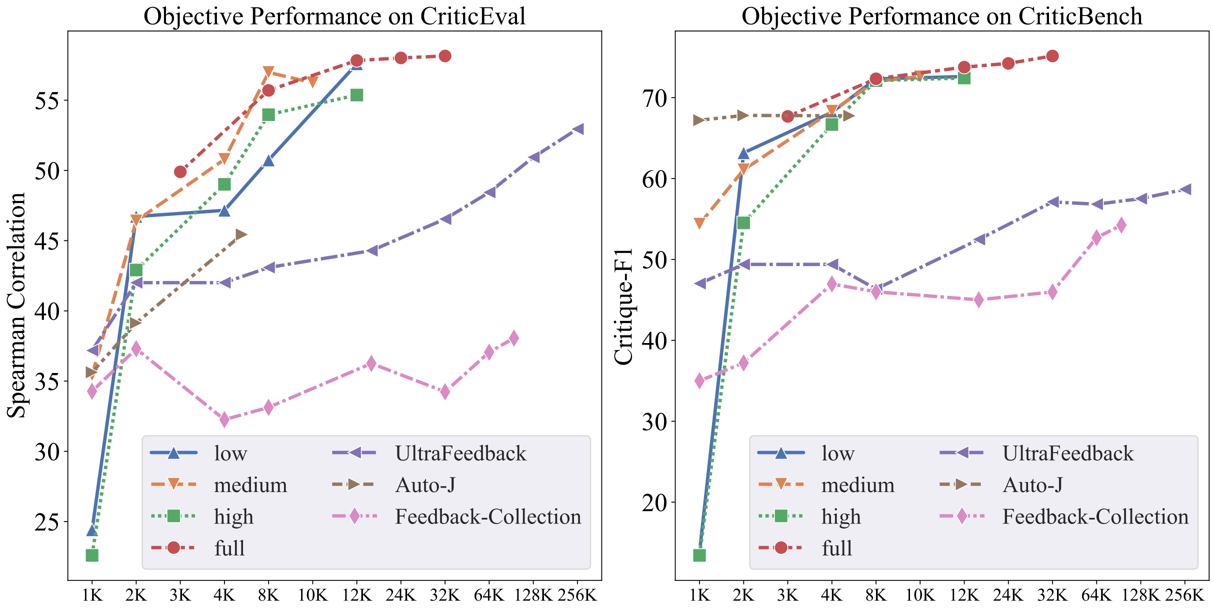

Figure 3: The correlation between the number of training samples in the SFT dataset (from 1K to 256K) and critique ability. *low, medium, high* and *full* represent the models that are trained on critiques in MultiCritiqueDataset-SFT for low-, medium-, high-quality, and all three response qualities (full), respectively. Please refer to Appendix B for complete results on CRITICBENCH and CRITICEVAL benchmarks

MetaMathQA (Yu et al., 2024a) and CodeFeedback (Zheng et al., 2024) datasets to collect critiques for reasoning tasks. Only the first conversation utterance (the coding question) in CodeFeedback is used. **None of the training samples are from the test set in CRITICEVAL and CRITICBENCH benchmarks.**

Response Quality	Average
Low	-1.41
Medium	0.70
High	1.69

Table 10: The average reward model scores for each response quality.

Collect Evaluated Responses To collect diverse evaluated responses for queries, we use eleven widely-used LLMs with varying scales and capabilities in this work: (1) Qwen-1.5-72B-Chat; (2) Qwen-1.5-7B-Chat; (3) InternLM2-20B-Chat; (4) Yi-34B; (5) Mixtral-8x7B-Instruct; (6) Llama2-13B-Chat; (7) Llama2-7B-Chat; (8) Gemma-2B; (9) Baichuan2-13B-Chat; (10) Vicuna; (11) WizardLM-7B-v0.1. The LMDeploy tookit (Contributors, 2023a) is used to inference these LLMs by random sampling decoding method, and the hyper-parameters are 0.95 top-p and 0.8 temperature. Besides, the InternLM2-20B-reward (Cai et al., 2024) model[6] is used to score the quality of responses, and the reward scores are used to classify responses into three quality levels. Our preliminary experiments reveal that this reward model exhibits a strong correlation with human judgments in distinguishing response quality. Therefore, we use the reward model to automatically complete this process. The average reward scores for each response quality are shown in Table 10. It can be observed that there exists a significant performance gap among these response qualities. We would like to clarify several important points about using reward models in this phase: (1) Model Performance: At the time we conducted this research, InternLM2-20B-reward was the top-performing model on RewardBench (Lambert et al., 2024). Our preliminary study demonstrate that this model could effectively assess the quality differences; (2) Coarse-grained classification purpose: We want to emphasize that our primary goal in using the reward model is to perform a coarse-grained classification of responses into different quality tiers, ensuring a balanced distribution in our dataset. It is not the core contribution of our work but rather a pre-processing step to balance the data distribution.

[6] InternLM2-20B-Reward was the top-tier reward model in RewardBench (Lambert et al., 2024) when we start our project.

Given that reward models fail to accurately evaluate the quality of responses in mathematical and coding questions, we only collect two kinds of response qualities: (1) high-quality responses generated by GPT-4o and (2) low-quality responses generated by eight 7B-20B open-source LLMs.

Collect Crucial Information The prompt for LLMs to generate task description, two-tier structured criteria and reference response are described in Appendix J. Our preliminary study reveals that reference responses tend to produce critiques that lack diversity for mathematical and coding questions. As a result, we set reference responses as empty for these two tasks.

Most previous works rely on human-annotated criteria for each task (Hu et al., 2024; Li et al., 2024b), which do not scale well. We propose generating a customized two-tier structure evaluation criteria tailored to each query using GPT-4. Besides, the user pre-defined criteria are provided as input optionally for better flexibility.

Pre-process Evaluated Responses Our proposed ACUs contain the location of flaws in the evaluated response for better interpretability. To achieve this goal, we pre-process the evaluated responses by appending labels for sentences in evaluated responses. For most tasks, punctuation marks such as periods, exclamation marks, and semicolons are used to divide sentences. For code-related task scenarios, the sentence is divided by the line breaks to represent lines of the evaluated code.

