Training Critique Models for Better Refinement

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated remarkable evaluation and critique capabilities, providing insightful feedback and identifying flaws in various tasks. These critique abilities have shown great potential in improving the performance of LLMs. However, limited research has explored which types of critiques are most effective for improving model responses or how to generate such critiques. To address this gap, we introduce Refinement-011 oriented Critique Optimization (RCO), a novel framework designed to train critic models using refinement signals. By evaluating refinement performance, RCO identifies effective critique strategies for improving model outputs and learns to generate these critiques. Extensive experiments demonstrate that RCO significantly 018 019 outperforms conventional LLM-generated critiques in refining responses. Notably, RCO not only enhances the policy model used during training but also exhibits strong transferability, effectively aiding other models in response refinement. Our code and data will be publicly available upon acceptance of this paper.

1 Introduction

027

042

Large language models (LLMs) (OpenAI, 2022; Achiam et al., 2023) have demonstrated remarkable performance across diverse tasks (Laskar et al., 2023; Ahn et al., 2024), ranging from instruction following (Ouyang et al., 2022) to reasoning (Plaat et al., 2024) and question answering (Allemang and Sequeda, 2024). However, their black-box nature often results in factual inaccuracies and hallucinations (Zhang et al., 2023), underscoring the need for robust mechanisms to ensure reliability and accuracy. critic models, designed to critically evaluate LLM outputs and generate natural language critiques, have emerged as a promising solution (Pan et al., 2024; Lan et al., 2024b; Chang et al., 2024). These models aim to provide comprehensive assessments, accurately identify errors,



Figure 1: A comparison between previous methods, which annotated preferences based on the quality of critique and not helpful for refinement, and our method, which annotate preference of refined responses and is helpful for refinement.

and suggest constructive improvements, enabling a feedback loop for enhanced LLM performance (Li et al., 2023; Kim et al., 2024). Recent advancements in critic model training involve the use of human-curated datasets and alignment algorithms, such as supervised fine-tuning (SFT) and reinforcement learning (Cui et al., 2023). These approaches have yielded encouraging results in enabling critic models to judge and detect flaws in LLM-generated responses effectively.

043

044

045

047

051

054

057

060

061

062

063

064

065

067

068

However, existing methods for training critique models, primarily focused on enhancing their ability to assess answer quality, face significant limitations (Li et al., 2023; Cui et al., 2023). These models, although proficient at generating "good" critiques, often fail to facilitate meaningful refinements in the responses of actor models. This problem arises because the critiques, while high-quality from a human perspective, are often difficult for actor models to interpret and act upon effectively. Furthermore, training a critique model typically requires large-scale human-annotated datasets (Li et al., 2023; Lan et al., 2024a) or human-curated error cases (McAleese et al., 2024), both of which are resource-intensive to produce and challenging to generalize. The inherent subjectivity of critique

087

094

100

101

102

103

104

105

107

111

117

quality adds another layer of complexity, leading to inconsistencies and biases in data collection that further undermine the effectiveness of these models (Sun et al., 2024).

To address these challenges, we propose Refinement-oriented Critique Optimization (RCO), a novel training paradigm for critic models. As shown in Figure 1, unlike conventional methods (Ke et al., 2024; McAleese et al., 2024) that rely on directly annotating preferences for critiques-a task that is both labor-intensive and prone to subjectivity—our approach integrates critique evaluation into the refinement process. Specifically, we feed each critique along with the initial response into the actor model (the model generating the initial response) and prompt it to produce multiple refined responses based on the critique. Subsequently, preferences between the refined responses and the initial response are annotated, and we quantify the proportion of refinements that are preferred. This proportion serves as the reward signal for training the critic model. By focusing on refinement outcomes, our method avoids the challenges of direct critique preference annotation and ensures that critiques leading to better refinements receive higher reward scores. Consequently, critic models trained with RCO are better equipped to generate actionable critiques that drive significant improvements in policy model outputs. We evaluate RCO across five tasks-dialog generation, summarization, question answering, mathematical reasoning, and code generation-using baseline models without reinforcement learning, models trained with Direct Preference Optimization (DPO) (Rafailov et al., 2023), and advanced open-source models. Experimental results reveal that RCO substantially enhances the referee capabilities of critic models, outperforming existing open-source approaches across multiple benchmarks.

The contribution of our work are three-fold: (1) We propose a method for training critic models 109 that prioritizes generating actionable critiques, en-110 abling more effective refinement of policy model responses. This approach addresses the limitations of 112 existing methods by aligning critiques with the im-113 provement of policy outputs. (2) Instead of directly 114 annotating preferences for critiques-a challeng-115 116 ing and error-prone process-we introduce a novel annotation scheme that focuses on preferences for refined responses. This approach significantly re-118 duces the cost and inaccuracies associated with 119 critique annotation while ensuring that critiques 120

leading to meaningful refinements are rewarded appropriately. (3) Our method is rigorously evaluated across five diverse tasks, demonstrating substantial improvements in critique quality and refinement capabilities compared to existing methods. Additionally, we conduct an in-depth analysis to explore the effectiveness and impact of our approach, providing valuable insights into the alignment of critic models with policy model improvement.

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

2 **Related Work**

Critique Ability of LLMs The rapid development of large language models (LLMs) has underscored the importance of enhancing their critique capabilities. Advanced models like GPT-4 (Achiam et al., 2023) have proven effective evaluators (Li et al., 2023; Kim et al., 2024; Cao et al., 2024). However, most state-of-the-art LLMs are only available via APIs, prompting researchers to reduce costs and improve evaluation stability by collecting critique data from these models and fine-tuning open-source models in a supervised manner. Despite these efforts, quality issues persist due to the complexity of critique tasks. To address this, Murugadoss et al. (2024) employs prompt engineering to design metrics for assessing critique ability. Verga et al. (2024) proposes "evaluation committees" of multiple LLMs to reduce model bias. Ke et al. (2024) generated golden critique dataset from paired model responses and reference responses and fit a critic model to the dataset, whileLan et al. (2024a) uses a multi-agent framework to collect and filter preference-based critique data. In contrast, our method improves critique data quality by annotating refined responses, with their preferred rates serving as rewards for training evaluator models. This method directly enhances critique quality by leveraging the refined responses.

Preference-Based Reinforcement Learning Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al., 2019) is widely used to guide LLMs in generating human-preferred responses. Scheurer et al. (2023) trained reward models via RLHF using human-annotated pairwise comparisons. Recent approaches have applied RLHF to improve LLM critique abilities, such as CriticGPT (McAleese et al., 2024), which trains critic models using human-crafted errors in code generation, and Wang et al. (2024), who collects preference critique pairs by comparing LLMgenerated and human-annotated scores. However,



Figure 2: The illustration of our method RCO, describing our data collection and training process.

these methods are limited by the high cost of human annotations and the uncertain quality of critique datasets (Sun et al., 2024). In contrast, our approach reduces reliance on human annotations, offering a clear standard for good critiques that effectively help actor models refine their responses.

3 Methodology

178

179

181

183

187

188

189

The overall structure of our methodology are illustrated in Figure 2.

3.1 Formulation

Our approach begins with a prompt dataset, denoted as \mathcal{D} . For each prompt $x \in \mathcal{D}$, an initial response y_0 is sampled using an actor model $\pi(y_0|x)$. Subsequently, the base critic model $p(c|y_0, x)$ is employed to generate N distinct critiques, denoted as c_1, c_2, \ldots, c_N , for the initial response y_0 . Each critique c_i is then fed back into the actor model π , which produces M distinct refined responses, $y_{i1}, y_{i2}, \ldots, y_{iM}$. The distribution for sampling these refined responses is denoted as $\pi_{c_i}(y|c_i, y_0, x)$.

Refinement Preferred Rate. In our framework,
the refinement preference rate serves as the reward
signal for training the critic model. Specifically,

the refinement preferered rate $R(c_i|y_0, x)$ for a given critique c_i is defined as the probability that a response y, sampled from $\pi_{c_i}(y|c_i, y_0, x)$, is preferred over the initial response y_0 :

$$R(c_i|y_0, x) = \mathbb{E}_{y \in \pi_{c_i}} P(y \succ y_0) \tag{1}$$

195

196

197

199

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

To estimate $R(c_i|y_0, x)$, we approximate it using the following sampling-based approach, by defining a preference score (PS) for the revised response y_{ij} and the initial response y_0 :

$$R(c_i|y_0, x) \approx \frac{1}{M} \sum_{j=1}^M PS(y_{ij}, y_0)$$
 (2)

In this equation, the preference score $PS(y_{ij}, y_0)$ is determined by annotators according to the following rules:

- 1. $PS(y_{ij}, y_0) = 1$ if the refined response y_{ij} is preferred over the initial response y_0 .
- 2. $PS(y_{ij}, y_0) = 0.5$ if both responses are considered equally good.
- 3. $PS(y_{ij}, y_0) = 0$ if the initial response y_0 is preferred over the refined response y_{ij} .

This process is repeated for each critique c_i , and the resulting refinement preference rates are used as rewards during the training of the critic model.

