# Training Language Models for Critiquing through Reinforcement Learning

Anonymous ACL submission

#### Abstract

Training critique models to assess and provide feedback on model outputs is a promising way to improve large language models (LLMs) for complex reasoning tasks. However, existing approaches typically rely on stronger supervisors for annotating critique data. To address this, we propose Critique-RL, an online RL framework for developing critique models without stronger supervision. Our framework operates on a two-player paradigm: the actor generates a response, the critic provides feedback, and the actor refines the response accordingly. We first reveal that relying solely on indirect reward signals from the actor's outputs for RL optimization often leads to unsatisfactory critics: while their helpfulness improves, the discriminability remains poor, resulting in marginal performance gains. To overcome this, Critique-RL adopts a two-stage optimization strategy. In stage I, it reinfoces the discriminability of the critic with direct rule-based reward signals; in stage II, it introduces indirect rewards based on actor refinement to improve the critic's helpfulness, while maintaining its discriminability via appropriate regularization. Extensive experiments across different tasks and models demonstrate that Critique-RL achieves significant performance improvements across different models and tasks.<sup>1</sup>

# 1 Introduction

018

037

With the development of large language models (Ouyang et al., 2022; OpenAI, 2023; Touvron et al., 2023; Jiang et al., 2023; Dubey et al., 2024), providing reliable supervision for them has become a critical research challenge (Bowman et al., 2022; Saunders et al., 2022), especially for tasks that are difficult even for humans, such as complex reasoning, sequential decision-making, and coding (Shinn et al., 2023; Snell et al., 2024; Qu et al., 2024; Kumar et al., 2024). This problem is often referred to



Figure 1: Left: Critique-RL achieves better performance and discrimination on MATH. **Right:** Inference compute scaling for Critique-RL, with @2k and @3k indicating sampling amounts that are 2 times and 3 times the x-axis value, respectively. Critique-RL improves the performance ceiling and is more compute-efficient.

as scalable oversight (Bowman et al., 2022). One effective method for scalable oversight is to train critique models to assess and provide feedback to model outputs (Welleck et al., 2023; Akyürek et al., 2023; Xi et al., 2024; Yao et al., 2024). Based on this feedback, actor models can refine and optimize their behavior or outputs.

Existing work in training critique models typically assumes a stronger supervisor to provide labeled critique data, which is often expensive and difficult to scale (Saunders et al., 2022; Xi et al., 2024; Bowman et al., 2022). Moreover, the data labeled by the supervisor often differs significantly from the learner's output distribution (Kumar et al., 2024). Another line of work does not train the model but instead relies on the model's inherent abilities, using prompt engineering to elicit its critiquing abilities (Bai et al., 2022; Madaan et al., 2023; Dhuliawala et al., 2024). However, such methods typically assume an oracle verifier during testing, allowing the critique model to bypass discrimination and focus solely on providing helpfulness feedback for revision (Huang et al., 2024).

In this work, we aim to develop critique models without relying on stronger annotators or an oracle reward function during testing. To this end, we propose Critique-RL, an online RL framework based

068

<sup>&</sup>lt;sup>1</sup>We will release our codes and data for further research.

105

- 107
- 108
- 109
- 110
- 111

112 113

- 114
- 115

116 117 on two-player actor-critic interaction (Yao et al., 2024; Xi et al., 2024) for developing critique models. In our framework, there are two main roles: the actor and critic. The critic assesses (discriminability) and provides feedback (helpfulness) for the actor's output, and the actor performs refinement accordingly (Saunders et al., 2022).

To build our method, we first use the correctness of the actor's two attempts as the reward signals for the RL optimization of critique models (Section 4.1), as intuitively, these indirect signals reflect the quality of critiques (Yao et al., 2024; Zheng et al., 2024). However, this approach fails to develop satisfactory critique models, i.e., with low performance. Delving into the optimization process, we reveal that while the helpfulness of the critique models improves, their discriminability is not well optimized, leading to an optimization bottleneck and even a collapse of RL training.

To address the challenges, Critique-RL employes a two-stage RL process (Section 4.2). Specifically, as shown in Figure 2, in the first stage, we optimize the discriminability of the critique models using direct rule-based reward signals. In the second stage, we introduce indirect rewards based on the correctness of actor refinement to enhance the helpfulness, while using appropriate regularization to maintain their discriminability. Indepth training dynamics shows that our method addresses the training collapse and stably optimizes both discriminability and helpfulness. Extensive experiments show that our method outperforms baselines across different models and tasks. It is also noteworthy that critique models trained with our method can generalize to unseen tasks, demonstrating its promise for scalable oversight.

In summary, our main contributions are:

- 1. Delving into the RL optimization process, we reveal that solely depending on indirect reward signals of actor's output correctness cannot develop satisfactory critique models, as their discriminability is not well optimized.
- We then propose Critique-RL, a new online RL framework to develop critique models for provide accurate assessment and helpful feedback for model outputs.
- 3. We perform in-depth experiments, ablation and analysis to show the effectiveness and stability of our method. We hope our work

provides insights for future development of scalable oversight in the community.

118

119

120

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

# 2 Related Work

Prompt engineering for eliciting critiquing ability from language models. As a key technique for scalable oversight (Bowman et al., 2022), many previous works have explored the use of prompt engineering to elicit the critiquing and reflection abilities of LLMs (Bai et al., 2022; Madaan et al., 2023; Ye et al., 2023; Dhuliawala et al., 2024). These methods typically rely on an oracle verifier at test time for discrimination, allowing the LM to focus solely on providing natural language feedback (Huang et al., 2024). However, in the absence of an external verifier, even SOTA models face significant challenges (Saunders et al., 2022; Welleck et al., 2023; Xu et al., 2024; Huang et al., 2024). In this work, we do not assume a oracle verifier; instead, we train critique models through RL to optimize both discriminability and the ability to provide helpful feedback.

# Fine-tuning language models for critiquing.

Previously, a line of work has explored fine-tuningbased approaches for training critique models (Saunders et al., 2022; Bowman et al., 2022; Xi et al., 2024). However, these methods primarily rely on a stronger supervisor for data annotation, which is costly and difficult to scale (Xi et al., 2024). To address this issue, some researchers have proposed self-improvement-based methods to train models for self-critiquing (Tang et al., 2025; Zheng et al., 2024; Yuan et al., 2025). Unlike these approaches, we adopt a two-player paradigm and train a separated critique model through RL.