Collect Preference Dataset The threshold of differences in accumulated severity scores is set as 5 in this paper. Besides, we leverage four additional 7B LLMs to revise the evaluated response eight times, given the model-generated critiques: (1) InternLM2.5-7B-Chat; (2) Llama-3.1-8B-Instruct; (3) Qwen2-7B-Chat; (4) Mistral-7B-Instruct. The random sampling decoding method is used to generate diverse revisions, and the hyper-parameters are (1) 0.95 top-p, (2) 50 top-k, and (3) 1.0 temperature. The InternLM-20B-reward model (Cai et al., 2024) is used to evaluate the response quality, which was the top-tier reward model in RewardBench (Lambert et al., 2024). For mathematical problems, we compute the exact answer matching rather than reward model scores. The vLLM toolkit (Kwon et al., 2023) is used to speed up the inference.

C.2 Experimental Details

Evaluation Noted that Prometheus (Kim et al., 2024) requires criteria and reference responses as inputs, which are unavailable in the two benchmarks. To address this, we fine-tune LLMs using our processed dataset, moving the evaluation criteria and reference responses into the output. Some experimental results are derived from existing work. All evaluation experimental results reported in this paper are averaged from 3 runs.

SFT During the SFT training stage, the InternLM2-7B-Chat model is fine-tuned by optimizing the Maximum Likelihood Estimation (MLE) loss:

$$L_{\text{MLE}} = -\frac{1}{\mathbb{N}}\sum_{i=1}^{\mathbb{N}} \log p_\theta(\mathcal{CI}_i, \mathcal{C}_i | q_i, r_i) \quad (1)$$

The training process is running on 2 A800 GPU Serves (16 GPUs) by using DeepSpeed[7]. To achieve a fair comparison, we set the training hyper-parameters as follows: (1) 4e-5 learning rate; (2) 6e-6 minimum learning rate; (3) 32,768 maximum sequence length; (4) 2 epoch; (5) 1 batch size; (6) AdamW optimizer.

We explore the effect of instruction format and data recipe of crucial information during the SFT stage in Appendix I. We fix the following experimental setup for supervised fine-tuning: (1) the proportion of the single-turn template is 5% and left 95% training samples for SFT are multi-turn conversations, consisting task description, two-tier structured evaluation criteria, reference responses, critiques consisting of a list of ACUs generated by MultiCritique-SFT pipeline and summarization of the final judgment for the evaluated response; (2) the crucial information for each query is only optimized once in 2 epochs.

In Section 6, we analyze the contributions of our proposed MultiCritique-SFT pipeline. We only collect the summarization of final judgments for the critiques generated by MultiCritique-SFT , and the critiques generated by each LLM do not have the corresponding summarizations. Thus, in this experiment, we do not fine-tune the model to predict the summarization of final judgments in 95% multi-turn training samples.

Ablation Study in SFT We conduct the ablation study in Section 6 (**Ablation Study on**

[7] https://github.com/microsoft/DeepSpeed

MultiCritique-SFT) to prove the effectiveness of aggregated critiques generated by our proposed MultiCritique pipeline. The dataset in this ablation study is slightly different from that in the main experiment, consisting of two parts:

- **Distinct parts**: crucial information and analytical critiques (a list of ACUs), without the summarization and judgment score.
  - **MultiCritique-SFT Critiques:** Critiques are generated through our MultiCritique-SFT pipeline, which aggregates accurate ACUs from multiple models via meta-critique classification.
  - **Four Individual LLMs Critiques:** Critiques are extracted from our raw MultiCritique-SFT dataset, using feedback generated independently by four models.
- **Shared parts**: to enable the objective evaluation on CriticEval and CriticBench benchmarks, we supplement 5% of samples from MultiCritique-SFT to ensure the fair comparison, which consists of crucial information, analytical critiques, summarization, and judgment score.

Reinforcement Learning During the reinforcement learning stage, we first train the InternLM2-7B-Chat as the reward model on MultiCritiqueDataset-RL by using xtuner toolkit (Contributors, 2023b), and the hyper-parameters are as follow: (1) 32,768 maximum sequence length; (2) 1 epoch; (3) 1 batch-size; (4) AdamW optimizer; (5) 2e-5 learning rate; (6) focal loss (Lin et al., 2018). For the i-th sample, the focal ranking loss is computed to optimize the reward model:

$$L_{\text{ranking}} = -(1-2\times\max(0, P^i_{j_+,j_-} - \frac{1}{2}))^2 \log(P^i_{j_+,j_-}), \quad (2)$$

where $P^i_{j_+,j_-} = \sigma(r^i_{j_+} - r^i_{j_-})$ represents the probability that the reward score of C_{i,j_+} is greater than that of C_{i,j_-}. The difficulty decay coefficient only takes effect when the model correctly predicts the preference of i-th training sample, *i.e.*, $P^i_{j_+,j_-} > 0.5$, otherwise it equals to 1.

Subsequently, we conduct the PPO algorithm to optimize the SFT model on six nodes of A800 GPU servers (48 GPU cards) with the ray toolkit.[8] The hyper-parameters during reinforcement learning are listed as below: (1) 30,000 maximum sequence length; (2) 64 batch-size; (3) deepspeed zero-2; (4) 0.9 top-p and 1.0 temperature sampling parameters for policy model.

Evaluation We leverage the publicly available codebase of CRITICEVAL and CRITICBENCH for evaluation. To ensure the robust objective evaluation of the revision critique dimension, we leverage five LLMs with varying capabilities to revise the responses given feedback generated by each baseline: InternLM2-7B-Chat, InternLM2.5-7B-Chat, InternLM2-20B-Chat (Cai et al., 2024), Mixtral-7x8B-Instruct (Jiang et al., 2024) and Llama-3.1-70B-Instruct. Due to the limited OpenAI API budget, we only conduct the subjective evaluation on the revision dimension to evaluate the quality of revisions generated by the Llama-3.1-70B-Instruct model.