218

219

222

223

225

228

229

234

follows:

240

	TabMWP (Lu et al., 2022)
Code	HumanEval (Zheng et al., 2023)
Generation	DS-1000 (Lai et al., 2022)

Reasoning

Table 1: All 14 source datasets and the amount for

Total

prompt dataset collection. **Training Data Collection** 3.2 Dataset Overview. The first step of our method involves the collection of the prompt dataset \mathcal{D} ,

which consists of five distinct tasks: dialog gener-

ation, summarization, question answering, math-

ematical reasoning, and code generation. These

Dialog HH-RLHF (Bai et al., 2022) 2,000 TL:DR (Stiennon et al., 2020) 710 Summarization CNN DailyMail (See et al., 2017) 1,000 Commonsense QA (Talmor et al., 2019) 500 Trivia QA (Joshi et al., 2017) 500 Ouestion AmbigQA (Min et al., 2020) 500 Answering ARC-Challenge (Clark et al., 2018) 500 ELI5 (Fan et al., 2019) 500 MathQA (Amini et al., 2019) 500 500 Mathematical TheoremOA (Chen et al., 2023a)

> AQuA (Ling et al., 2017) TabMWP (Lu et al., 2022)

(4)
where each $R(c_i y_0, x)$ for critique c_i is determined
by Equation 2. Compared to traditional preference-
based learning methods such as Direct Preference
Optimization (DPO) (Rafailov et al., 2023), DRO
leverages scalar reward values more effectively, en-
abling the critic model to learn a more nuanced re-
ward representation. This advantage allows DRO to
better capture complex reward structures, thereby
improving overall model performance.
· · · · · · · · · · · · · · · · · · ·
Task Dataset Amount

 $V_{\beta}(y_0, x) = \beta \log \mathbb{E}_{p(c|y_0, x)} \exp\left(\frac{1}{\beta} R(c|y_0, x)\right)$ $\approx \beta \log \frac{1}{N} \sum_{i=1}^{N} \exp\left(\frac{1}{\beta} R(c_i | y_0, x)\right)$

where $V_{\beta}(y_0, x)$ is the value function. This term

can be approximated via sampling:

 $\mathcal{L}_{\text{DRO}}(p_{\theta}) = \frac{1}{2N} \sum_{i=1}^{N} \left(R(c_i | y_0, x) - V_{\beta}(y_0, x) \right)$ $-\beta \log \frac{p_{\theta}(c_i|y_0, x)}{p(c_i|y_0, x)} \Big)^2$ (3)

Training Objective. The Direct Reward Opti-

mization (DRO) (Richemond et al., 2024) objec-

tive is employed as the training objective in our

experiment. The DRO loss function is defined as

Collection of Initial Responses. To generate the initial responses, we utilize four distinct ac-

tor models: LLaMa-2-7B-Chat, LLaMa-2-13B-Chat, LLaMa-2-70B-Chat, and LLaMa-3-8B-Instruct (Touvron et al., 2023). These models generate responses to the 10,000 prompts, resulting in 40,000 unique responses. From this pool, 1,200 responses are randomly selected to form the test set, ensuring that 800 responses are assigned to each task and 1,000 responses are allocated to each actor model. The remaining responses are used to construct the training set. For each critic model, 2,000 responses are selected from each actor model's output, resulting in a total of 8,000 responses for training. Crucially, the prompts in the training and test sets are disjoint, ensuring unbiased evaluation.

Critique Generation. To facilitate the critique process, we employ five distinct base critic models: LLaMa-2-7B-Chat, LLaMa-2-13B-Chat, LLaMa-3-8B-Instruct, Auto-J-13B (Li et al., 2023), and UltraCM-13B (Cui et al., 2023). Each model is tasked with generating critiques for the initial responses. Specifically, each critic model produces N = 4 critiques for every initial response y_0 in the training set. This process ensures a more accurate approximation of the normalization constant $Z_{\beta}(y_0, x)$, contributing to the refinement of the training procedure.

Refinement Generation Based on Critiques. During the refinement phase, the actor model that generated the initial response y_0 is responsible for refining its output based on the critiques it receives. For each critique c_i , the actor model generates M = 5 distinct refined responses. This multipleresponse approach enables a better approximation of the refinement preference rate $R(c_i|y_0, x)$, allowing for a more nuanced assessment of the model's ability to improve based on feedback.

Refinement Preferred Rate Calculation. As depicted in Figures 1 and 2, we calculate the refinement preferred rate $R(c_i|y_0, x)$ as the reward signal for each critique. To achieve this, we utilize the Qwen-2.5-72B-Instruct (Team, 2024) model to annotate the preference of each refined response y_{ii} in comparison to its corresponding initial response

500

500

820

970

10,000

tasks are sourced from a total of 14 different datasets, with additional details provided in Table 1. In total, 10,000 unique prompts are gathered for our experiment.

289

241

242

243

244

245

246

247

248

250

251

252

253

254

255

 y_0 . The preference between responses is calcu-290 lated according to the rules outlined in Section 3.1, 291 where we compute $P(y_{ij} \succ y_0)$. To mitigate positional bias in the annotation process, we alternate the positions of the refined response and the initial response for each annotation. This ensures that each response is evaluated in both positions. The 296 final refinement preferred rate $R(c_i|y_0, x)$ for each critique c_i is computed as the average of 2M=10individual annotations. This approach ensures a robust and balanced evaluation of the refined responses, accounting for any potential bias in the 301 annotation process. We report the prompts we used 302 during data collection in Appendix A.

4 Experiment

In this section, we describe the benchmarks, evaluation metrics, baselines, and experimental results used to assess the performance of our proposed method.

4.1 Benchmarks and Evaluation Metrics

In our experiment, we use the test dataset described in Section 3.2 for evaluation of refinement preferred rate, and CriticEval (Lan et al., 2024b) and CriticBench (Lin et al., 2024) benchmark for human evaluation.

Test Dataset The first benchmark is the test 315 dataset described in Section 3.2, which consists 316 of 1,200 distinct responses. For each prompt, we 317 generate a critique from the critic model, followed 318 by a refined response generated by the actor model. 319 Both the critique generation and refinement pro-320 cesses employ greedy sampling. We use two met-321 rics for this benchmark: (1) average preference 322 score (PS) and (2) average response quality score 323 (RQS). For preference score, we prompt GPT-4 324 evaluator to return the preference of each refined 325 response compared to its corresponding initial re-326 sponse. Subsequently, we calculate the average preference score as the definition provided in Sec-328 tion 3.1. For average response quality score, we prompt GPT-4 to provide a preference score for 330 each of the initial response and all refined responses 332 on a scale from 1 to 10, where a higher score indicates a better response. We report the prompts we used during valuation in Appendix B.

335 CriticEval and CriticBench The second bench336 mark combines two widely used evaluation
337 datasets, CriticEval and CriticBench, to assess the

critique ability of various models across multiple338tasks. Human evaluation is used to assess the qual-339ity of critiques and refinements. Specifically, the340evaluation is structured as follows:341

- 1. We directly compare the critiques generated by our method and the baselines.
- 2. We generate refinements for both the critiques produced by our method and the baselines using *LLaMa-2-7B-Chat*, and the refinements are then compared for quality based on human assessment.

For the human evaluation, we sample 200 responses from the benchmark, ensuring that each task has 40 responses, and engage 3 NLP researchers to annotate the preferences of critiques and refinements. Given that the code generation task in CriticEval overlaps with our training dataset, and that CriticBench contains only reasoning and code generation tasks, we select the code generation task from CriticBench and the remaining four tasks from CriticEval. Notably, we ensure that the prompts in the test sets are from disjoint source datasets with the prompt in the training sets, ensuring unbiased evaluation.

4.2 Baselines

We evaluate our method against the following four types of baselines:

- 1. **Base Critic Models:** The five base critic models described in Section 3.2.
- 2. Self-refinement Results: According to prior works (Madaan et al., 2023; Chen et al., 2023b), directly prompting the actor model itself to revise its own response without any critiques proves a strong baseline. Following previous works (Akyürek et al., 2023), we add Self-refinement as a baseline for our experiment. For each initial response in the test set, we use the actor model that generated it to directly refine its own response into a better one. We then evaluate the average preference score and response quality score of the refined response.
- 3. Large Open-Source LLMs: We compare against large models such as *LLaMa-2-70B-Instruct* and *LLaMa-3-70B-Instruct*.

313

304

305

362 363

343

344

346

348

350

351

352

354

355

356

357

358

360

361

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

381

382

Critique	Method	Dialog		Summ.		QA.		Math		Code		Overall	
Model		PS	RQS	PS	RQS	PS	RQS	PS	RQS	PS	RQS	PS	RQS
Initial	Answer	_	6.37	-	6.42	_	6.06	-	4.48	_	4.22	-	5.51
					BASE	LINES							
LLaMa-2	2-70B-Chat	91.46	7.83	88.12	8.20	78.33	7.33	62.50	5.09	65.21	5.49	77.12	6.79
LLaMa-3-	70B-Instruct	91.25	7.53	86.88	8.01	75.73	7.04	75.52	6.08	82.38	6.15	82.35	6.96
Self-re	finement	87.29	7.34	79.37	7.77	71.25	6.46	59.62	5.02	69.37	5.47	73.38	6.41
Al	Aligner		6.30	50.16	6.64	47.52	6.19	54.61	4.90	38.00	4.39	46.92	5.68
CRITIC MODELS													
LL Ma 2	Base model	91.04	7.85	84.58	8.15	76.67	7.11	60.62	4.88	62.29	5.30	75.04	6.66
TD Chat	+DPO	94.58	7.95	83.75	8.27	86.04	7.65	61.46	5.09	70.00	5.11	79.17	6.81
/B-Cliat	+RCO (Ours)	95.42	7.99	92.08	8.30	93.51	8.06	65.83	5.61	74.27	5.69	84.22	7.13
II-M- 0	Base model	89.58	7.72	86.88	7.99	81.80	7.04	61.72	5.15	66.46	5.34	77.29	6.65
LLaMa-2-	+DPO	87.50	7.70	85.83	8.02	84.79	7.19	66.87	5.45	60.62	5.29	77.12	6.73
13B-Chat	+RCO (Ours)	92.92	7.88	90.62	8.08	92.71	7.81	67.08	5.73	72.71	5.49	83.21	7.00
	Base model	87.50	7.70	88.33	7.94	74.27	7.04	65.06	5.53	79.17	6.03	78.87	6.85
LLaMa-3-	+DPO	91.46	7.81	91.25	8.14	73.74	7.13	64.58	5.34	67.78	5.37	77.76	6.76
8B-Instruct	+RCO (Ours)	94.17	7.84	92.08	8.29	94.17	7.82	75.73	6.30	82.50	6.24	87.73	7.30
	Base model	79.79	7.66	89.17	8.08	71.55	6.84	59.58	4.57	66.95	5.50	73.41	6.53
Auto-J-13B	+DPO	87.50	7.85	84.58	8.11	84.38	7.47	67.23	5.57	68.54	5.56	78.45	6.91
	+RCO (Ours)	91.25	7.91	93.13	8.20	90.42	7.83	71.46	5.64	71.34	5.58	83.52	7.03
Lilta CM	Base model	70.00	7.19	76.46	7.66	63.54	6.58	64.58	5.05	69.58	5.54	68.83	6.40
12D	+DPO	79.17	7.50	86.25	7.95	73.12	6.76	62.71	4.83	64.38	5.36	73.13	6.48
13B	+RCO (Ours)	91.46	7.76	94.58	8.06	88.12	7.43	67.44	5.53	77.71	5.71	83.86	6.90