**Reinforcement learning for language models.** RL has become an essential component of LLM post-training, such as RLHF for alignment (Ouyang et al., 2022; Zheng et al., 2023; Wang et al., 2024; Shao et al., 2024). Additionally, various works have leveraged RL to enhance language models' performance in reasoning (Snell et al., 2024; Kumar et al., 2024), coding (Kumar et al., 2024), and decision-making tasks (Shinn et al., 2023). Furthermore, some studies explores using RL to improve LM's ability for self-reflection and self-correction (McAleese et al., 2024; Kumar et al., 2024; Welleck et al., 2023; Shinn et al., 2023; Xu et al., 2024; Ye et al., 2023). The most relevant work to ours is ReTroformer, which focuses



Figure 2: Left: A case illustrating the two-player actor-critic interaction, including the original response from the actor, the critique from the critic, and the refinement from the Actor. **Right:** Overview of our method and its comparison with baseline RL. The snowflake icon <sup>3</sup>/<sub>4</sub> on the Actor indicates that it is fixed, while the fire icon <sup>4</sup>/<sub>4</sub> on the Critic indicates that it will be updated. Our method employs a two-stage RL process. It optimize discriminability of critique models in Stage I, and optimize helpfulness while maintaining discriminability in Stage II.

on decision-making tasks, and leverages indirect reward signals to optimize critique model's helpfulness (Yao et al., 2024). Different from them, we propose a two-stage Critique-RL framework to optimize both discriminability and helpfulness, effectively developing critique models.

#### **3** Preliminaries

167

170

171

172

174

175

176

177

178

181

184

188

#### 3.1 The Two-Player Interaction Framework

The multi-agent framework in this work consists of two main roles (Yao et al., 2024; Xi et al., 2024): the actor model and the critique model. It operates through a response-critique-refinement process.

Specifically, given a question x, the actor model is expected to generate an original response  $y = \pi_{\theta}(x)$ , which includes both the reasoning trajectory and the final answer. The correctness verifier then provides an oracle reward  $r_{\text{oracle}}(x, y)$  to the actor model. Subsequently, the critique model  $\pi_{\phi}$  takes the question-response pair (x, y) as input and produces critique  $c = \pi_{\phi}(x, y)$ , which should include assessment of the response correctness (discriminability) and offer constructive natural language feedback (helpfulness). Based on this critique, the actor model generates a refinement response  $y' = \pi_{\theta}(x, y, c)$ , and subsequently receives an oracle reward  $r_{\text{oracle}}(x, y')$ . Using these rewards, i.e.,  $r_{\text{oracle}}(x, y)$  and  $r_{\text{oracle}}(x, y')$ , we can design different reward functions  $r_{c}(\cdot)$  for critique models, which will be shown in Section 4. 189

190

191

192

193

196

197

198

200

201

202

205

209

# 3.2 Policy Gradient for LLMs

Policy gradient methods (Sutton et al., 1999), e.g., REINFORCE (Ahmadian et al., 2024; Kumar et al., 2024), are common techniques to perform RL on LLMs. For the policy critique model  $\pi_{\phi}$  parameterized by  $\phi$ , the objective of policy gradient is to find an optimal policy that maximizes the reward function  $r_{\rm c}(\cdot)$ . It is typically expressed as maximizing:

$$\mathbb{E}_{c \sim \pi_{\phi}(\cdot|x,y), y' \sim \pi_{\theta}(x,y,c)}[r_{\mathsf{c}}(x,y,c,y')], \quad (1)$$

where  $\mathbb{E}_{c \sim \pi_{\phi}(\cdot|x,y), y' \sim \pi_{\theta}(x,y,c)}$  denotes the expectation over the critique sampled from the critic  $\pi_{\phi}$ and the refinement response sampled from the actor  $\pi_{\theta}$ . Using this gradient, we can perform gradient

212

214

215

216

217

218

219

221

226

227

228

236

237

240

241

242

243

245

246

247

253

254

ascent to optimize the critique model. When the critique is positive, it is "reinforced" by increasing its probability.

#### 3.3 Evaluating Metrics

To evaluate the performance of the critique model, we consider the following metrics: (1) Acc@Refine: the accuracy of the actor model's refinement response; (2)  $\Delta$ : the improvement in the actor model's accuracy between the original and refinement response, which measures the effectiveness of the critique model; (3)  $\Delta^{c \rightarrow i}$ : the change rate from an originally correct response to an incorrect refinement response. A lower value is better; (4)  $\Delta^{i \rightarrow c}$ : the change rate from an originally incorrect response to a correct refinement response. A higher value is better; (5) Acc@Dis: a direct metric to measure the discriminability of the critique model, which quantifies the accuracy of whether the correctness accessed by the critic aligns with the true correctness of the original response.

#### Methodology 4

#### **Motivating Findings: RL with Indirect** 4.1 **Reward Signals Is Insufficient for** Training Satisfactory Critique Models

In the two-player actor-critic framework (Yao et al., 2024; Xi et al., 2024), a natural and intuitive way to optimize the critique models is to shape the reward signals derived from the actor's two attempts (original and refinement responses). We explore several reward shaping approaches, demonstrate their failure modes, and investigate why they fail to incentivize satisfactory critiquing ability.

Analysis setups: data, models, and training 244 methods. Our preliminary experiments are on GSM8K (Cobbe et al., 2021), and the backbone model is Qwen-2.5-3B (Team, 2024). Following previous work (Xi et al., 2024), we train an actor model capable of generating responses and faithfully refining them according to critiques. To build the SFT dataset for initializing a base critique model, we prompted the backbone model to obtain critique data  $\mathcal{D}_{\text{SFT}} = \{x, y, c\}_{i=1}^{|\mathcal{D}_{\text{SFT}}|}$ , rather than using annotations from SOTA commercial models like GPT-40 (OpenAI, 2023). We filter the critique data based on the correctness of refinement to ensure the quality.