In CRITICBENCH benchmark, the responses with ≥ 7 Likert Scores generated by our fine-tuned models are treated as the positive samples since responses with ≥ 7 are comparable or better than the reference answers in our defined score rubrics, which is described in Appendix J.6. The responses with > 2 quality scores are treated as positive samples for the Prometheus model since the overall score range is 1 to 5.

Benchmark	w/o Our SFT	w/ Our SFT
MMLU (Hendrycks et al., 2021a)	62.28	**62.57**
CMMLU (Li et al., 2024a)	61.13	**61.22**
C-Eval (Huang et al., 2023)	57.91	**58.34**
GaoKaoBench (Zhang et al., 2024c)	**55.89**	55.13
TriviaQa (Joshi et al., 2017)	67.56	**67.66**
NQ (Kwiatkowski et al., 2019)	**27.04**	26.23
RACE (Lai et al., 2017)	**88.16**	88.08
Winogrande (Sakaguchi et al., 2019)	**74.59**	73.32
HellaSwag (Zellers et al., 2019)	**93.38**	93.36
BBH (Suzgun et al., 2022)	**60.92**	60.3
GSM8K (Cobbe et al., 2021)	**75.44**	74.3
MATH (Amini et al., 2019)	41.92	**42.72**
TheoremQA (Chen et al., 2023)	15.75	**16.88**
HumanEval (Chen et al., 2021)	56.1	**57.93**
MBPP (Austin et al., 2021)	55.25	**57.59**
CodeBench (LCBench)	**16.07**	12.95
GPQA (Rein et al., 2023)	26.77	**29.8**
IFEval (Zhou et al., 2023)	58.04	**59.33**
Average	55.23	**55.43**

Table 11: Complete results on objective benchmarks.

Regarding the general capability evaluation in Section 6, we evaluate 18 objective evaluation benchmarks. The complete results are shown in Table 11. Experimental results that LLM's perfor-

[8] https://github.com/ray-project/ray

mance on these objective benchmarks is slightly improved with our proposed MultiCritiqueDataset-SFT.

To ensure reproducibility, the greedy search decoding strategy is used for inference. As for the models we fine-tuned on our proposed MultiCritiqueDataset, the optional user pre-defined criteria are empty during inference.

D Statistics of MultiCritiqueDataset

The statistical information of our proposed MultiCritiqueDataset is shown in Table 12. Our proposed MultiCritiqueDataset significantly outperforms existing critique datasets from multiple dimensions, like response quality and the number of tasks. Although the size of UltraFeedback and Feedback-Collection are greater than our proposed MultiCritiqueDataset, the models fine-tuned on them are much worse than that fine-tuned on MultiCritiqueDataset, demonstrating the better quality of our proposed dataset. Although Feedback-Collection and Preference-Collection consist of 5 response qualities, they are synthesized by GPT-4, resulting in very similar content with reference responses.

The complete list of the task scenarios in our proposed MultiCritiqueDataset is shown in Table 13, consisting of 123 tasks. Except for 58 fine-grained tasks defined in Auto-J (Li et al., 2024b), our proposed dataset includes 65 categories defined in the OpenHermes-2.5 dataset.

The overall quota for using OpenAI and Claude API to construct our proposed MultiCritiqueDataset are 9,180$ and 125.6$, respectively. The average API cost for each sample is 0.29$. Given that the average price of one human-annotated critique is 8$ (Wang et al., 2023), our data generation pipeline is much cheaper and easier to scale to more diverse task scenarios.

E Excluded Baselines and Benchmarks

E.1 Excluded Baselines

Some baselines are excluded during our evaluation, and the reasons are described as follows: (1) Prometheus2 (Kim et al., 2024) extends the Prometheus to pairwise-evaluation. It is unsuitable in our evaluation; (2) InstructScore (Xu et al., 2023) is trained on samples with limited tasks, failing to extend to other diverse tasks, like mathematics reasoning and code generations; (3) JudgeLM (Zhu et al., 2023) is mainly trained to compare two responses with critiques. Although it can be used to score the single responses, the reference responses should be supplied[9], which are unavailable in CRITICEVAL and CRITICBENCH; (4) Reward models (Lambert et al., 2024) are also widely used to evaluate the quality of responses. However, their scores can only reflect the relative differences in response quality, so reward models cannot be assessed in CRITICBENCH. Additionally, due to the lack of textual critiques, reward models are unsuitable for evaluation under CRITICEVAL. Recently, OpenAI o1 model has demonstrated powerful critique ability. Due to the huge cost on evaluating o1 model on large-scale CRITICEVAL and CRITICBENCH benchmarks, we do not include o1 models in this paper.

Besides, due to the un disclosed model parameters, some recent works cannot be evaluated in our work, like Critic-RM (Yu et al., 2024b) and SFR-Judge (Wang et al., 2024a).

E.2 Excluded Benchmarks

Existing benchmarks for evaluating the critique ability of LLMs could be classified into two categories: (1) single-response evaluation; (2) pairwise response comparison (Li et al., 2024b; Kim et al., 2024). Single-response evaluation aims to evaluate the quality of a single response given the context of the conversation or user query. For example, CRITICEVAL and CRITICBENCH evaluate whether LLMs could accurately score the quality of responses. Pairwise response comparison selects the better response from a pair of responses. For example, RewardBench (Lambert et al., 2024), Feedback-Bench (Kim et al., 2024) and PandaLM test set (Wang et al., 2024c) consist of numerous pairs of responses with clear performance gap. Since pairwise response comparison is much simpler than comparing the scores corresponding to the two responses, it is unfair for our models under the pairwise response comparison benchmarks. Therefore, these benchmarks are not used in our paper. We will extend our proposed MultiCritiqueDataset from single-response evaluation to the pairwise response comparison (Li et al., 2024b).

Recently, RealCritique (Tang et al., 2025b) propose a closed-loop methodology that evaluates the quality of corrections generated from critiques.