Table 2: Evaluation results of our method and baselines, in terms of preference score (PS) and refinement quality score (RQS). Summ. and QA. are the shorter form of summarization and question answering tasks, respectively.

4. *Aligner-7B-V1.0: Aligner* (Ji et al., 2024) is a model-agnostic plug-and-play module that directly refines the response into a better one. It is trained on dialog generation tasks from the base model *LLaMa-2-7B*. In our experiment, we prompt the model *Aligner-7B-V1.0* to generate refined responses and then evaluate the average preference score and response quality score of them.

384

385

389

5. DPO-trained Critic Models: We use the same annotator model, Qwen-2.5-72B-Instruct, to directly annotate critique preference pairs in the training dataset. For the N = 4 setting used in our experiments, we annotate the preference between critique pairs (c_1, c_2) and (c_3, c_4) . After filtering out in-398 valid and inconsistent annotations, we gather around 11,000 preference pairs for each critic 400 model. These preference pairs are used to 401 402 train critic models via the DPO algorithm.

4.3 Experimental Results on the Test Dataset

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

We evaluate the performance of our method, trained on all base critic models, against the baseline models on the test dataset. The results, presented in Table 2, demonstrate that our method significantly outperforms all baseline models, both in terms of the refinement preferred rate and the quality of the refined responses. This indicates that our method effectively trains critic models to enhance the actor model's ability to revise and improve its responses.

Specifically, our approach surpasses the base critic model across all tasks, providing robust evidence for the benefits of training critic models. Furthermore, smaller models trained with our method consistently outperform larger models within the same model series (e.g., *LLaMa-2-7B-Chat* vs. *LLaMa-2-70B-Chat*, *LLaMa-3-8B-Instruct* vs. *LLaMa-3-70B-Instruct*), highlighting the competitive performance of our method even with relatively smaller models. Remarkably, our model exhibit superior results in terms of both PS and RQS against refinement baselines, selfrefinement and *Aligner*, showcasing the effectiveness of our generated critiques. Especially, our



Figure 3: Visualization of human evaluation results in terms of critique quality evaluation and refinement quality evaluation. The agreement rate of annotators for critique and refinement evaluation is 54.5% and 73%.



Figure 4: Refinement quality score of iterative refinement results.

method outperform self-refinement across all tasks with every base critic models, while other baselines fail on at least one task.

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

In contrast, the DPO-trained models show only marginal or no improvements over the base models and fail to outperform the larger models. This suggests that directly annotating critique preferences with large language models yields low annotation accuracy, which in turn limits the effectiveness of the DPO approach for training robust critic models.

When analyzing across different tasks, our method demonstrates a distinct advantage over larger baselines in the domain of question answering. Compared to base models and DPO-trained models, the superiority of our approach are most evident in mathematical reasoning and code generation tasks. In contrast, DPO models exhibit comparable performance to our method in dialog generation and summarization tasks. 443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

4.4 Human Evaluation

In the human evaluation, we focus on the base critic model that exhibited the best performance in the automated evaluation, *LLaMa-3-8B-Instruct*, due to the massive cost associated with human evaluation. The baseline models considered include the base model, the large model *LLaMa-3-70B-Instruct*, and the DPO-trained *LLaMa-3-8B-Instruct*. The overall results from human evaluation, summarized in Figure 3, align with the findings from the test dataset, demonstrating that our method outperforms the baseline models in

the quality of both critique and refined responses.
Specifically, our method consistently surpasses the
baseline models in terms of both the quality of
critiques and the quality of the refined responses.
These results underscore the efficacy of our approach in enhancing actor model responses through
structured critique generation and refinement.

An intriguing observation from the human eval-465 466 uation is that human preferences for critiques do not always align with the preferences for the refine-467 468 ments generated from those critiques. Specifically, critiques generated by LLaMa-3-70B-Instruct were 469 the least preferred by human annotators, and cri-470 tiques generated by DPO method are the most pre-471 ferred, even achieving comparable performance to 472 our method. However, all three baselines exhibit 473 474 similar performance on human preferences of refinements. This suggests that a critique considered 475 favorable by human annotators may not necessarily 476 result in a refined response that aligns with human 477 478 preferences, highlighting the inherent complexity of the critique-refinement process. 479

5 Analysis

480

481

5.1 Iterative Refinement

Previous studies have explored the enhancement of 482 483 LLMs through iterative self-critique and refinement processes. However, these approaches have been 484 critiqued in subsequent research, which suggests 485 that LLMs may not always improve through this 486 method. In contrast, our work aims to demonstrate 487 488 that critic models, trained using our refinementoriented methodology, can effectively facilitate 489 continuous improvement in the responses gener-490 ated by actor models during iterative critique and 491 refinement. We conducted experiments using two 492 base models, LLaMa-2-7B-Chat and LLaMa-3-8B-493 Instruct, subjecting them to a three-turn critique-494 refinement cycle. The quality of the refinements 495 was evaluated across each iteration for both our 496 method and several baseline approaches, with the 497 results presented in Figure 4. The findings indi-498 cate that our method consistently maintains a sta-499 ble upward trend in performance across iterations, 501 whereas the baseline methods show limited improvement after the second iteration. This progres-502 sive enhancement underscores the superiority of our approach in guiding the iterative refinement of actor models. 505

5.2 Case Analysis

To further investigate why our method produces more effective critiques for refining actor models, we selected two representative examples from the dataset. These cases, along with the critiques generated by our method and the baseline approaches, as well as the refinements produced by *LLaMa-2-7B-Chat*, are presented in Figure 5-9 in Appendix due to limitation of spaces.

The case analysis reveals that our proposed method is capable of generating correct and concise critiques, offering clear, actionable suggestions that are easy for the actor model to follow. In contrast, DPO models, which are trained using LLM-annotated preferences, tend to provide more detailed analyses but their suggestions are often vague and less specific. For instance, a recommendation such as "Use specific language" is too ambiguous for effective implementation by the actor model. Furthermore, DPO models occasionally misidentify the target of critique, mistakenly focusing on the article rather than the summary. These observations highlight the efficacy of our method in training critic models that generate more precise and helpful critiques for the iterative improvement of actor models, especially when compared to alternative approaches such as DPO.

6 Conclusion

In this paper, we presented RCO, a novel approach to training critic models to enhance the effectiveness of actor model refinement. We proposed a new annotation scheme that focuses on preferences for refined responses, significantly reducing the cost and inaccuracies associated with traditional critique annotations while maintaining the effectiveness of the critique process. Finally, we provide rigorous evaluations across five diverse tasks, demonstrating the substantial improvements in both critique quality and refinement capabilities when compared to existing methods. The in-depth analysis further highlights the effectiveness of our approach and its potential for advancing the alignment of critic models with policy model enhancement. These contributions offer valuable insights into the design of more efficient and scalable systems for model refinement, paving the way for future research in this area.

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

572

573

575

577

582

584

586

590

592

594

Limitation

Despite the effectiveness and strong potential of RCO for broader applications, several limitations 555 warrant further investigation and improvement. 556 One key limitation is the inaccurate estimation of $R(c_i|y_0, x)$ and $V_\beta(y_0, x)$. In our study, we sample 4 critiques to estimate $V_{\beta}(y_0, x)$ and 5 refined responses to estimate $R(c_i|y_0, x)$, which maybe not sufficient. However, to achieve more accurate estimates, more data and annotations are required, which increase the cost for data collection. Addi-563 tionally, our approach focus solely on critic models, 564 failing to train actor models for improved utiliza-565 tion of critiques for refinement. Moving forward, we aim to develop more efficient methods for training the critic model. Furthermore, we are interested 568 in advancing techniques for actor models to better interpret natural language critiques and leverage them to enhance their responses.

Ethical Consideration

In this work, we leveraged several available datasets to construct the training dataset of RCO. The HH-RLHF (Bai et al., 2022), TL;DR (Stiennon et al., 2020), Commonsense QA (Talmor et al., 2019), TheoremQA (Chen et al., 2023a) and TabMWP (Lu et al., 2022) are under MIT licenses; the CNN DailyMail (See et al., 2017), MathQA (Amini et al., 2019), AQuA (Ling et al., 2017) and HumanEval (Zheng et al., 2023) are under Apache licenses; the AmbigQA (Min et al., 2020), ARC-Challenge (Clark et al., 2018) and DS-1000 (Lai et al., 2022) are under CC BY-SA licenses; the ELI5 (Fan et al., 2019) dataset is under BSD license.