Next, we train the critique model  $\pi_{\phi}$  using the

SFT loss:

$$\mathcal{L}_{\text{SFT}}(\phi) = \mathbb{E}_{(x,y,c)\sim\mathcal{D}_{\text{SFT}}} \Big[\log \pi_{\phi}(c|x,y)\Big].$$
(2)

We then employ policy gradient (Sutton et al., 1999) to maximize:

$$\mathbb{E}_{c \sim \pi_{\phi}^{\mathsf{RL}}(\cdot|x,y), y' \sim \pi_{\theta}(\cdot|x,y,c)} \left[ r_{c}(x,y,c,y') - \beta \mathsf{KL}(\pi_{\phi}^{\mathsf{SFT}}(c|x,y) || \pi_{\phi}^{\mathsf{RL}}(c|x,y)) \right],$$
(3)

where  $\pi_{\theta}$  is the fixed actor model,  $\pi_{\phi}^{\text{SFT}}$  is the SFT model. Each x is a query sampled from the RL dataset  $\mathcal{D}_{RL}$ , y is the original response.  $KL(\cdot || \cdot)$ means the KL-divergence which constrains the distance between the RL model and the SFT model, and  $\beta$  is a scaling factor.  $r_{\rm c}(\cdot)$  is the reward function for critique models. Here, with  $r_{\text{oracle}}$  being the oracle reward function that verifies the correctness of an actor response,  $r_{\rm c}(\cdot)$  can be  $r_{\rm refine}$  which represents the correctness of the refinement:

$$r_{\text{refine}}(x, y, c, y') = r_{\text{oracle}}(x, y'), \qquad (4)$$

or it can be  $r_{\Delta}$  which represents the difference in correctness between the actor's two attempts:

$$r_{\Delta}(x, y, c, y') = r_{\text{oracle}}(x, y') - r_{\text{oracle}}(x, y).$$
(5)

Moreover, we also include  $r_{\text{correction}}$  as  $r_{\text{c}}(\cdot)$  for reinforcing the ability in correcting incorrect responses:

$$r_{\text{correction}}(x, y, c, y') = \begin{cases} 1.0, r_{\text{oracle}}(x, y) = 0 \text{ and } r_{\text{oracle}}(x, y') = 1, \\ 0.2, r_{\text{oracle}}(x, y) = 1 \text{ and } r_{\text{oracle}}(x, y') = 1, \\ 0.0, r_{\text{oracle}}(x, y') = 0. \end{cases}$$
(6)

**Empirical findings and behavior analysis.** We illustrate the training dynamics during RL in Figure 3. Optimizing with  $r_{\text{refine}}$  and  $r_{\Delta}$  can reduce  $\Delta^{c \to i}$ , preventing originally correct responses from being altered incorrectly, but its  $\Delta^{i \rightarrow c}$  is not significantly optimized, meaning its error correction performance is not good enough. This phenomenon reveals that the critique model is overly conservative, encouraging the actor to not change its answers. As a result, the final Acc@Refine is not satisfactory.

In contrast, optimizing with  $r_{\text{correction}}$  improves  $\Delta^{i \to c}$ , but fails to effectively reduce  $\Delta^{c \to i}$ . This means it often provides more aggressive suggestions, encouraging the actor model to correct incorrect responses, but it also introduces a greater risk

261

263

264

265

267

268

270

271

272

273

274

275

276

277

278

279

281

282

283

284

285

286

287

288

290

291

292

293

294

295



Figure 3: Training dynamics of preliminary experiments. "Acc@Dis Originally Correct" and "Acc@Dis Originally Incorrect" refer to the discrimination accuracy of originally correct and incorrect responses, respectively. Baselines using indirect reward signals to optimize helpfulness tend to exhibit overly conservative or aggressive behavior as the discriminability is not well optimized. In contrast, our Critique-RL optimizes discriminability in Stage I, and optimizes helpfulness while maintaining discriminability in Stage II. As a result, Critique-RL performs better in Acc@Refine,  $\Delta^{c \rightarrow i}$  and  $\Delta^{i \rightarrow c}$ .

of turning originally correct answers into incorrect ones. Similarly, the final Acc@Refine is also not satisfactory.

297

298

299

303

311

312

Analyzing underlying reasons for the failure modes. To reveal the reasons behind the above failure modes, we also visualize the discrimination performance of the critique models during RL in Figure 3. We find that as RL progresses, all three reward functions  $r_{refine}$ ,  $r_{\Delta}$  and  $r_{correction}$  fail to optimize discriminability effectively. For originally correct and incorrect responses, they can only optimize the judgment for one, while the ability to judge the other is reduced. This may be because both of the indirect reward functions are based on the actor's responses, targeting helpfulness and overlooking discriminability. This motivates the proposal of our method.

#### 4.2 Two-Stage Critique-RL

**Key challenges.** Based on the previous analysis, we have identified two key challenges in RL for critique models: (1) optimizing the discriminability of critique models to improve their accuracy in judging both correct and incorrect original responses; (2) improving the quality of the model's feedback, i.e., helpfulness, while maintaining its discriminability, to prevent the issues of being overly aggressive or overly conservative. **Method overview.** To address the above challenges, we propose the two-stage Critique-RL. In the first stage, our method explicitly optimizes the discriminability of the critique models using direct reward signals. We then use the model optimized by the first stage  $\pi_{\phi}^{\text{Stage-II}}$  as the initialization for the second stage. In the second stage, we introduce a reward function based on the actor's response to optimize critic's helpfulness, while also incorporating appropriate regularization to maintain its discriminability. We illustrate our method in Figure 2 and the algorithm is summarized in Algorithm 1.

324

326

328

329

330

331

332

333

334

335

337

338

339

341

343

344

345

346

349

Stage I: optimizating discriminability through direct reward signals. We decouple the discriminability and helpfulness of the critique models (Saunders et al., 2022). In Stage I, we shape the reward based solely on the actor's original response. Given (x, y), critique models are prompted to give correctness judgments for each step, and also provide a judgment for the final answer. Based on this, we define the discriminability reward function of the critique models as:

$$r_{\rm dis}(x, y, c) = \mathbb{1}\Big(f(x, y, c) = r_{\rm oracle}(x, y)\Big), \quad (7)$$

where f(x, y, c) is the critique model's judgment of the correctness of the original response.  $\mathbb{1}(\cdot)$  is indicator function that returns 1 only when the con-