[9] https://github.com/baaivision/JudgeLM/tree/main/judgelm/llm_judge

Dataset	Type	Task Desc.	Criteria	Ref.	Tokens	Resp. Quality	Num. Task	Num. Query	Num. Resp.	Avg. Turn	Public
Auto-J	SFT	✗	✗	✗	3.8M	-	58	4.4K	4.4K	1	✓
UltraFeedback	SFT	✗	✗	✗	227M	-	9	257K	257K	1	✓
TIGERScore	SFT	✗	✗	✗	23.7M	-	-	42.5K	42.5K	1	✓
Feedback-Collection	SFT	✗	✓	✓	191.5M	5	-	20K	100K	1	✓
Preference-Collection	SFT	✗	✓	✓	382.9M	5	-	40K	200K	1	✓
Themis	SFT,RL	✗	✓	✗	-	-	9	67K	67K	1	✗
JudgeLM	SFT	✗	✗	✓	-	-	-	100K	200K	1	✓
MultiCritiqueDataset-SFT	SFT	✓	✓	✓	531.1M	3	123	10.7K	32.1K	2.40	✓
MultiCritiqueDataset-RL	RL	✓	✓	✓	352.9M	3	123	19.7K	39.4K	2.35	✓

Table 12: The comparison between our proposed MultiCritiqueDataset and existing critique datasets. **Avg. Turn** represents the average number of utterances in the multi-turn conversation history, and the user query is the last utterance in it. The number of tokens is counted based on the InternLM2-7B-Chat tokenizer.

However, this benchmark mainly focuses on reasoning tasks, like mathematics, neglecting the evaluation on diverse open-domain tasks. Therefore, we mainly leverage CRITICEVAL (Lan et al., 2024) to evaluate our models and baselines.

F Preliminary Study on Crucial Information

During designing our data generation pipeline, we conducted a preliminary study to verify whether crucial information helps reduce the complexity of critique tasks and improve the quality of collected critiques. Specifically, we conducted the self-critique prompting (Pan et al., 2024) to collect critiques and corresponding revisions and evaluated the quality of the critiques by measuring the quality of their corresponding revisions.

F.1 Experimental Setup

We first random sample 1,280 queries and evaluated responses from MultiCritiqueDataset. Then, four LLMs are prompted to generate critiques and subsequent revisions with or without each crucial information: (1) InternLM2.5-7B-Chat; (2) Qwen2-7B-Chat; (3) Llama-3.1-8B-Instruct; and (4) Mixtral-7B-Instruct. Each model generates critiques and revisions eight times, leading to overall $4 \times 8 = 32$ revisions, and the advanced InternLM2-20B-reward model judges the quality of these revisions.

F.2 Experimental Results

First of all, as shown in Table 14, it can be found that the quality of reference responses generated given the customized evaluation criteria is much better, indicating the effectiveness of our proposed two-tier structure evaluation criteria.

	Reward
Ref. w/ Criteria	0.54
Ref. w/o Criteria	0.45

Table 14: Avg. rewards.

Besides, as shown in Table 15, it can be found that the quality of revisions becomes worse when task descriptions and reference answers are removed. Besides, removing all the crucial information leads to the worst performance ($-0.005 < 0.085$). Note that we do not evaluate the contributions of customized evaluation criteria since its contribution is proven in Table 14, *i.e.,* improves the quality of reference responses.

	Reward
Origin Response	-0.11
w/ All	**0.085**
w/o Task	0.076
w/o Ref.	0.029
w/o All	-0.005

Table 15: Avg. reward scores.

G Can Other LLMs Conduct Meta-Critique?

Model	Corr.	Agree.
Claude-3.5-Sonnet	0.58	0.71
Qwen2.5-72B-Instruct	0.60	0.72

Table 16: Correlation and agreement between advanced LLMs and GPT-4.

Our MultiCritique framework is a general and model-agnostic framework. GPT-4 serves as one possible meta-critique judge model. While we chose GPT-4 due to its advanced meta-critique capabilities (Sun et al., 2024; Lan et al., 2024), any sufficiently advanced LLM can fulfill this role. To demonstrate this flexibility, we conducted additional experiments with Claude-3.5-Sonnet and Qwen2.5-72B-Instruct. Specifically, we randomly sample 200 samples from MultiCritiqueDataset-SFT, and prompt Qwen2.5-72B-Instruct and Claude-3.5-Sonnet to re-run the Meta-Critique Classification and Critique Summarization processes. Subsequently, we assess two metrics to reflect the correlations between GPT-4 and these models: (1) Spearman correlation scores between GPT-4 and these models on assessing the response quality; (2) Agreement on judging the correctness of each ACU in Meta-Critique classification. As shown in Table 16, these models achieve not only high meta-critique agreement ($>$ 70%, the random baseline is 50%) with GPT-4 but also strong correlation (Spearman $\rho \approx 0.60$) in final judgment scores. These results indicate that the effectiveness of our MultiCritique does not critically depend on GPT-4. Other advanced models could also be used in MutiCritique to conduct the Meta-Critique classification.

H Experiments on More LLMs

Except for InternLM2-7B-Chat-SFT, we also conduct experiments on three more advanced LLMs: (1) Llama-3-8B-Instruct (Meta, 2024); (2) Qwen2.5-7B-Instruct (Team, 2024); and (3) InternLM2.5-7B-Chat (Cai et al., 2024). Table 17 demonstrates that MultiCritique-SFT also effectively improves the critique ability of these advanced 7B LLMs. Notably, the fine-tuned Qwen2.5-7B-Instruct model outperforms GPT-4 on CRITICBENCH (79.04% $>$ 78.75%).

I How Factors Affect Performance During SFT Stage

In this section, we analyze two factors that influence the performance of models fine-tuned on our proposed MultiCritiqueDataset-SFT: (1) instruction format; and (2) the data recipe of crucial information.