In these datasets, there exists some instructions with security issues. However, in RCO training, we constructed optimized prompt pairs that provide safety enhancements to these unsafe instructions, further mitigating the security issues.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

- Afra Feyza Akyürek, Ekin Akyürek, Aman Madaan, Ashwin Kalyan, Peter Clark, Derry Wijaya, and Niket Tandon. 2023. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs. *arXiv preprint arXiv:2305.08844*.
- Dean Allemang and Juan Sequeda. 2024. Increasing the llm accuracy for question answering: Ontologies to the rescue! *arXiv preprint arXiv:2405.11706*.
- 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.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- 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. *arXiv preprint arXiv:2410.16256*.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology, 15(3):1–45.
- Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023a. Theoremqa: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7901.
- Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023b. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.
- 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.
- Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. arXiv preprint arXiv:1907.09190.

605 606 607 608 609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

602

603

- 660 661
- 66
- 66
- 66
- 669 670 671 672 673 674 675
- 676 677 678
- 679 680
- 6
- 686 687
- 6
- 69 69
- 69 69

697 698

7

- 70
- 702 703
- 704

706 707

- 7
- 710
- 711 712

- Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, and Yaodong Yang. 2024. Aligner: Achieving efficient alignment through weak-to-strong correction. *arXiv preprint arXiv:2402.02416*.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, arXiv:1705.03551.
- Pei Ke, Bosi Wen, Andrew Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, et al. 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 (Volume 1: Long Papers)*, pages 13034–13054.
 - Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.
- Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Scott Wen tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2022. Ds-1000: A natural and reliable benchmark for data science code generation. *ArXiv*, abs/2211.11501.
- Tian Lan, Wenwei Zhang, Chengqi Lyu, Shuaibin Li, Chen Xu, Heyan Huang, Dahua Lin, Xian-Ling Mao, and Kai Chen. 2024a. Training language models to critique with multi-agent feedback. *arXiv preprint arXiv:2410.15287*.
- Tian Lan, Wenwei Zhang, Chen Xu, Heyan Huang, Dahua Lin, Kai Chen, and Xian-ling Mao. 2024b. Criticbench: Evaluating large language models as critic. *arXiv preprint arXiv:2402.13764*.
- Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. *arXiv preprint arXiv:2305.18486*.
- Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.
- Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. 2024. Criticbench: Benchmarking llms for critique-correct reasoning. *arXiv preprint arXiv:2402.14809.*
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

713

714

715

717

718

719

720

721

722

723

724

725

726

727

728

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.
- Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. 2024. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215*.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*.
- Bhuvanashree Murugadoss, Christian Poelitz, Ian Drosos, Vu Le, Nick McKenna, Carina Suzana Negreanu, Chris Parnin, and Advait Sarkar. 2024. Evaluating the evaluator: Measuring llms' adherence to task evaluation instructions. *arXiv preprint arXiv:2408.08781*.
- OpenAI. 2022. Introducing chatgpt. Accessed: 2023-12-10.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- 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.
- Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. 2024. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.
- Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, et al. 2024. Offline regularised reinforcement learning for large language models alignment. *arXiv preprint arXiv:2405.19107*.

868

869

870

871

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*.

767

776

778

779

780

781

782

783

784

785

787

790

795

796

800

801

802

809

810

811

812

813

814

815

816

817

818

819

821

822

- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008– 3021.
- Shichao Sun, Junlong Li, Weizhe Yuan, Ruifeng Yuan, Wenjie Li, and Pengfei Liu. 2024. The critique of critique. *arXiv preprint arXiv:2401.04518*.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.
- Qwen Team. 2024. Qwen2.5: A party of foundation models.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- 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. *arXiv preprint arXiv:2404.18796*.
- Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.
- Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Zihan Wang, Lei Shen, Andi Wang, Yang Li, Teng Su, Zhilin Yang, and Jie Tang. 2023. Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x. In

Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 5673–5684.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Data Construction Prompts

Since our dataset construction process involves multiple stages, including prompt generation, initial responses, critiques, and revised responses, and 5 different tasks, we design distinct prompts for each stage and task. The prompts for initial response generation are shown in Table 3. The prompts for critique collection are shown in Table 4-8. The prompts for refinement generation are shown in Table 11-15. The prompts for self-refinement generation are shown in Table 16-20. Note that we have distinct prompts for two code generation datasets, HumanEval and DS-1000, since a prompt format is already provided for DS-1000 in its repository. For Auto-J-13B and UltraCM-13B whose input prompt formats are given, we directly adopt their prompt format in data construction, training and testing, as shown in Table 9 and 10. For the rest base models, we use the designed input prompt formats.

For annotation, the prompt for annotating preferences for refinements are shown in Table 21-25, and the prompt for annotating preferences for critiques are shown in Table 26-30.

B Evaluation Prompts

Our evaluation process involves GPT-4 scoring and preference evaluation, and we design distinct prompts for each of the evaluation settings and task. The prompts for preference annotation and preference scoring are shown in Table 21-25, and the prompts for response quality scoring are shown in Table 31-35.

C Training Details

In our experiment, we train our method RCO on 5 base critic models: *LLaMa-2-7B-Chat*, *LLaMa-2-13B-Chat*, *LLaMa-3-8B-Instruct*, *Auto-J-13B*, and *UltraCM-13B*. We train the smallest model *LLaMa-2-7B-Chat* on 4 NVIDIA H800 80GB GPUs with a batch size of 2, a gradient accumulation of 4 and 100 warmup steps. We train the medium-sized model *LLaMa-3-8B-Instruct* on 4 NVIDIA H800

872	80GB GPUs with a batch size of 1, a gradient ac-
873	cumulation of 8 and 100 warmup steps. For the
874	largest base models LLaMa-3-8B-Instruct, Auto-
875	J-13B, and UltraCM-13B, we train each of these
876	models on 6 NVIDIA H800 80GB GPUs with
877	a batch size of 1, a gradient accumulation of 8
878	and 50 warmup steps. Each of these models are
879	fully trained for 5 epochs, with a learning rate of
880	1×10^{-6} and a linear warmup schedule. We use
881	$\beta = 0.1$ throughout our study. We adopt the same
882	hyper-parameter settings for DPO training.

Dialog Generation Directly adopt the original prompt **Summarization - TL;DR** Summarize the following Reddit Post: SUBREDDIT: {subreddit} TITLE: {title} POST: {post} Your Summary: Summarization - CNN-DailyMail Summarize the following news: {news} Your Summary: Question Answering - Multiple Choices (Commonsense QA, ARC-Challenge) Question: {prompt} Answer choices: {choices} Explain your reasoning. You must choose only one option from A to E. Your final answer should be a single letter from A to E, in the form [[answer]], at the end of your response. Your answer: Question Answering - Others (Trivia-QA, ELI5, AmbigQA) Question: {prompt} Your Answer: Mathematical Reasoning - Multiple Choices (MathQA, AQuA) Can you solve the following math problem? {prompt} Answer choices: {choices} Explain your reasoning. You must choose only one option from A to E. Your final answer should be a single letter from A to E, in the form [[answer]], at the end of your response. Your answer: Mathematical Reasoning - Table (TabMWP) Can you solve the following math problem according to the information given in the table? {prompt} Table: {table title} {table content} Explain your reasoning. Your final answer should be a single numerical number, in the form \boxed{answer}, at the end of your response. Mathematical Reasoning - Others (TheoremOA) Can you solve the following math problem? {prompt} Your final answer should be a single numerical number, in the form Explain your reasoning. \boxed{answer}, at the end of your response. **Code Generation - HumanEval** Complete the code following the instruction given in the comment, as well as the format and the indentation. {prompt} **Code Generation - DS-1000** Write a short code to solve the problem following the given format and indentation. Place the executable code between <code> and </code> tags, without any other non-executable things. {prompt}

Prompts for Initial Response Generation

Table 3: Prompts for collecting initial responses from actor models.

Prompts for Critique Generation **Dialog Generation**

You are an impartial judge. You are shown a dialog between a human and an AI assistant. Comment on the assistant's response to the dialog according to the criteria of helpful, harmless and correct. After that, provide suggestions for improving the assistant's response according to your comment, starting with "Suggestions for improvement:".

{prompt}

Assistant's Response: {response}

Your comment:

Table 4: Prompts for collecting critiques of dialog generation tasks, for *Llama-2-7B-Chat*, *Llama-2-13B-Chat* and *Llama-3-8B-Instruct* base and trained models.

Summarization - TL;DR

You are an impartial judge. You are shown a Reddit post and a summary. Comment on the summary by finding if it misses any key information from the post or contains any unnecessary information. After that, provide suggestions for improving the summary according to your comment, starting with "Suggestions for improvement:".

SUBREDDIT: {subreddit} TITLE: {title} POST: {post}

Assistant's Summary: {response}

Your comment:

Summarization - CNN-DailyMail

You are an impartial judge. You are shown a piece of news and a summary. Comment on the summary by finding if it misses any key information from the post or contains any unnecessary information. After that, provide suggestions for improving the summary according to your comment, starting with "Suggestions for improvement:".

{news}

Assistant's Summary: {response}

Your comment:

Table 5: Prompts for collecting critiques of summarization tasks, for *Llama-2-7B-Chat*, *Llama-2-13B-Chat* and *Llama-3-8B-Instruct* base and trained models.

Question Answering - Multiple Choices (Commonsense QA, ARC-Challenge)

You are an impartial judge. You are shown a question and an answer. Comment on the answer and find problems with it. After that, provide suggestions for improving the answer according to your comment, starting with "Suggestions for improvement:".