Model	Method	MATH		GSM8K			AQuA			
		Acc	$\Delta$	Acc@Dis	Acc	$\Delta$	Acc@Dis	Acc	$\Delta$	Acc@Dis
Qwen2.5-3B	No Critic	36.90	_	_	66.03	_	_	50.00	_	_
	SFT	44.24	7.34	66.51	69.14	3.11	76.34	46.46	-3.54	61.97
	STaR	44.38	7.48	66.97	71.95	5.91	74.79	50.39	0.39	66.13
	Base-PPO	44.54	7.64	65.11	70.51	4.47	77.59	51.18	1.18	58.44
	Base-RLOO	46.14	9.24	69.29	70.58	4.55	76.71	53.54	3.54	62.20
	Critique-RL	$\underline{48.60}$	$\underline{11.70}$	$\underline{82.80}$	$\underline{75.89}$	<u>9.86</u>	87.44	<u>56.69</u>	<u>6.69</u>	<u>69.92</u>
Qwen2.5-7B	No Critic	45.74	_	_	75.66	_	_	63.39	_	_
	SFT	51.84	6.10	67.59	78.77	3.11	79.42	59.45	-3.94	68.67
	STaR	54.06	8.32	69.71	80.52	4.85	81.03	57.87	-5.51	$\underline{72.18}$
	Base-PPO	52.34	6.60	68.03	80.82	5.16	77.05	63.39	0.00	70.56
	Base-RLOO	53.86	8.12	$\underline{71.42}$	81.35	5.69	83.44	64.96	1.57	71.66
	Critique-RL	$\underline{58.40}$	12.66	85.20	87.72	$\underline{12.05}$	<u>90.43</u>	65.75	2.36	$\underline{78.09}$

Table 1: Main results. The best performance is in **bold** and <u>underlined</u>, while the second-best performance is <u>underlined</u>. Our method is marked in <u>blue</u>. No Critic means the actor model perform reasoning only, and we report the reasoning performance. For other methods, we report the Acc@Refine performance for the acc column.

dition inside the parentheses holds, and 0 otherwise. Based on this, our Stage I RL maximizes:

$$\mathbb{E}_{c \sim \pi_{\phi}^{\text{Stage-I}}(\cdot|x,y)} \Big[ r_{\text{dis}}(x,y,c) \\
-\beta \text{KL}(\pi_{\phi}^{\text{SFT}}(c|x,y) || \pi_{\phi}^{\text{Stage-I}}(c|x,y)) \Big],$$
(8)

where the KL divergence with the SFT model is still used to stabilize the training. As shown in Figure 3, our Stage I RL can effectively and stably optimize discriminability, regardless of the correctness of the original response.

Stage II: optimizating helpfulness while maintaining discriminability. The goal of the second stage of Critique-RL is to optimize the helpfulness of the critique models without sacrificing their discriminability, thereby avoiding overly conservative or overly aggressive behavior patterns. To achieve this, we introduce a reward function based on actor refinement correctness,  $r_{refine}$ . Meanwhile, to preserve the model's discriminability, we retain  $r_{dis}$  and introduce a regularization term based on the KL divergence with the Stage I model  $\pi_{\phi}^{Stage-I}$ . Specifically, we maximize the following objective:

$$\mathbb{E}_{c \sim \pi_{\phi}^{\text{Stage-II}}(\cdot|x,y), y' \sim \pi_{\theta}(\cdot|x,y,c)} \Big[ r_{\text{refine}} + \beta_1 r_{\text{dis}}(x,y,c) - \beta_2 \text{KL}(\pi_{\phi}^{\text{Stage-II}}(c|x,y)) \| \pi_{\phi}^{\text{Stage-II}}(c|x,y)) \Big],$$
(9)

where  $\beta_1$  and  $\beta_2$  are scaling factors. As shown in Figure 3, our Stage II effectively optimizes the model's helpfulness, increasing  $\Delta^{i \to c}$  and decreasing  $\Delta^{c \to c}$ , ultimately leading to a stable improvement in Acc@Refine and  $\Delta$ . On the test set, our method also achieves excellent performance (see Section 5).

# 5 Experiments

#### 5.1 Experimental Setup

**Datasets.** Focusing on mathematical reasoning tasks, we select 5 different commonly-used tasks, including free-from and multiple-choice. Following Ding et al. (2024), we construct training set with the train-split of MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), AQUA (Ling et al., 2017). The testset of the three tasks are used as in-domain testset, while the test-split of SVAMP (Patel et al., 2021), TheoremQA (Chen et al., 2023), are used as our OOD (out-of-domain) testset.

**Models and baselines.** Our experiments are conducted on Qwen2.5 series (Team, 2024), i.e., Qwen2.5-3B and Qwen2.5-7B. We include several baselines: (1) SFT which fine-tunes models with high-quality critique data. (2) STaR which iteratively fine-tunes critique models on data generated by themselves and filtered based on the refinement correctness of the actor. (3) RL with  $r_{\text{refine}}$  for optimizing helpfulness of critique models. We include two widely used algorithms PPO and RLOO for this baseline.

Implementation details.All experiments are401conducted on 8 NVIDIA A800 GPUs. To initialize402an actor that can reason and refine based on the403critiquing feedback, we follow Ding et al. (2024);404Xi et al. (2024) to construct a dataset of 21,973405reasoning traces and 12,000 refinement responses.406

353

354

357

365

367

- 376 377

378

379

380

381

382

383

384

385

389

390

391

392

393

394

395

397

398

399

For critique data, we construct a set of 6,000 ex-407 amples, with 2,000 examples in each training task. 408 For fine-tuning actors, we set the epoch to 3 and 409 learning rate to 5e - 6; for fine-tuning critics, we 410 set the epoch to 5 and learning rate to 5e - 6. For 411 STaR and RL, we perfrom SFT to obtain an ini-412 tialized model. In all RL methods, we set the KL 413 coefficient to 0.01. In Critique-RL, we use RLOO 414 as our base algorithm as it performs well and does 415 not require a value model. We train the critique 416 model for 500 steps at each stage and report best 417 results. During evaluation, the temperature is set 418 to 0. For inference-compute scaling analysis or 419 Pass@K, we set temperature to 0.7. 420

#### 5.2 Main Results

421

422

423

424

425

426

427

428

429

430

431

432

Generally, critique models can significantly improve actor's reasoning performance. The results in Table 1 demonstrate that when introducing critique models, the actor's reasoning performance can be boosted by a large margin. For example, in the MATH task, even the SFT Baseline outperforms the model without a critic by 7.34 and 6.10 points on the 3B and 7B models, respectively. This suggests that critique models are an effective supervision method, as discussed in Saunders et al. (2022); McAleese et al. (2024).