I.1 Instruction Format

Our proposed MultiCritiqueDataset-SFT consists of mult-turn conversations for generating critiques. To ensure the generalization of fine-tuned models, in this paper, we construct the single-turn and multi-turn prompt templates in the instruction dataset for the supervised fine-tuning (SFT) stage. Our experimental results reveal that the proportion of single-turn and multi-turn templates in the training data significantly affects the model's performance. As shown in Table 18, it can be observed that when the proportion of single-turn templates is 5% of the total data size, the fine-tuned models could achieve optimal performance on the feedback objective evaluation of CRITICEVAL during the SFT stage. Therefore, this setting is used in all the experiments in our paper.

Rate	$F_{\text{obj.}}$
1.0%	61.14
2.5%	57.32
5.0%	**63.85**
10.0%	60.21

Table 18: The proportion of single turn prompts.

I.2 Data Recipe of Crucial Information During SFT stage

As described in Appendix D, our proposed MultiCritiqueDataset-SFT consists of 32.1K evaluated responses and 10.7K queries, and the same query has the same crucial information: task description, criteria, and reference response. Therefore, the training volume on crucial information will be three times larger than that of critiques. This might lead to overfitting crucial information, influencing the optimization of critiques. To address this problem, we mask the loss of crucial information at varying rates. As shown in Table 19, it can be found that the performance of fine-tuned model on CRITICEVAL benchmark decreases when the proportion of training volume on crucial information increases, and the best proportion of training volume is 16.67%, *i.e.,* **the crucial information for each query is only optimized once in 2 epochs**. We leverage this experimental setting in all our experiments.

Rate	$\boldsymbol{F}_{\text{obj.}}$
16.67%	63.85
33.33%	60.19
66.67%	56.99
100.0%	57.47

Table 19: The proportion of training volume on crucial information.

J Designed Prompts in MultiCritique

In this section, we provide the detailed prompts that used in our proposed MultiCritiqueDataset data generation pipeline.

J.1 Task Description

The prompt for GPT-4 model to generate the task description is shown in Figure 4, while the multi-turn conversations are not provided.

J.2 Criteria Generation

The prompt for GPT-4 to generate the two-tier structured criteria is shown in Figure 8. Note that the user could provide their pre-defined criteria. If the criteria provided by users are not empty, GPT-4 is asked to generate the two-tier structured criteria from scratch; otherwise, GPT-4 is asked to expand on the criteria provided by the user and must not generate content that conflicts with the user's provided criteria. Besides, it can be found that each item of criteria consists of 3 fundamental values: (1) criteria name, (2) criteria fine-grained description, and (3) importance degree of the criteria (normal, medium, important).

J.3 Reference Generation

Given the two-tier structured criteria, GPT-4 is asked to generate high-quality reference responses that satisfy all the evaluation criteria, as shown in Figure 5.

J.4 Multi-Agent Analytical Critique

After generating the three crucial information, multiple LLMs are asked to follow the instructions in Figure 9 to critique the evaluated responses. It can be found that LLMs are asked to critique the evaluated responses sentence by sentence and generate a list of Analytical Critique Units (ACUs) consisting of 5 key values: (1) citation symbol of the sentence in evaluated response; (2) description of this flaw; (3) which criteria this flaw belongs to; (4) severity of this flaw; (5) revision suggestions.

J.5 Meta-Critique Classification

As shown in Figure 6, after collecting multiple critiques generated by LLMs, the GPT-4 model is asked to conduct the meta-critique to analyze the quality of each ACU. Each ACU is classified into seven categories.

The detailed descriptions of each meta-critique and corresponding severity score are shown in Table 20.

J.6 Critique Summarization

Finally, the GPT-4 model is asked to summarize the critiques from multiple LLMs and generate the final critiques and summarization for the evaluated responses. As shown in Figure 7, it can be found that the judgment scores for evaluated responses are the floating numbers ranging from 1 to 10, and the ≥ 7 scores indicate the comparable and even better qualities of responses than reference responses.

Now, you are a helpful assistant aiming to provide valuable critiques and analysis for the previous conversation history, thereby assisting in the analysis of the quality of subsequent responses in relation to this conversation history history.

# Your Tasks
Analyze and describe the primary purpose of user's query in conversation history. Do NOT generate very lengthy description, keep it concise and precise. **If the conversation history contains multiple turns between assistant and human, MUST analyze the main purpose of the user's last query by considering the previous conversation history.**

# Output Template
Generate the task description in following Markdown template. Do NOT add comment (//) in the template.
—
// a string for task description
# Task Description
A string analyze the attribute of the task
—

Figure 4: The prompt for generating task description about the last user query in conversation.

# Task Goal
Good! Your task is to generate a high-quality response for the **conversation history (before we provided the criteria list)**, which perfectly satisfies all the generated **first-tier and second-tier** criteria in last turn.

# NOTICE!!!
1. The conversation history here represents the conversations before we provided the criteria list. Do NOT respond to the last utterance.
2. Do NOT generate any explanation or analysis about your generated response.

Figure 5: The prompt for generating reference response given the criteria.

K Case Study in MultiCritiqueDataset

K.1 Case Study of Customized Evaluation Criteria

We provide one case of two-tier structured evaluation criteria for one query in MultiCritiqueDataset in Figure 10. Compared with existing works, like Themis and Auto-J, our evaluation criteria contain a more diverse and customized evaluation dimension for the user query, which is beneficial for robust and accurate evaluation.

K.2 Case Study of Critiques

We provide one case of analytical critique units (ACU), summarization, and judgment of critiques in Figure 11. Each ACU points out one flaw in a located sentence in the evaluated responses.

Good! Now, I want you to carefully re-check (meta-evaluation) each feedback entry generated by these models.

## Categories of Errors in Feedback Entries
Please carefully analyze each feedback entry in this list sequentially and categorize them into the following error types based on their errors:
E0. the feedback entry is helpful, perfect, and satisfying and accurately points out the flaw in the response, providing helpful suggestions for improvement.
E1. the cited sentence in the feedback entry is good without any flaws belonging to the mentioned criteria, and it should not be critiqued for the mentioned criteria.
E2. the cited sentence in the feedback entry has flaws belonging to the mentioned criteria, but the type of criteria is misclassified or does not exist in the previous criteria list.
E3. the severity of this flaw is misclassified.
E4. the description of this flaw is unreasonable and inaccurate.
E5. the suggestions for revising this flaw are unreasonable or introduce new problems.
E6. although revision suggestions for the flaw are reasonable without any problems, revision with suggestions will not necessarily improve the quality of the response.