Question: {prompt} Answer choices: {choices}

Assistant's Answer: {response}

Your comment:

Question Answering - Others (Trivia-QA, ELI5, AmbigQA)

You are an impartial judge. You are shown a question and an answer. Comment on the answer and find problems with it. After that, provide suggestions for improving the answer according to your comment, starting with "Suggestions for improvement:".

Question: {prompt}

Assistant's Answer: {response}

Your comment:

Table 6: Prompts for collecting critiques of question answering tasks, for *Llama-2-7B-Chat*, *Llama-2-13B-Chat* and *Llama-3-8B-Instruct* base and trained models.

Mathematical Reasoning - Multiple Choices (MathQA, AQuA)

You are an expert on mathematics. You are shown a math problem and the answer to it. Comment on the answer and find problems with it. After that, provide suggestions for improving the answer according to your comment, starting with "Suggestions for improvement:".

Problem: {prompt} Answer choices: {choices}

Assistant's Answer: {response}

Your comment:

Mathematical Reasoning - Table (TabMWP)

You are an expert on mathematics. You are shown a math problem and the answer to it. Comment on the answer and find problems with it. After that, provide suggestions for improving the answer according to your comment, starting with "Suggestions for improvement:".

Problem: {prompt}
Table: {table_title}
{table_content}

Assistant's Answer: {response}

Your comment:

Mathematical Reasoning - Others (TheoremQA)

You are an expert on mathematics. You are shown a math problem and the answer to it. Comment on the answer and find problems with it. After that, provide suggestions for improving the answer according to your comment, starting with "Suggestions for improvement:".

Problem: {prompt}

Assistant's Answer: {response}

Your comment:

Table 7: Prompts for collecting critiques of mathematical reasoning tasks, for *Llama-2-7B-Chat*, *Llama-2-13B-Chat* and *Llama-3-8B-Instruct* base and trained models.

Code Generation - HumanEval

You are an expert on computer programming. You are shown a code completion according to the requirements presented in the comment line of the code. Evaluate the correctness and readability of the code, and find if it meet the presented requirements. After that, provide suggestions for improving the code according to your evaluation, starting with "Suggestions for improvement:".

{prompt} {response}

Your evaluation:

Code Generation - DS-1000

You are an expert on computer programming. You are shown a code that proposes to solve the coding problem. Evaluate the correctness and readability of the code completion, and find if it meet the presented requirements. Remember not to comment on anything between 'A:' and 'BEGIN SOLUTION'. After that, provide suggestions for improving the code according to your evaluation, starting with "Suggestions for improvement":

{prompt} {response}

Your evaluation:

Table 8: Prompts for collecting critiques of code generation tasks, for *Llama-2-7B-Chat*, *Llama-2-13B-Chat* and *Llama-3-8B-Instruct* base and trained models.

Auto-J-13B Prompt [INST] Write critiques for a submitted response on a given user's query, and grade the response: [BEGIN DATA] *** [Query]: {prompt} *** [Response]: {answer} *** [END DATA] Write critiques for this response. After that, you should give a final rating for the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]". [/INST]

Table 9: Prompt for collecting critiques for Auto-J-13B base and trained models.

UltraCM-13B Prompt

Given my answer to an instruction, your role is to provide specific and constructive feedback for me. You should find the best way for me to learn from your feedback and improve my performance. You should consider multiple aspects of my answer, including helpfulness, truthfulness, honesty, and to what extent the answer follows instructions.

Instruction
{prompt}
Answer
{answer}

Please act as a teacher and provide specific and constructive feedback. Besides describing the weaknesses of the answer, you should also provide specific suggestions to guide me toward understanding how to improve. Please note, however, that your suggestions should help me better complete the instructions, but you should not introduce new requirements that are not mentioned in the instructions. Your feedback should focus on enhancing my ability to think critically and respond accurately. However, never explicitly provide the reference answer, nor do polite phrases be required. Only respond with concise feedback in chat style. Finally, score the overall quality of the answer from 1 to 10, where 1 is the worst and 10 is the best.

Format ### Feedback Overall Score: [1-10] [Your feedback]

Feedback Overall Score:

Table 10: Prompts for collecting critiques for Ultra-CM-13B base and trained models.

Prompts for Refined Response Generation

Dialog Generation

You are shown a dialog between a human and an AI assistant. An impartial judge on AI assistants has made comments on the assistant's response to the dialog. Please revise the assistant's response to improve its quality according to the suggestions for improvement provided in the comment, starting with "My revised response:".

{prompt}

Assistant's Response: {response}

Comment by the judge: {critique}

Your revision:

Table 11: Prompts for collecting refined responses of dialog generation tasks.

Summarization - TL;DR

You are shown a Reddit post and a summary. An impartial judge has made comments on the summary. Please revise the summary to improve its quality according to the suggestions for improvement provided in the comment, starting with "My revised summary:".

SUBREDDIT: {subreddit} TITLE: {title} POST: {post}

Original Summary: {response}

Comment by the judge: {critique}

Your revision:

Summarization - CNN-DailyMail

You are shown a piece of news and a summary. An impartial judge has made comments on the summary. Please revise the summary to improve its quality according to the suggestions for improvement provided in the comment, starting with "My revised summary:".

{news}

Original Summary: {response}

Comment by the judge: {critique}

Your revision:

Table 12: Prompts for collecting refined responses of summarization tasks.

Question Answering - Multiple Choices (Commonsense QA, ARC-Challenge)

You are shown a question and an answer. An impartial judge has made comments on the answer. Please revise the answer to improve its quality according to the suggestions for improvement provided in the comment, starting with "My revised answer:".

Question: {prompt} Answer choices: {choices}

Original Answer: {response}

Comment by the judge: {critique}

Your revision:

Question Answering - Others (Trivia-QA, ELI5, AmbigQA)

You are shown a question and an answer. An impartial judge has made comments on the answer. Please revise the answer to improve its quality according to the suggestions for improvement provided in the comment, starting with "My revised answer:".

Question: {prompt}

Original Answer: {response}

Comment by the judge: {critique}

Your revision:

Table 13: Prompts for collecting refined responses of question answering tasks.

Mathematical Reasoning - Multiple Choices (MathQA, AQuA)

You are shown a math problem and an answer. An expert on mathematics has made comments on the answer. Please revise the answer to improve its quality according to the suggestions for improvement provided in the comment, starting with "My revised answer:".

Problem: {prompt} Answer choices: {choices}

Original Answer: {response}

Comment by the expert: {critique}

Your revision:

Mathematical Reasoning - Table (TabMWP)

You are shown a math problem and an answer. An expert on mathematics has made comments on the answer. Please revise the answer to improve its quality according to the suggestions for improvement provided in the comment, starting with "My revised answer:".

Problem: {prompt}
Table: {table_title}
{table_content}

Original Answer: {response}

Comment by the expert: {critique}

Your revision:

Mathematical Reasoning - Others (TheoremQA)

You are shown a math problem and an answer. An expert on mathematics has made comments on the answer. Please revise the answer to improve its quality according to the suggestions for improvement provided in the comment, starting with "My revised answer:".

Problem: {prompt}

Original Answer: {response}

Comment by the expert: {critique}

Your revision:

Table 14: Prompts for collecting refined responses of mathematical reasoning tasks.

Code Generation - HumanEval

You are shown a code completion according to the requirements presented in the comment. An expert on computer programming has made critiques and advice for improvement on the code. Please revise the code completion to improve its quality according to the suggestions for improvement provided in the critique, starting with "My revised code:".

------Critiques and Advice-----

{critique}

———Your Revision———

{prompt}

Code Generation - DS-1000

You are shown a code that proposes to solve the coding problem. An expert on computer programming has made critiques and advice for improvement on the code. Please revise the code completion to improve its quality according to the suggestions for improvement provided in the critique, starting with "My revised code:".

Table 15: Prompts for collecting refined responses of code generation tasks.

Prompts for Self-Refinement Response Generation

Dialog Generation

You are shown a dialog between a human and an AI assistant. Please revise the assistant's response to improve its quality according to your analysis, starting with "My revised response:".

{prompt}

Assistant's Response: {response}

Your revision:

Table 16: Prompts for collecting self-refinement responses of dialog generation tasks.

Summarization - TL;DR

You are shown a Reddit post and a summary of it. Please revise the summary to improve its quality according to your analysis, starting with "My revised summary:".

SUBREDDIT: {subreddit} TITLE: {title} POST: {post}

Original Summary: {response}

Your revision:

Summarization - CNN-DailyMail

You are shown a piece of news and a summary of it. Please revise the summary to improve its quality according to your analysis, starting with "My revised summary:".

{news}

Original Summary: {response}

Your revision:

Table 17: Prompts for collecting self-refinement responses of summarization tasks.

Question Answering - Multiple Choices (Commonsense QA, ARC-Challenge)

You are shown a question and an answer. Please revise the answer to improve its quality according to your analysis, starting with "My revised answer:".

Question: {prompt} Answer choices: {choices}

Original Answer: {response}

Your revision:

Question Answering - Others (Trivia-QA, ELI5, AmbigQA)

You are shown a question and an answer. Please revise the answer to improve its quality according to your analysis, starting with "My revised answer:".

Question: {prompt}

Original Answer: {response}

Your revision:

Table 18: Prompts for collecting self-refinement responses of question answering tasks.

Mathematical Reasoning - Multiple Choices (MathQA, AQuA)

You are shown a math problem and an answer. Please revise the answer to improve its quality according to your analysis, starting with "My revised answer:".

Problem: {prompt} Answer choices: {choices}

Original Answer: {response}

Your revision:

Mathematical Reasoning - Table (TabMWP)

You are shown a math problem and an answer. Please revise the answer to improve its quality according to your analysis, starting with "My revised answer:".