**RL**-based methods outperforms fine-tuning-433 based. Both SFT and STaR methods lead to 434 promising critique models, but in most cases, on-435 line RL-based methods perform better, especially 436 our Critique-RL. For instance, on the 3B model, 437 our method surpasses the SFT method by an aver-438 age of 7.11 points on accuracy across three datasets. 439 It is worth noting that on AQuA, fine-tuning-based 440 SFT and STaR may lead to negative impact on per-441 442 formance, while our method provides significant positive improvements. This reveals that online RL 443 methods have greater potential and adaptability in 444 eliciting the model's critiquing ability, similar to 445 the findings in (McAleese et al., 2024). 446

Critique-RL consistently outperforms other 447 baselines in discrimination and final accuracy. 448 In terms of discrimination, our method also signifi-449 cantly outperforms other baselines, such as surpass-450 ing Base-RLOO by 5.31, 6.36 points for 3B and 451 452 7B models on GSM8K, respectively. This reveals that our discrimination-related reward shaping can 453 effectively optimizes discriminability. Thanks to 454 this and the helpfulness reward design in the second 455 stage, our method shows a significant improvement 456

Method	MAT	Ή	AQuA		
	Acc@Refine	Acc@Dis	Acc@Refine	Acc@Dis	
Critique-RL (Ours)	48.60	82.80	56.69	<u>69.92</u>	
-w/o Stage I	47.62	79.71	53.94	66.53	
-w/o Stage II	45.90	78.68	54.72	68.22	
-Stage II w/o discrimination	47.32	77.66	53.54	61.56	
-Stage II w/ $r_{\Delta}$	48.16	<u>82.57</u>	53.94	68.44	
-Stage II w/ r <sub>correction</sub>	47.74	81.96	54.72	68.39	

Table 2: Ablation study using Qwen2.5-3B. We report the Acc@Refine. "w/o" means without; "Stage II w/o discrimination" means in Stage II, we remove  $r_{\rm dis}$  and  ${\rm KL}(\pi_{\phi}^{\rm Stage-I}||\pi_{\phi}^{\rm Stage-II})$ ; "Stage II w/  $r_{\Delta}$ " and "Stage II w/  $r_{\rm correction}$ " mean replacing the  $r_{\rm refine}$  with the corresponding reward function.

in final performance compared to other baselines. For example, on the 7B model, our method outperforms Base-PPO by an average of 5.11 and 12.69 points on accuracy and discriminability, across three datasets. 457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

# 6 Discussion and Analysis

#### 6.1 Ablation Study

We conduct ablation experiments to validate the importance of different components in our method. The experimental results are shown in Table 2.

Ablation on different stages. Both Stage I and Stage II are crucial, and removing either of them leads to a decrease in performance. This indicates that optimizing both discriminability and helpfulness is essential in developing critique models.

Ablation on reward design for Stage II. Next, we perform a deeper analysis of the reward design in Stage II. First, if we remove the discrimination-related  $r_{dis}$  and KL-based regularization  $\text{KL}(\pi_{\phi}^{\text{Stage-I}} || \pi_{\phi}^{\text{Stage-II}})$ , the discriminability and accuracy suffer a significant drop. This further emphasizes that when optimizing for helpfulness, it is crucial to maintain the model's discrimination ability. Second, when we replace the reward function  $r_{\text{refine}}$  in Stage II with another reward function, i.e.,  $r_{\Delta}$  and  $r_{\text{correction}}$ , we observe a slight performance drop. This may be because  $r_{\text{refine}}$  directly optimizes the Acc@Refine metric, which aligns most closely with the actual test-time scenario.

# 6.2 Analyzing Helpfulness When the Oracle Verifier Is Available

Many previous works have relied on an external oracle verifier to assess the actor's reasoning results (Bai et al., 2022; Madaan et al., 2023; Ye et al., 2023; Dhuliawala et al., 2024). In this scenario, the model's judgment ability is isolated, allowing us



Figure 4: Performance with and without the oracle verifier. When the oracle verifier is available, the model no longer needs to make discriminations and just needs to provides useful feedback. This allows us to evaluate the model's helpfulness more accurately.

to better evaluate the critique model's helpfulness. 493 We conducted relevant experiments, and the results 494 are shown in Figure 4. We found that when the oracle verifier is available, all baselines show perfor-496 mance improvements. In this case, our method still 497 outperforms others across different datasets and 498 models, indicating that our approach significantly 499 enhances the model's helpfulness. Furthermore, comparisons with other RL baselines reveal that the optimization of discriminability in our method 502 503 also implicitly contributes to the improvement of helpfulness, suggesting that the two abilities are 504 not entirely independent. This further emphasizes the importance of optimizing both abilities jointly in developing critique models.

# 6.3 Scaling Test-Time Inference Compute for Critique-RL

509

511

512

513

514

516

517

520

521

522

524

We investigate whether Critique-RL can be combined with inference-time compute scaling strategy. Following Qu et al. (2024); Snell et al. (2024); Xi et al. (2024), we leverage the commonly used majority vote (MV@K) (Wang et al., 2023) which evaluates whether the most frequent answer among K samples is correct. The results of MATH are shown in Figure 1 and the results of GSM8K are shown in Figure 5 of Appendix B. Compared to not using critique models, Critique-RL significantly increases the performance ceiling and shows a more sustained upward trend as inference compute scales. More importantly, performing  $K \times$  response-critique-refinement sampling is more compute-efficient than conducting  $3K \times$  par-

Model	Method	SV	AMP	TheoremQA		
		Acc	Pass@10	Acc	Pass@10	
Qwen2.5-3B	No Critic	70.67	92.00	15.13	34.75	
	Critique-RL	78.33	96.33	16.75	37.75	
Owen2 5 7P	No Critic	80.33	95.67	19.38	39.75	
Qwell2.3-7B	Critique-RL	89.67	97.00	21.38	43.00	

Table 3: Out-of-domain evaluation of Critique-RL.

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

552

553

554

555

556

558

559

560

561

562

563

564

565

566

allel sampling responses.

#### 6.4 Generalization to OOD Tasks

We also validated the generalization of the models trained by Critique-RL on OOD tasks. The results in Table 3 show that the models trained still delivers significant performance improvements, further demonstrating the potential of this scalable oversight approach.