## NOTICE!!!
1. Ensure the number of the generated analysis entries equals the number of feedback entries generated by the corresponding model. **Do NOT miss any feedback entries for analysis.**
2. If one feedback entry is similar to or the same as some analyzed feedback entries, **Do NOT regard it as a redundant feedback entry (redundant error). Please evaluate this feedback entry by focusing on analyzing errors (E0 to E6) in the feedback entry content.**

Please analyze each feedback entry one by one and sequentially, which will be used to summarize the final feedback generation.

Figure 6: The prompt for generating meta-critiques for all the critiques generated by multiple LLMs.

Task	Num.	Task	Num.	Task	Num.
default	9362	math reasoning	6228	code generation	5280
explaining general	3452	open question	3048	seeking advice	2674
value judgement	2586	roleplay	1210	functional writing	958
verifying fact	838	brainstorming	828	analyzing general	780
code correction rewriting	720	chemistry	718	physical	710
chitchat	702	bio	702	asking how to question	690
creative writing	632	rejecting	540	planning	538
counterfactual	528	awareness	480	editor	480
misconception	480	general	480	cot	480
experience	480	song	480	plan	480
joke	480	rp	480	multiple choice	480
trivia	480	counterfactual contextual	478	stylized response	478
theory of mind	478	writing	478	greeting	478
orca	478	riddle	478	wordgame	478
gtkm	468	recommendation	462	solving exam question without math	456
coding	452	writing personal essay	432	text summarization	430
summarization	424	explaining code	408	agent	406
text to text translation	396	writing email	372	question generation	372
card	372	instructional rewriting	360	ranking	358
writing song lyrics	318	writing cooking recipe	314	information extraction	300
post summarization	300	data analysis	294	writing job application	294
writing presentation script	292	classification identification	276	solving exam question with math	276
paraphrasing	240	detailed writing	222	writing advertisement	142
writing social media post	138	title generation	132	text correction	120
language polishing	114	writing product description	108	writing blog post	96
code to code translation	92	writing legal document	90	writing technical document	74
reading comprehension	66	text simplification	60	writing scientific paper	48
keywords extraction	40	writing marketing materials	36	topic modeling	18
writing news article	18	quiz	18	writing chapter	16
code simplification	12	note summarization	12	writing letter	12
writing history essay	6	predicting general	6	writing feature story	6
criticism	6	challenges	6	writing social responsibility report	6
impact	6	impact analysis	6	changing mindset	6
overview	6	writing consumer complaint	6	writing dialogue	6
writing sequel	6	writing historical document	6	exit planning	6
writing screenplay	6	writing deployment script	6	data conversion	6
time zone conversion	6	language history	6	writing press release	6
writing survival manual	6	writing movie review	6	writing biography	6
reward	6	writing comedy skit	6	writing note	6
writing love note	6	writing love letter	6	writing config file	6
writing script	6	writing kubernetes deployment file	6	writing code	2

Table 13: The complete list of task scenarios in our proposed MultiCritiqueDataset-SFT. The number of samples is also listed.

Models	CRITICEVAL	CRITICBENCH					
	$F_{\text{obj.}}$	Math	Comm.	Symb.	Algo.	Code	Overall
Qwen2.5-7B-Instruct	62.50	87.54	55.60	65.41	50.96	**84.05**	74.22
+ MultiCritiqueDataset-SFT	**64.82**	**93.45**	**60.59**	**66.67**	**62.96**	82.06	**79.04**
Llama3-8B-Instruct	37.20	78.33	**62.64**	**62.05**	**62.19**	**76.41**	70.71
+ MultiCritiqueDataset-SFT	**51.87**	**87.00**	59.92	61.41	60.22	74.61	**73.92**
InternLM2.5-7B-Chat	44.84	59.46	**63.97**	48.26	42.63	37.21	53.98
+ MultiCritiqueDataset-SFT	**58.29**	**88.56**	62.13	**57.02**	**57.35**	**78.37**	**73.72**

Table 17: Evaluation of critique ability of advanced Qwen2.5-7B-Instruct, Llama-3-8B-Instruct and InternLM2.5-7B-Chat models.

Label	Meaning	Detailed Description of Quality Category	Severity (1-5)
L0	Correct ACU	This feedback is accurate and provide helpful suggestions.	0
L1	False Negative ACU	The content is free from any flaws and should not be critiqued.	5
L2	Wrong Criteria	The type of criteria of feedback is misclassified or does not exist.	2
L3	Wrong Severity	The severity of this flaw is misclassified.	1
L4	Wrong Description	The descriptions of flaws are unreasonable or inaccurate.	4
L5	Wrong Suggestion	The suggestions for revisions are unreasonable or introduce errors.	4
L6	Unhelpful Suggestion	Revision suggestions are reasonable but not helpful.	1

Table 20: Our human-annotated quality categories of ACUs. A higher severity score indicates the worse performance of corresponding ACUs.

# Task Goal
Your goal is to summarize your final feedback entry list based on your meta-evaluation decisions. In your meta-evaluation decision, you have carefully analyzed all the feedback generated by various models and decided which feedback entries should be included in your final feedback entry list in the last conversation turn.