Problem: {prompt}
Table: {table_title}
{table_content}

Original Answer: {response}

Your revision:

Mathematical Reasoning - Others (TheoremQA)

You are shown a math problem and an answer. Please revise the answer to improve its quality according to your analysis, starting with "My revised answer:".

Problem: {prompt}

Original Answer: {response}

Your revision:

Table 19: Prompts for collecting self-refinement responses of mathematical reasoning tasks.

Code Generation - HumanEval

You are shown a code completion according to the requirements presented in the comment. Please revise the code to make it more correct and readable, starting with "My revised code:".

-----Original Code-{prompt} {response}

_____Your Revision_____

{prompt}

Code Generation - DS-1000

You are shown a code that proposes to solve the coding problem. Please revise the code to make it more correct and readable, starting with "My revised code:".

-----Original Code----

{prompt} {response}

———Your Revision—

Table 20: Prompts for collecting self-refinement responses of code generation tasks.

Dialog Generation [SYSTEM]

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

[USER]

[Conversation] {prompt} [The Start of Assistant A's Response] {answer_0} [The End of Assistant A's Response] [The Start of Assistant B's Response] {answer_1} [The End of Assistant B's Response]

Table 21: Prompt for refinement preference annotation and evaluation for dialog generation tasks.

Summarization

[SYSTEM]

Please act as an impartial judge and evaluate the quality of the summaries provided by two AI assistants to the {kind} displayed below. Your evaluation should consider whether their summaries include all key information from the original article and avoid false or unnecessary sentences. Your should decide which assistant's summary is better. Begin your evaluation by comparing both assistants' summaries and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

[USER]

[{kind}] {prompt} [The Start of Assistant A's Summary] {answer_0} [The End of Assistant A's Summary] [The Start of Assistant B's Summary] {answer_1} [The End of Assistant B's Summary]

Table 22: Prompt for refinement preference annotation and evaluation for summarization tasks. The "kind" field will be "Reddit post" or "News", conditioned to whether this prompt is from TL;DR or CNN-DailyMail dataset.

Question Answering [SYSTEM]

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

[USER]

[User Question] {prompt} [The Start of Assistant A's Answer] {answer_0} [The End of Assistant A's Answer] [The Start of Assistant B's Answer] {answer_1} [The End of Assistant B's Answer]

Table 23: Prompt for refinement preference annotation and evaluation for question answering tasks.

Mathematical Reasoning

[SYSTEM]

Please act as an impartial judge and evaluate the quality of the answers provided by two AI assistants to the math problem displayed below. Your evaluation should consider correctness and helpfulness. You will be given a reference answer, assistant A's answer, and assistant B's answer. Your job is to evaluate which assistant's answer is better. Begin your evaluation by comparing both assistants' answers with the reference answer. Identify and correct any mistakes. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

[USER]

[Math Problem] {prompt} [The Start of Reference Answer] {ref_answer} [The End of Reference Answer] [The Start of Assistant A's Answer] {answer_0} [The End of Assistant A's Answer] [The Start of Assistant B's Answer] {answer_1} [The End of Assistant B's Answer]

Table 24: Prompt for refinement preference annotation and evaluation for mathematical reasoning tasks.

Code Generation [SYSTEM]

Please act as an impartial judge and evaluate the quality of the code provided by two AI assistants to the requirements displayed below. Your evaluation should consider correctness and helpfulness. Your should decide which assistant's provided code is better. Begin your evaluation by comparing both assistants' codes and provide a short explanation. Identify and correct any mistakes. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie. **[USER]**

[Code Requirements] {prompt} [The Start of Assistant A's Code] {answer_0} [The End of Assistant A's Code] [The Start of Assistant B's Code] {answer_1} [The End of Assistant B's Code]

Table 25: Prompt for refinement preference annotation and evaluation for code generation tasks.

Dialog Generation

Please act as an impartial judge and evaluate the quality of the critiques provided by two AI assistants for my response to the conversation displayed below. Your evaluation should focus on the quality, clarity, and constructiveness of the critiques, particularly the "Suggestions for improvement" field.

You will be given the conversation, my response, assistant A's critique, and assistant B's critique. Your job is to assess which assistant's critique is better based on the following criteria:

1. **Accuracy:** Does the critique accurately identify any issues with my response? Are any mistakes or shortcomings in my response correctly pointed out?

2. **Clarity:** Is the critique clearly written, easy to understand, and well-structured? Does it explain the issues in a way that is accessible and straightforward?

3. **Constructiveness:** Does the critique provide practical and actionable suggestions for improvement? Are the suggestions detailed, specific, and relevant to the issues identified?

4. **Objectivity:** Is the critique unbiased and impartial? Does it focus solely on the quality of my response and avoid unnecessary personal opinions or judgments?

5. **Thoroughness:** Does the critique cover all significant aspects of my response, or does it overlook any important issues? Does it delve into the reasoning behind the suggestions for improvement?

6. **Tone:** Is the critique delivered in a respectful and professional tone, avoiding any condescension or harshness?

You should focus particularly on the "Suggestions for improvement" field in each critique and evaluate how well each assistant has provided guidance to improve the response. Avoid being influenced by the length of the critiques or the order in which they are presented. Do not favor one assistant over the other based on irrelevant factors. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

[USER]

[Conversation] {prompt} [The Start of My Response] {answer} [The End of My Response] [The Start of Assistant A's Critique] {critique_0} [The End of Assistant A's Critique] [The Start of Assistant B's Critique] {critique_1} [The End of Assistant B's Critique]

Table 26: Prompt for critique preference annotation for dialog generation tasks.

Summarization

[SYSTEM]

Please act as an impartial judge and evaluate the quality of the critiques provided by two AI assistants for my summary to the {kind} displayed below. Your evaluation should focus on the quality, clarity, and constructiveness of the critiques, particularly the "Suggestions for improvement" field.

You will be given the {kind}, my summary, assistant A's critique, and assistant B's critique. Your job is to assess which assistant's critique is better based on the following criteria:

1. **Accuracy:** Does the critique accurately identify any issues with my summary? Are any mistakes or shortcomings in my summary correctly pointed out?

2. **Clarity:** Is the critique clearly written, easy to understand, and well-structured? Does it explain the issues in a way that is accessible and straightforward?

3. **Constructiveness:** Does the critique provide practical and actionable suggestions for improvement? Are the suggestions detailed, specific, and relevant to the issues identified?

4. **Objectivity:** Is the critique unbiased and impartial? Does it focus solely on the quality of my summary and avoid unnecessary personal opinions or judgments?

5. **Thoroughness:** Does the critique cover all significant aspects of my summary, or does it overlook any important issues? Does it delve into the reasoning behind the suggestions for improvement?

6. **Tone:** Is the critique delivered in a respectful and professional tone, avoiding any condescension or harshness?

You should focus particularly on the "Suggestions for improvement" field in each critique and evaluate how well each assistant has provided guidance to improve the summary. Avoid being influenced by the length of the critiques or the order in which they are presented. Do not favor one assistant over the other based on irrelevant factors. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

[USER]

[{kind}] {prompt} [The Start of My Summary] {answer} [The End of My Summary] [The Start of Assistant A's Critique] {critique_0} [The End of Assistant A's Critique] [The Start of Assistant B's Critique] {critique_1} [The End of Assistant B's Critique]

Table 27: Prompt for critique preference annotation for summarization tasks. The "kind" field will be "Reddit post" or "News", conditioned to whether this prompt is from TL;DR or CNN-DailyMail dataset.

Question Answering [SYSTEM]

Please act as an impartial judge and evaluate the quality of the critiques provided by two AI assistants for my answer to the question displayed below. Your evaluation should focus on the quality, clarity, and constructiveness of the critiques, particularly the "Suggestions for improvement" field.

You will be given the question, my answer, assistant A's critique, and assistant B's critique. Your job is to assess which assistant's critique is better based on the following criteria:

1. **Accuracy:** Does the critique accurately identify any issues with my answer? Are any mistakes or shortcomings in my answer correctly pointed out?

2. **Clarity:** Is the critique clearly written, easy to understand, and well-structured? Does it explain the issues in a way that is accessible and straightforward?

3. **Constructiveness:** Does the critique provide practical and actionable suggestions for improvement? Are the suggestions detailed, specific, and relevant to the issues identified?

4. **Objectivity:** Is the critique unbiased and impartial? Does it focus solely on the quality of my answer and avoid unnecessary personal opinions or judgments?

5. **Thoroughness:** Does the critique cover all significant aspects of my answer, or does it overlook any important issues? Does it delve into the reasoning behind the suggestions for improvement?

6. ******Tone:****** Is the critique delivered in a respectful and professional tone, avoiding any condescension or harshness?

You should focus particularly on the "Suggestions for improvement" field in each critique and evaluate how well each assistant has provided guidance to improve the answer. Avoid being influenced by the length of the critiques or the order in which they are presented. Do not favor one assistant over the other based on irrelevant factors. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

[USER]

[User Question] {prompt} [The Start of My Answer] {answer} [The End of My Answer] [The Start of Assistant A's Critique] {critique_0} [The End of Assistant A's Critique] [The Start of Assistant B's Critique] {critique_1} [The End of Assistant B's Critique]

Table 28: Prompt for critique preference annotation for question answering tasks.

Mathematical Reasoning [SYSTEM]

"Please act as an impartial judge and evaluate the quality of the critiques provided by two AI assistants for my answer to the math problem displayed below. Your evaluation should focus on the quality, clarity, and constructiveness of the critiques, particularly the "Suggestions for improvement" field.