## 6.5 Qualitative Analysis of Critique-RL

We perform a qualitative investigation into how Critique-RL works and provide several examples in Appendix C. In Figure 6, facing the originally incorrect response, the critique model after SFT is unable to detect errors, leading the actor's refinement response to retain the same errors. However, the model trained after Critique-RL identifies the errors in the original response and provides detailed, constructive suggestions for modification, leading to the correct refinement response. In Figure 7, model trained after Critique-RL Stage I is able to detect errors, demonstrating its discriminability. However, the model provides the actor with low-quality suggestion, causing the actor's refinement response to be incorrect. In contrast, for the same erroneous original response, model trained after Critique-RL Stage II not only detects the error but also offers a constructive suggestion, ultimately leading to the correct refinement response, demonstrating the advantage of two-stage RL process.

# 7 Conclusion

In this paper, we propose Critique-RL, an RL framework for developing critique models without the need for additional labeled data. Through in-depth analysis, we highlight the importance of explicitly optimizing model discriminability and propose a two-stage RL approach that effectively optimizes both discriminability and helpfulness. We validate its stability and superiority through detailed experiments, and further uncover its working mechanism through ablation studies and analyses. We hope that our work can provide insights for the scalable oversight community of large models.

672

673

674

675

676

677

678

679

680

621

622

623

# Limitations

567

585

586

587

589

590

594

595

596

597

598

609

610

611

612 613

614

615

616

617

618

619

620

In this paper, we propose an RL method for training critique models and validate its effectiveness 569 and stability through detailed experiments, abla-570 tion studies, and analyses. However, there are still 571 some limitations that need to be addressed in future work. First, our main analysis focuses on mathe-573 matical reasoning tasks, and in the future, it should 574 be extended to more tasks to test its generaliza-575 tion ability. Second, our method is primarily based 576 on the Qwen2.5 series of models, and future work 577 should explore its applicability to a broader range of models. Third, our approach relies on an ex-579 plicit two-stage training process, which increases manual effort and reduces flexibility. Future re-581 search should explore how to integrate these two 582 stages and train stronger critique models, making the approach more scalable. 584

## References

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 12248–12267. Association for Computational Linguistics.
- Afra Feyza Akyürek, Ekin Akyürek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. RL4F: generating natural language feedback with reinforcement learning for repairing model outputs. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 7716–7733. Association for Computational Linguistics.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.
- Samuel R. Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosiute, Amanda Askell, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Christopher Olah, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Jackson Kernion, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Liane Lovitt, Nelson Elhage, Nicholas Schiefer, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Robin

Larson, Sam McCandlish, Sandipan Kundu, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, and Jared Kaplan. 2022. Measuring progress on scalable oversight for large language models. *CoRR*, abs/2211.03540.

- Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023. Theoremqa: A theorem-driven question answering dataset. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 7889–7901. Association for Computational Linguistics.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.
- Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of* the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024, pages 3563–3578. Association for Computational Linguistics.
- Yiwen Ding, Zhiheng Xi, Wei He, Zhuoyuan Li, Yitao Zhai, Xiaowei Shi, Xunliang Cai, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Mitigating tail narrowing in llm self-improvement via socratic-guided sampling. *arXiv preprint arXiv:2411.00750*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi,

787

788

789

790

791

792

793

Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. CoRR, abs/2407.21783.

681

688

701

702

705

707

710

711

712

713

715

716

718

719

720

721

722

723

724

726

727

728

730

731

732

733

734

735

736

737

738

- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual.
- Xinyun Chen, Jie Huang, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengvel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. CoRR, abs/2310.06825.
- Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal M. P. Behbahani, and Aleksandra Faust. 2024. Training language models to self-correct via reinforcement learning. CoRR, abs/2409.12917.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pages 158-167. Association for Computational Linguistics.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.
- 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.

- OpenAI. 2023. GPT-4 technical report. CoRR, abs/2303.08774.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In Proceedings of the 2021 *Conference of the North American Chapter of the* Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 2080–2094. Association for Computational Linguistics.
- Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. CoRR, abs/2407.18219.
- William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. CoRR, abs/2206.05802.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. CoRR, abs/2402.03300.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. CoRR, abs/2408.03314.
- Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999], pages 1057-1063. The MIT Press.

888

889

890

891

892

893

853

854

- 794 795 796
- 798
- 79

810

811

812

813

816

818

819

820

821

822

824

825

826

832

834

835

836

837 838

841

842

844

847

848

852

- Zhengyang Tang, Ziniu Li, Zhenyang Xiao, Tian Ding, Ruoyu Sun, Benyou Wang, Dayiheng Liu, Fei Huang, Tianyu Liu, Bowen Yu, et al. 2025. Enabling scalable oversight via self-evolving critic. *arXiv preprint arXiv:2501.05727*.
- 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, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and finetuned chat models. CoRR, abs/2307.09288.
  - Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, Songyang Gao, Nuo Xu, Yuhao Zhou, Xiaoran Fan, Zhiheng Xi, Jun Zhao, Xiao Wang, Tao Ji, Hang Yan, Lixing Shen, Zhan Chen, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2024. Secrets of RLHF in large language models part II: reward modeling. *CoRR*, abs/2401.06080.
  - Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference* on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.
  - Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023.* OpenReview.net.
  - Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Dou, Wenyu Zhan, Xiao Wang, Rui Zheng, Tao Ji, Xiaowei Shi, Yitao Zhai, Rongxiang Weng, Jingang Wang, Xunliang Cai, Tao Gui, Zuxuan Wu, Qi Zhang, Xipeng Qiu, Xuanjing Huang, and Yu-Gang Jiang. 2024. Enhancing LLM reasoning via

critique models with test-time and training-time supervision. *CoRR*, abs/2411.16579.

- Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and prejudice: LLM amplifies self-bias in self-refinement. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 15474–15492. Association for Computational Linguistics.
- Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh R. N., Zeyuan Chen, Jianguo Zhang, Devansh Arpit, Ran Xu, Phil Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. 2024. Retroformer: Retrospective large language agents with policy gradient optimization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May* 7-11, 2024. OpenReview.net.
- Seonghyeon Ye, Yongrae Jo, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, and Minjoon Seo. 2023. Selfee: Iterative self-revising llm empowered by selffeedback generation. Blog post.
- Siyu Yuan, Zehui Chen, Zhiheng Xi, Junjie Ye, Zhengyin Du, and Jiecao Chen. 2025. Agent-r: Training language model agents to reflect via iterative self-training. *arXiv preprint arXiv:2501.11425*.
- Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. 2023. Secrets of RLHF in large language models part I: PPO. *CoRR*, abs/2307.04964.
- Xin Zheng, Jie Lou, Boxi Cao, Xueru Wen, Yuqiu Ji, Hongyu Lin, Yaojie Lu, Xianpei Han, Debing Zhang, and Le Sun. 2024. Critic-cot: Boosting the reasoning abilities of large language model via chain-ofthoughts critic. *Preprint*, arXiv:2408.16326.



Figure 5: Inference compute scaling for Critique-RL, with @2k and @3k indicating sampling amounts that are 2 times and 3 times the x-axis value, respectively. Critique-RL improves the performance ceiling and is more compute-efficient.

# A Algorithm of Critique-RL

Our main algorithm is summarized in Algorithm 1.

#### **B** More Test-time Scaling Results

The results of test-time scaling on GSM8K are illustrated in Figure 5. Similar to the findings on MATH, Critique-RL is more compute-efficient and significantly increases the performance ceiling, validating the potential of our approach.

#### C Examples of Qualitative Analysis

The examples of our qualitative analysis are in Figure 6 and Figure 7.

From Figure 6, we find the model trained after Critique-RL shows strong discriminability and helpfulness compared to the model after SFT. From Figure 7, we find that the model trained after Critique-RL Stage II not only maintains strong discriminability but also provides accurate and constructive suggestions, outperforming the model trained after Critique-RL Stage I.

906

907

908

909

910

911

# Algorithm 1: Critique-RL

**Input:** Actor model  $\pi_{\theta}$ , base critique model  $\pi_{\phi}$ , SFT dataset  $\mathcal{D}_{SFT}$ , RL dataset  $\mathcal{D}_{RL}$ , function that extracts the correctness of a response judged by a critique f, oracle reward function  $r_{oracle}$ , discrimination reward function  $r_{dis}$ . **Procedure** Supervised Fine-tuning:

$$\pi_{\phi}^{\text{SFT}} \leftarrow \pi_{\phi}$$

Update  $\pi_{\phi}^{\text{SFT}}$  by minimizing  $\mathcal{L}_{\text{SFT}}(\phi) = \mathbb{E}_{(x,y,c)\sim\mathcal{D}_{\text{SFT}}}\left[\log \pi_{\phi}(c|x,y)\right];$ 

# **Procedure** Critique-RL Stage I: optimizating discriminability through direct reward signals. $\pi_{\phi}^{\text{Stage-I}} \leftarrow \pi_{\phi}^{\text{SFT}};$

for batch in  $\mathcal{D}_{RL}$  do

**for** x in batch **do** 

Generate y and c with  $\pi_{\theta}$  and  $\pi_{\phi}^{\text{Stage-I}}$ ;

Compute discrimination reward with  $r_{dis}(x, y, c) = \mathbb{1}(f(x, y, c) = r_{oracle}(x, y));$ 

end

Update  $\pi_{\phi}^{\text{Stage-I}}$  by maximizing

$$\mathbb{E}_{c \sim \pi_{\phi}^{\text{Stage-I}}(\cdot|x,y)} \Big[ r_{\text{dis}}(x,y,c) - \beta \text{KL}(\pi_{\phi}^{\text{SFT}}(c|x,y) || \pi_{\phi}^{\text{Stage-I}}(c|x,y)) \Big];$$

end

**Procedure** Critique-RL Stage II: optimization helpfulness while maintaining discriminability.  $\pi_{\phi}^{\text{Stage-II}} \leftarrow \pi_{\phi}^{\text{Stage-II}};$ 

for batch in  $\mathcal{D}_{RL}$  do for x in batch do Generate y, c and y' with  $\pi_{\theta}$  and  $\pi_{\phi}^{\text{Stage-II}}$ ; Compute discrimination reward with  $r_{\text{dis}}(x, y, c) = \mathbb{1}(f(x, y, c) = r_{\text{oracle}}(x, y))$ ; Compute refinement reward with  $r_{\text{refine}} = r_{\text{oracle}}(x, y')$ ; end Update  $\pi_{\phi}^{\text{Stage-II}}$  by maximizing  $\mathbb{E}_{c \sim \pi_{\phi}^{\text{Stage-II}}(\cdot|x,y), y' \sim \pi_{\theta}(\cdot|x,y,c)} [r_{\text{refine}} + \beta_1 r_{\text{dis}}(x, y, c) - \beta_2 \text{KL}(\pi_{\phi}^{\text{Stage-I}}(c|x, y)||\pi_{\phi}^{\text{Stage-II}}(c|x, y))]$ . end