# Your Task
## 1. Reorganize the Helpful Feedback Entry List
Now, please reorganize the previous output and strictly abide by the following notes:
(1) Include all the feedback entries from all the models you think are helpful and have been considered "Yes" for inclusion. **Do NOT miss any helpful and essential feedback entries**;
(2) Appropriately summarize and consolidate multiple feedback entries with the same cited sentences from different models into one feedback entry. Ensure the summarized descriptions and suggestions contain helpful details in these multiple feedback entries. Also, ensure that the final feedback entry list does not have numerous feedback entries with duplicate content;
(3) If a flaw is labeled as E6 (not helpful for improvement) and the meta-evaluation acknowledges it, it is optional whether to remove this feedback entry based on your preference. Always remember your goal is to generate "helpful and valuable" feedback entries that are beneficial for refinement;
(4) If some problematic feedback entries (not labeled as E0 or the consideration is "No") could become more reasonable and valid after being revised according to your meta-evaluation description, and these feedback entries have not been considered in other helpful feedback entries, please also revise these feedback entries and supplement them to your final output;
(5) Each feedback entry contains only one criteria. Do NOT assign multiple criteria to one feedback entry. If the sentence has numerous flaws, please list them in multiple feedback entries.

## 2. Summarize
### 2.1 Summarize Your Analysis
Please summarize and describe the performance of evaluated response on each first-tier primary criteria.
### 2.2 Generate Your Judgements
In the end, you should provide your final judgement score, ranging from 1 to 10. The score ranges and definitions are shown as follows:
1. $1 \leq x < 3$: The quality is very low, containing numerous severe flaws; there are also other flaws, with Important error criteria.
2. $3 \leq x < 5$: The quality is low, making it difficult to fulfill user query; There are many flaws, and a small number of severe flaws may be included.
3. $5 \leq x < 7$: The quality is moderate, somewhat addressing the user query; There are a few errors, and a small number of severe errors may be included.
4. $7 \leq x < 9$: The quality is approximately the same as the reference response (with the reference response scoring around 8). The response effectively answers user query.
5. $9 \leq x < 10$: The quality is better than the reference, perfectly answering the user query in the conversation history.

## NOTICE!!!
1. Quality scores (1-10) can be expressed as floating-point numbers.
2. Within specific score ranges, the more flaws there are, the lower quality score, and vice versa.
3. You should compare the evaluated response the reference before giving your quality score. Please follow the important guideline as follows: if evaluated response is worse than the reference, its score should be lower.

Figure 7: The prompt for generating final critiques and summarization for the evaluated responses, which is used for the supervised fine-tuning stage.

Now, you are a helpful assistant aiming to provide valuable critiques and analysis for the previous conversation history, thereby assisting in the analysis of the quality of subsequent responses in relation to this conversation history history. Now, we have provided our criteria list (maybe empty) for you from different evaluation perspectives as below.
—
# Our Provided Criteria List
{user_pre_defined_criteria}
—

# Your Tasks
## Supplement and Decompose the Criteria
Generate the criteria list of the two-tier structure: (1) The first-tier structure consists of primary criteria, i.e., the evaluation dimensions broadly conceptualized and distinct based on conversation history; (2) The second-tier structure decomposes these primary evaluation dimensions into several fine-grained and precise criteria based on the information in conversation history. **Note that our provided criteria list are only the primary criteria list (first-tier) without the fine-grained criteria definition (second-tier).**

### 2.1 If our provided criteria list is **EMPTY**

Please directly generate this two-tier criteria structure from scratch.
Do NOT generate redundant criteria; keep the final criteria precise, helpful, and concise.

### 2.2 If our provided criteria list is **NOT EMPTY**

Firstly, you should keep all our provided criteria as the primary criteria in your final output. You could expand other primary criteria not considered in our provided criteria but are essential for analyzing flaws in responses for previous conversation history.
1. **But NEVER expand primary criteria that conflict with our provided criteria.**
2. **NEVER generate criteria that are redundant with our provided criteria.**
3. **Do NOT miss any criteria that exists in our provided criteria list.**
Secondly, you should decompose these primary criteria into several fine-grained and precise criteria by considering the conversation history.

### 2.3 NOTICE!!!
Keep the number of all fine-grained criteria within 15, and each primary criterion includes no more than 3 fine-grained criteria.

# Output Template
Generate the task description in following Markdown template. Do NOT add comment (//) in the template.
—
# Two-tier Structure of Criteria
// a block for one primary criteria consisting of no more than 3 fine-grained criteria. Keep output following structure in order. Variable in '{{}}' should be replaced.
## {{Name of First Primary Criteria}}
// a string of the description and details of this first-tier primary criteria
Description: {{description}}

### {{Name of Fine-grained Criteria}}
// a string of the description and details of this second-tier
fine-grained criteria
Description: {{description}}
// a word reflects the significance of fine-grained criteria, select degree from three types (least to most significance): (1) normal; (2) medium; (3) important Degree: {{degree}}
...
—

Figure 8: The prompt for generating two-tier structured criteria. We also allow user to input their specific evaluation criteria.

# Task Input
We provide the evaluated response that responds to the conversation history as below.

—

{evaluated_response}

—

## NOTICE!!!
1. The conversation history represents the conversations before we provided the criteria list.
2. The evaluated response contains citation symbols, like [S1] and [S2] ([S1] means sentence 1), which represent the ID of their preceding sentences and are helpful for our following analysis.
3. Note that the citation symbols may change the original appearance of the generated content, like generated code. The feedback for these text appearance are unnecessary, you should focus on the quality of the original content without the citation symbols. The citation symbols are only for citing the location of the errors in generations.

# Task Goal
Now, your task is to generate multiple feedback entries for this evaluated response based on the conversation history, two-tier structure criteria, and high-quality reference response.
Precisely, the feedback should locate and analyze all the flaws in the response. Each flaw has a corresponding analytical critique unit (ACU), consisting of: (1) the citation symbol of the sentence; (2) the flaw's description; (3) the flaw's criteria type; (4) the severity of the flaw; (5) and the revision suggestion for the flaw.

## Please Strictly Abide by Following Rules:
(1) Please Do NOT critique and analyze these citation symbols, like [S1] and [S2], since they only highlight its preceding sentence in the response;
(2) Do NOT critique and analyze the sentences that are free from any flaws;
(3) Each feedback entry contains only one criteria. **Do NOT add multiple criteria in one feedback entry. If you think the sentence have multiple flaws, please list them into multiple feedback entries.
(4) Each flaw in the feedback entry should follow one fine-grained second-tier criterion. Only select the primary first-tier criteria when all its second-tier fine-grained criteria are inappropriate.