You will be given the question, my answer, the reference answer, assistant A's critique, and assistant B's critique. Your job is to assess which assistant's critique is better based on the following criteria:

1. **Accuracy:** Does the critique accurately identify any issues with my answer? Are any mistakes or shortcomings in my answer correctly pointed out?

2. **Clarity:** Is the critique clearly written, easy to understand, and well-structured? Does it explain the issues in a way that is accessible and straightforward?

3. **Constructiveness:** Does the critique provide practical and actionable suggestions for improvement? Are the suggestions detailed, specific, and relevant to the issues identified?

4. **Objectivity:** Is the critique unbiased and impartial? Does it focus solely on the quality of my answer and avoid unnecessary personal opinions or judgments?

5. **Thoroughness:** Does the critique cover all significant aspects of my answer, or does it overlook any important issues? Does it delve into the reasoning behind the suggestions for improvement?

6. **Tone:** Is the critique delivered in a respectful and professional tone, avoiding any condescension or harshness?

You should focus particularly on the "Suggestions for improvement" field in each critique and evaluate how well each assistant has provided guidance to improve the answer. Avoid being influenced by the length of the critiques or the order in which they are presented. Do not favor one assistant over the other based on irrelevant factors. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

[USER]

[Math Problem] {prompt} [The Start of Reference Answer] {ref_answer} [The End of Reference Answer] [The Start of My Answer] {answer} [The End of My Answer] [The Start of Assistant A's Critique] {critique_0} [The Start of Assistant A's Critique] [The Start of Assistant B's Critique] {critique_1} [The End of Assistant B's Critique]

Table 29: Prompt for critique preference annotation for mathematical reasoning tasks.

Code Generation [SYSTEM]

Please act as an impartial judge and evaluate the quality of the critiques provided by two AI assistants for my code to the requirements displayed below. Your evaluation should focus on the quality, clarity, and constructiveness of the critiques, particularly the "Suggestions for improvement" field.

You will be given the question, my code, assistant A's critique, and assistant B's critique. Your job is to assess which assistant's critique is better based on the following criteria:

1. **Accuracy:** Does the critique accurately identify any issues with my code? Are any mistakes or shortcomings in my code correctly pointed out?

2. **Clarity:** Is the critique clearly written, easy to understand, and well-structured? Does it explain the issues in a way that is accessible and straightforward?

3. **Constructiveness:** Does the critique provide practical and actionable suggestions for improvement? Are the suggestions detailed, specific, and relevant to the issues identified?

4. **Objectivity:** Is the critique unbiased and impartial? Does it focus solely on the quality of my code and avoid unnecessary personal opinions or judgments?

5. **Thoroughness:** Does the critique cover all significant aspects of my code, or does it overlook any important issues? Does it delve into the reasoning behind the suggestions for improvement?

6. ******Tone:****** Is the critique delivered in a respectful and professional tone, avoiding any condescension or harshness?

You should focus particularly on the "Suggestions for improvement" field in each critique and evaluate how well each assistant has provided guidance to improve the code. Avoid being influenced by the length of the critiques or the order in which they are presented. Do not favor one assistant over the other based on irrelevant factors. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

[USER]

[Code Requirements] {prompt} [The Start of My Code] {answer} [The End of My Code] [The Start of Assistant A's Critique] {critique_0} [The End of Assistant A's Critique] [The Start of Assistant B's Critique] {critique_1} [The End of Assistant B's Critique]

Table 30: Prompt for critique preference annotation for code generation tasks.

Response Quality Scoring: Dialog Generation [SYSTEM]

Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the conversation displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Do not allow the length of the response to influence your evaluation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[USER]

[Conversation] {prompt} [The Start of Assistant's Response] {answer} [The End of Assistant's Response]

Table 31: Prompt for refinement quality scoring for dialog generation tasks.

Response Quality Scoring: Summarization

[SYSTEM]

Please act as an impartial judge and evaluate the quality of the summary provided by an AI assistant to the kind displayed below. Your evaluation should consider whether the summary include all key information from the original article and avoid false or unnecessary sentences. Begin your evaluation by providing a short explanation. Do not allow the length of the summary to influence your evaluation. Be as objective as possible. After providing your explanation, you must rate the summary on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[USER]

 $[\{kind\}]$ {prompt} [The Start of Assistant's Summary] {answer} [The End of Assistant's Summary]

Table 32: Prompt for refinement quality scoring for summarization tasks. The "kind" field will be "Reddit post" or "News", conditioned to whether this prompt is from TL;DR or CNN-DailyMail dataset.

Response Quality Scoring: Question Answering

[SYSTEM]

Please act as an impartial judge and evaluate the quality of the answer provided by an AI assistant to the user question. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail in the answer. Begin your evaluation by providing a short explanation. Do not allow the length of the answer to influence your evaluation. Be as objective as possible. After providing your explanation, you must rate the answer on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[USER] [User Question] {prompt} [The Start of Assistant's Answer] {answer} [The End of Assistant's Answer]

Table 33: Prompt for refinement quality scoring for question answering tasks.

Response Quality Scoring: Mathematical Reasoning [SYSTEM]

Please act as an impartial judge and evaluate the quality of the answer provided by an AI assistant to the math problem. Your evaluation should consider correctness and helpfulness. You will be given a reference answer and the assistant's answer. Begin your evaluation by comparing the assistant's answer with the reference answer. Identify and correct any mistakes. Do not allow the length of the answer to influence your evaluation. Be as objective as possible. After providing your explanation, you must rate the answer on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[USER]

[Math Problem] {prompt} [The Start of Reference Answer] {ref_answer} [The End of Reference Answer] [The Start of Assistant's Answer] {answer} [The End of Assistant's Answer]

Table 34: Prompt for refinement quality scoring for mathematical reasoning tasks.

Response Quality Scoring: Code Generation [SYSTEM]

Please act as an impartial judge and evaluate the quality of the code provided by an AI assistant to the requirements displayed below. Your evaluation should consider correctness and helpfulness. Begin your evaluation by providing a short explanation. Identify and correct any mistakes. Do not allow the length of the response to influence your evaluation. Be as objective as possible. After providing your explanation, you must rate the code on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[USER] [Code Requirements] {prompt} [The Start of Assistant's Code] {answer} [The End of Assistant's Code]

Table 35: Prompt for refinement quality scoring for code generation tasks.



Human: I like to host guests at my home from time to time, and I am gathering recipes of different dishes and drinks to keep things interesting. I am interested in trying some Danish dishes. Can you give me a recipe for Koldskål? Assistant:

Initial Response

Here is a recipe for Koldskål: Ingredients: 4 Quart Jar, 3/4 cup chopped onion 1/2 cup chopped carrot 1/2 cup chopped celery 1 1/2 receips chopped cabage 1/2 cup chopped mushrooms 1/2 cup chopped comatos 1/2 cup chopped catos 1/2 cup chopped labor 1/2 cup chopped process (and the second second

The assistant's response is a mess, to put it mildly. However, it's not entirely harmless, but it's mostly helpful, but not in a good way. However, it's not a recipe for Koldskål, but a list of ingredients that seems to be a repetition of the same ingredients over and over again, with no measurements, no cooking method, and no nutritional information.

The assistant's response is missing some crucial information, such as the fact that Koldskål is a traditional Danish cold sour soup made with buttermilk, eggs, and pickled vegetables, not a salad or a fruit salad. It's also missing the fact that the ingredients are not cooked together, but rather, the pickled vegetables are added to the buttermilk and eggs, not mixed with them.

Suggestions for improvement:

1. Provide a clear and concise recipe for Koldskål, including measurements, cooking method, and nutritional information.

- Use accurate and authentic ingredients and cooking techniques to ensure the recipe is true to the traditional Danish dish.
 Consider adding a brief introduction to the recipe, explaining the history and cultural context of Koldskål in Denmark.
 Consider adding a visual representation of the dish, such as a photo or diagram, to help illustrate the cooking process.

on critique by Llama-3-8B-Ins

Koldskål is a traditional Danish cold sour soup made with buttermilk, eggs, and pickled vegetables. Here's a simple recipe to make this refreshing and tangy soup at home

Ingredients: * 1 cup buttermilk

- * 2 eggs * 1/2 cup pickled red cabbage, chopped
- * 1/4 cup pickled carrots, chopped * Salt and pepper to taste

Cooking Method:

- 1. In a large bowl, whisk together the buttermilk and eggs until smooth.
- Add the chopped pickled cabbage and carrots to the bowl and stir to combine.
 Season with salt and pepper to taste.
 Chill the soup in the refrigerator for at least 30 minutes to allow the flavors to meld.
- 5. Serve the Koldskål cold, garnished with chopped fresh herbs or a sprinkle of paprika.

Critique by Llama-3-8B-Instruct+RCO

Helpfulness: 0/5 - The assistant's response is not helpful at all. It provides a list of ingredients with no instructions on how to prepare the dish, no quantities for the ingredients, and no information on how to serve it.

* 5/5 - The assistant's response does not contain any harmful or offensive content

Correctness: 0/5 - The assistant's response is not correct. Koldskål is a traditional Danish cold sour milk dessert, not a savory dish with a long list of ingredients. The recipe provided is not accurate and does not follow any known recipe for Koldskål.

**Suggestions for improvement:

1. Provide a correct and accurate recipe for Koldskål, including ingredients, quantities, and instructions on how to prepare and serve the dish.