# Output Format
Please answer in following Markdown format template. Do NOT add comment (//) in the template.

—

// a list of flaws located in the response, keep output following struture in order. Replace ‘{{}}‘ with your generations.
# List of Flaws in Response

## Feedback Entry 1
// Mark the location of the sentences that contain flaws with their corresponding citation symbols (like [S1] and [S2]). If the error spans multiple sentences, please output their related multiple citation symbols sequentially, like "[S1] [S2] [S3]".
Citation Symbol: {{citation_symbol}}
// a string describes the flaws
Description: {{description}}
// a string reflects the criteria for this flaw, which should be selected from the criteria defined above. Note that only one criteria is allowed for each feedback entry. Do NOT add multiple criteria in one feedback entry.
Belong to Criteria: {{criteria_name}}
// a string reflects the severity of this flaw. Select the severity of flaws from four types (least to most severe): (1) Negligible; (2) Minor; (3) Moderate; (4) Severe.
Severity: {{severity}}
// a revision suggestion for this flaw
Suggestion: suggestion
...

—

Figure 9: The prompt for generating analytical critiques for evaluated responses.

# User Query
What would you do if you found a dead body in a hotel room?

# Two-tier Structure of Criteria
## Completeness of Instruction Following
Description: The response must cover all necessary actions without omitting any critical steps in dealing with the discovery of a dead body in a hotel room.
### Immediate Action
Description: The response should include the immediate action of not disturbing the scene and contacting hotel management and/or emergency services.
Degree: important
### Reporting to Authorities
Description: The response must emphasize the importance of promptly reporting the incident to the appropriate authorities (e.g., police, emergency medical services).
Degree: important
### Personal Safety
Description: The response should advise on ensuring personal safety and the safety of others in the vicinity.
Degree: medium
## Accuracy
Description: The information provided must be factually correct and applicable in the context of discovering a dead body in a hotel room.
### Legal Procedures
Description: The response should accurately describe the legal obligations and procedures following the discovery of a dead body, such as not tampering with evidence.
Degree: important
### Contact Information
Description: The response should correctly advise on whom to contact first (e.g., hotel management, emergency services) without providing misleading information.
Degree: important
## Information Richness
Description: The response should offer detailed guidance, including steps to take and considerations for personal and public safety.
### Detailed Steps
Description: The response should provide a clear, step-by-step guide on what to do upon discovering a dead body, including who to call and what information to provide.
Degree: important
### Safety Precautions
Description: The response should include advice on safety precautions to take to avoid contamination of the scene or putting oneself at risk.
Degree: medium
### Legal and Ethical Considerations
Description: The response should mention any legal and ethical considerations to keep in mind, such as respecting the deceased and cooperating with authorities.
Degree: medium
...

Figure 10: One case of customized evaluation criteria for one query. ## represents the first tier criteria and ### represents the second tier criteria, which is a more fine-grained and specific evaluation criteria for the first tier criteria.

# User Query
You are a viking conquering enemy lands. Your name is

# Evaluated Response
I am a Viking conqueror named Bjorn Ironside [S1]. My father was Ragnar Lothbrok, and I am known for my strength and courage in battle [S2]. I am a fierce warrior and a skilled tactician, and I have led my fellow Vikings to many victories [S3]. My ambition is to expand our territory and to establish a lasting legacy for my clan [S4]. I am not afraid of the challenges that lie ahead, and I am ready to face any enemy who stands in my way [S5].
# Two-tier Structure of Criteria

# The List of ACUs
Citation Symbol: [S2]
Description: The narrative lacks historical and cultural depth regarding Bjorn Ironside's background and personal attributes, missing an opportunity to enrich the story with Viking era specifics.
Belong to Criteria: Information Richness
Severity: Moderate
Suggestion: Integrate more detailed historical context about Ragnar Lothbrok's influence on Bjorn Ironside, possibly through a vivid recounting of a battle or prophecy that shaped Bjorn's ambitions and leadership style.
...
Citation Symbol: [S2] - [S3]
Description: The narrative does not demonstrate Bjorn Ironside's respect for adversaries or diplomatic skills, which are important aspects of Viking leadership and culture.
Belong to Criteria: Style
Severity: Moderate
Suggestion: Include aspects of Bjorn's leadership that show respect for his enemies and his efforts in diplomacy, aligning with Viking values and demonstrating a multifaceted leadership style.
...

# Summarization
1. Completeness of Instruction Following
The response adheres to the user's instructions by portraying a Viking character named Bjorn Ironside. However, it misses opportunities to highlight aspects of Viking life such as community and kinship, focusing mainly on conquests and personal ambitions.
2. Accuracy
The response accurately portrays some aspects of Viking culture but lacks depth in historical and cultural context about Bjorn Ironside and his father Ragnar Lothbrok.
3. Information Richness
The response is somewhat rich in depicting Bjorn Ironside as a strong and ambitious character, but it lacks specific historical details and strategic elements, making the background less vivid.
4. Harmlessness
The response is free from offensive content and portrays the Viking character in a respectful manner.
5. Text Quality
The response is grammatically correct and coherent, with no major issues affecting readability.
6. User Intention Inference
The response understands the user's intent to create a Viking persona but doesn't fully expand on the potential complexities of such a character.
7. Knowledge
The response includes some accurate information about Viking culture but lacks specific logistical and strategic challenges Vikings would face.
8. Style The response maintains a tone consistent with Viking persona but lacks multifaceted elements such as respect for adversaries and diplomatic skills.
# Final Judgement
Score: 6.2

Figure 11: One case of critiques in MultiCritiqueDataset-SFT. Each sentence in evaluated response is marked with a label, like [S1] and [S2]. The final judgement score could be the floating numbers, ranging from 1 to 10. Except for flaws in each sentence, GPT-4 also locate flaws across sentences, like the **"Moderate —Style"** flaw across sentence 2 and sentence 3 "[S2] - [S3]".