- 2. Consider providing a brief description of the dish and its origins to give the user a better understanding of what they are making.
- Use a clear and concise format for the recipe, making it easy to read and follow.
 Consider providing variations or alternatives to the recipe to cater to different tastes and dietary restrictions

By following these suggestions, the assistant can provide a helpful, harmless, and correct response that will assist the user in making a delicious Danish dish

Refinement based on critique by Llama-3-8B-Instruct+RCO

Thank you for your interest in trying a traditional Danish dish! Koldskål is a cold sour milk dessert that is popular in Denmark. Here is a recipe for Koldskål that you can try:

Ingredients

- 1 cup sour milk
- * 1/2 cup granulated sugar
- * 1/4 cup cornstarch 1/4 teaspoon salt
- * 1/4 teaspoon vanilla extract
- * 1/4 cup chopped fresh fruit (such as strawberries or blueberries)
- 1. In a medium-sized bowl, whisk together the sour milk, sugar, cornstarch, and salt until well combined.
- Add the vanilla extract and mix well.
 Fold in the chopped fresh fruit.
- 4. Cover the bowl with plastic wrap and refrigerate for at least 2 hours or overnight
- Serve the Koldskål chilled, garnished with additional fresh fruit if desired.

I hope you enjoy this recipe for Koldskål! Let me know if you have any questions or if you would like any variations on the recipe

Figure 5: One typical case from the dialog generation task in CriticEval dataset.

Question

[U.S. President Donald Trump shortly after delivering his first State of the Union address Tuesday night. Democrats, and others, see Trump's criticisms of federal law enforcement as a dangerous attempt to undermine special counsel Robert Mueller's Russia probe.] [Passage is omitted]

Answer the following question based on the above passage: Summarize how the Federal Bureau of Investigation and Department of Justice challenged Trump.

Initial Response

They challenged Trump by publicly refuting his criticisms of the FBI and Justice Department.

Critique by Llama-3-8B-Instruct+DPO

The article is a comprehensive analysis of the ongoing controversy between President Trump and the Federal Bureau of Investigation (FBI) and Department of Justice (DOJ). However, it misses some crucial details, such as the specific ways in which they challenged Trump, and it doesn't address the controversy's implications for the country. Here's a breakdown of the article's summary:

* The article mentions that the FBI and DOJ "challenged Trump" but doesn't provide any specific examples of how they did so. However, it's important to note that the FBI and DOJ didn't necessarily challenge Trump directly, but rather, they challenged the accuracy and misleading nature of the Nunes memo, which is a key document in the controversy.
* The article doesn't mention the FBI's deputy director, Andrew McCabe, stepping down under pressure from Trump, which is an

important event that has significant implications for the controversy.

* The article doesn't mention the reported conversation between Trump and Rosenstein about whether Rosenstein is "on my team", which is another important event that raises questions about Trump's interactions with law enforcement.

* The article doesn't mention the reported request by Trump to Rosenstein about supporting the release of the Nunes memo, which is another important event that raises questions about Trump's attempts to influence the investigation.

Suggestions for improvement:

* Provide specific examples of how the FBI and DOJ challenged Trump, such as the FBI's statement about the Nunes memo's material omissions of fact, and the DOJ's objections to the release of the memo.

* Include important events that are missing from the article, such as McCabe's stepping down, the conversation between Trump and Rosenstein, and the request by Trump to Rosenstein about supporting the release of the Nunes memo.

* Consider adding a more detailed analysis of the controversy's implications for the country, such as the potential damage to the FBI and DOJ's credibility, and the potential consequences for the country's institutions.

Refinement based on critique by Llama-3-8B-Instruct+DPO

The controversy between President Trump and the Federal Bureau of Investigation (FBI) and Department of Justice (DOJ) has reached a boiling point. Despite Trump's criticisms of the FBI and DOJ, the agencies have publicly refuted his claims, challenging the accuracy and misleading nature of the Nunes memo. The FBI has stated that the memo contains material omissions of fact, and the DOJ has objected to its release. Specifically, the FBI has expressed concerns about the memo's omission of key facts and its potential to mislead the public. Additionally, the controversy has been marked by important events, such as the stepping down of FBI deputy director Andrew McCabe under pressure from Trump, and the reported conversation between Trump and Rosenstein about whether Rosenstein is "on my team". The controversy raises significant questions about Trump's interactions with law enforcement, including his request to Rosenstein about supporting the release of the Nunes memo.

Critique by Llama-3-8B-Instruct+RCO

The answer is generally faithful to the article, but it could be improved in a few ways.

Firstly, the answer is a bit too brief and doesn't fully capture the extent of the challenge posed by the FBI and Justice Department. The article mentions that the FBI and Justice Department challenged Trump "on the record and through what appeared to be anonymous leaks to CNN", but the answer doesn't elaborate on what this means.

Secondly, the answer could benefit from more specific examples of how the FBI and Justice Department challenged Trump. For instance, the article mentions that the FBI made its objections to the release of the memo public, and that CNN broke two stories that further called into question Trump's interactions with and statements about law enforcement. The answer could include more details like these to make it more comprehensive.

Suggestions for improvement:

* Provide more context about what it means for the FBI and Justice Department to challenge Trump "on the record and through what appeared to be anonymous leaks to CNN".

* Include more specific examples of how the FBI and Justice Department challenged Trump, such as the FBI's public objections to the release of the memo and the CNN stories about Trump's interactions with and statements about law enforcement.

* Consider rephrasing the answer to make it more concise and clearer. For example, instead of saying "They challenged Trump by publicly refuting his criticisms of the FBI and Justice Department", consider saying "The FBI and Justice Department pushed back against Trump's criticisms by publicly refuting his claims and releasing information that contradicted his statements".

Refinement based on critique by Llama-3-8B-Instruct+RCO

The Federal Bureau of Investigation and Department of Justice challenged Trump by publicly refuting his criticisms of the FBI and Justice Department, providing specific examples of how his statements were misleading or inaccurate. For instance, the FBI made its objections to the release of a secret memo public, and CNN broke two stories that further called into question Trump's interactions with and statements about law enforcement. Additionally, the FBI and Justice Department pushed back against Trump's claims of bias and improper use of surveillance by releasing information that contradicted his statements and showed that their investigations were conducted fairly and properly. By providing these specific examples, the FBI and Justice Department challenged Trump's attacks on their institutions and credibility.

Figure 6: One typical case from the summarization task in CriticEval dataset.

Questio

Question: What material is a steel rocking chair made out of? Your answer:

Steel is a structural element used to make chairs.

Critique by Llama-3-8B-Instruct+DPO

The answer is a good start, but it falls short in several areas. Here are some specific issues:

1. Lack of clarity: The answer doesn't address the question directly. It's more of a generalization about the use of steel in construction. This doesn't provide any specific information about the steel rocking chair.

2. No supporting evidence: The answer doesn't provide any evidence or references to support its claim. This lack of credibility can undermine the answer's authority.

3. No depth or nuance: The answer doesn't provide any depth or nuance about the steel used in the rocking chair. It doesn't mention the specific types of steel or techniques used to create the chair.

Suggestions for improvement:

1. Address the question directly: The answer should clearly and explicitly state that it's discussing the material used in a specific object, in this case, a steel rocking chair.

2. Provide specific evidence: The answer should provide references or evidence to support its claims. This could include links to reputable sources or citations to academic studies.

3. Provide depth and nuance: The answer should provide some depth and nuance about the steel used in the rocking chair. This could include specific types of steel used, techniques employed, or processes involved in its creation.

4. Use clear language: The answer should use clear and concise language to describe the material and its properties. This could include specific measurements or technical specifications.

Refinement based on critique by Llama-3-8B-Instruct+DPO

A steel rocking chair is made from a durable and lightweight metal alloy, typically consisting of iron and carbon. The steel is often coated with a rust-resistant finish and may be reinforced with additional materials for added strength. According to a study published in the Journal of Materials Science, the steel used in rocking chairs is typically chosen for its combination of strength, durability, and aesthetic appeal. By addressing these factors, manufacturers can create a comfortable and stylish seating option that meets the needs of users.

Critique by Llama-3-8B-Instruct+RCO

Comment: The answer provided is incomplete and does not fully address the question. The question asks specifically about a steel rocking chair, but the answer only mentions that steel is a structural element used to make chairs in general, without providing any information about the specific type of chair or the materials used in its construction. Additionally, the answer does not provide any additional details about the rocking mechanism or other components of the chair.

Suggestions for improvement:

* Be more specific and clearer in the answer by stating that a steel rocking chair is made of steel, which is used as the frame and structural support, and possibly other materials such as wood, plastic, or fabric for the seat, backrest, and rocking mechanism.
* Provide more context and details about the construction of a steel rocking chair, such as the type of steel used, the manufacturing

process, and any notable features or characteristics.

* Ensure that the answer is accurate and comprehensive, and addresses the specific question being asked.

Refinement based on critique by Llama-3-8B-Instruct+RCO

A steel rocking chair is made out of steel, which serves as the frame and structural support. The seat, backrest, and rocking mechanism may also be made of other materials such as wood, plastic, or fabric. The steel used in the construction of a steel rocking chair is typically strong and durable, allowing it to support the weight of the user and provide a comfortable rocking motion. The manufacturing process involves welding or brazing the steel components together, and may also include additional steps such as cutting, drilling, or shaping the metal to create the desired form and function.

Figure 7: One typical case from the question answering task in CriticEval dataset.



Can you solve the following math problem? According to its nutritional info, a bag of chips has 250 calories per serving. If a 300g bag has 5 servings, how many grams can you eat if your daily calorie target is 2000 and you have already consumed 1800 calories? Explain your reasoning. Your final answer should be a single numerical number, in the form \boxed{answer}, at the end of your response.



 Calculate the number of grams that can be eaten by dividing the remaining calories by the calories per gram, not by multiplying the remaining calories by the grams per calorie.

5. Return the result in grams, not in calories

Figure 8: One typical case from the mathematical reasoning task in CriticEval dataset.



Figure 9: One typical case from the code generation task in CriticBench dataset.