<b>Problem:</b> Each person in a certain household consumes 0.2 kg of rice every meal. Suppo bag of rice last?	sing 5 members of the household eat rice every lunch and dinner, how many weeks will a 42 $\mathrm{kg}$
<ul> <li>Ground Truth Answer:</li> <li>Each person eats 0.2 kg/meal x 2 meals/day = &lt;&lt;0.2*2=0.4&gt;&gt;0.4 kg of rice per day.</li> <li>So, the 5 members consume 0.4 kg/day/person x 5 people = &lt;&lt;0.4*5=2&gt;&gt;2 kg of rice per day.</li> <li>In a week, the household consumes 2 kg/day x 7 days/week = &lt;&lt;2*7=14&gt;&gt;14kg of rice.</li> <li>Thus, 42 kg of rice will last 42 kg / 14 kg/week = &lt;&lt;42/14=3&gt;&gt;3 weeks. The answer is 3.</li> </ul>	Actor Model (original response): Let's break it down step by step: 1. The household consumes 0.2 kg of rice per meal * 2 meals per day = 0.4 kg of rice per day. 2. The household consumes 0.4 kg of rice per day * 5 days per week = 2 kg of rice per week. 3. A 42 kg bag of rice will last 42 kg / 2 kg per week = 21 weeks. The answer is 21. Correctness Verifier: S
Critique Model after SFT	Critique Model after Critique-RL Training
<ul> <li>Critique Model (critique):</li> <li>Step sentence: The household consumes 0.2 kg of rice per meal * 2 meals per day = 0.4 kg of rice per day.</li> <li>Correctness of the step: Correct</li> <li>Explanation: This step correctly calculates the daily rice consumption by multiplying the amount consumed per meal (0.2 kg) by the number of meals per day (2).</li> <li>Step sentence: The household consumes 0.4 kg of rice per day * 5 days per week = 2 kg of rice per week.</li> <li>Correctness of the step: Correct</li> <li>Explanation: This step correctly calculates the weekly rice consumption by multiplying the daily consumption (0.4 kg) by the number of days in a week (5).</li> </ul>	Critique Model (critique):         Step sentence: The household consumes 0.2 kg of rice per meal * 2 meals per day = 0.4 kg of rice per day.         Correctness of the step: Wrong         Explanation: This step is wrong because it incorrectly calculates the daily rice consumption. Since 5 members of the household eat rice every lunch and dinner, and each person consumes 0.2 kg of rice per meal, the daily consumption is: 5 members × 2 meals/day × 0.2 kg/meal = 2 kg/day         Step sentence: The household consumes 0.4 kg of rice per day * 5 days per week = 2 kg of rice per week.         Correctness of the step: Wrong         Explanation: This step is wrong because it incorrectly calculates the weekly rice consumption. Given the correct daily consumption is 2 kg/day, the weekly consumption is:
Step sentence: A 42 kg bag of rice will last 42 kg / 2 kg per week = 21 weeks. Correctness of the step: Correct Explanation: This step correctly calculates the number of weeks the 42 kg bag of rice will last by dividing the total amount of rice by the weekly consumption (2 kg per week).	2 kg/day × 7 days/week = 14 kg/week Step sentence: A 42 kg bag of rice will last 42 kg / 2 kg per week = 21 weeks. Correctness of the step: Wrong Explanation: This step is wrong because it incorrectly calculates the duration the rice will last. Given the correct weekly consumption is 14 kg/week, the duration the 42 kg bag will
Correctness of the final answer: Correct	last is: $42 \text{ kg} \div 14 \text{ kg/week} = 3 \text{ weeks}$
Explanation: The final answer of 21 weeks is correct based on the calculations provided in the solution. Discrimination Verifier: 😣	Correctness of the final answer: Wrong Explanation: The final answer of 21 weeks is incorrect. The correct answer is 3 weeks. Discrimination Verifier:
<ul> <li>Actor Model (refinement response):</li> <li>Let's break it down step by step:</li> <li>1. Calculate the daily rice consumption: The household consumes 0.2 kg of rice per meal. Since there are 2 meals per day, the daily consumption is:</li> <li>0.2 kg/meal × 2 meals/day = 0.4 kg/day.</li> <li>2. Calculate the weekly rice consumption: The household consumes 0.4 kg of rice per day. Since there are 5 days per week, the weekly consumption is:</li> <li>0.4 kg/day × 5 days/week = 2 kg/week.</li> <li>3. Calculate how many weeks a 42 kg bag of rice will last: A 42 kg bag of rice will last:</li> <li>42 kg + 2 kg/week = 21 weeks.</li> <li>The answer is 21.</li> </ul>	Actor Model (refinement response):         Let's break it down step by step:         To determine how many weeks a 42 kg bag of rice will last for a household, we need to consider the following:         1. Daily Rice Consumption Calculation: There are 5 members in the household. Each member consumes 0.2 kg of rice per meal. Each member cats rice at 2 meals per day (lunch and dinner). Therefore, the total daily rice consumption is:         5 members × 2 meals/day × 0.2 kg/meal = 2 kg/day         2. Weekly Rice Consumption Calculation: The daily consumption is 2 kg. There are 7 days in a week. Therefore, the weekly rice consumption is:         2 kg/day×7 days/week = 14 kg/week         3. Duration the Rice Will Last: The household has 42 kg bag of rice. The weekly consumption is 14 kg. Therefore, the number of weeks the rice will last is:         42 kg + 14 kg/week = 3 weeks         The answer is 3.

Figure 6: Example 1 of qualitative analysis. The actor's original response is incorrect. The model after SFT is unable to detect errors in the response, leading the actor's refinement response to retain the same errors. However, the model trained after Critique-RL identifies the errors in the original response and provides detailed, constructive suggestions for modification, leading to the correct refinement response.

<b>Problem:</b> Dennis uses 1 pound of butter for every dozen croissants that he makes. He no of butter get one half off. If the butter costs \$4.00 a pound, how much will it cost him to put	eeds to make 6 dozen croissants. The grocery store currently has a promotion for buy one pound rchase 6 pounds of butter?
<ul> <li>Ground Truth Answer:</li> <li>1 pound of butter makes 1 dozen croissants and he needs to make 6 dozen so he needs 1%6 = &lt;&lt;1%6-6&gt;&gt;6 pounds of butter.</li> <li>The butter is \$4.00 a pound and currently on sale for buy one get one half off so every unit of 2 pounds of butter will cost 4*1.5 = \$&lt;&lt;4*1.5 = 6.00&gt;&gt;6.00.</li> <li>He needs 6 pounds of butter and the current promotion is for a unit of 2 pounds of butter and the current promotion is for a unit of 2 pounds of butter so the needs to buy 6/2 = &lt;&lt;6/2=3&gt;3 units worth of butter.</li> <li>The sale price is \$6.00 per 2 pounds of butter and he needs to buy 3 units of butter so 6*3 = \$&lt;&lt;6*3=18.00&gt;&gt;18.00 worth of butter.</li> </ul>	Actor Model (original response): Let's break it down step by step: 1. He needs to buy 3 pounds of butter because 6 / 2 = 3. 2. 3 pounds of butter would cost \$12 because 3 x 4 = 12. 3. He will pay \$6 because 12 / 2 = 6. The answer is 6. Correctness Verifier: 8
<b>Critique Model after Critique-RL Stage I Training</b> <b>Critique Model (critique):</b> <b>Step sentence:</b> He needs to buy 3 pounds of butter because 6/2 = 3. <b>Correctness of the step: Wrong</b> <b>Explanation:</b> Dennis needs to make 6 dozen croissants, and he uses 1 pound of butter per dozen. Therefore, he needs 6 pounds of butter. The calculation 6/2	Critique Model after Critique-RL Stage II Training Critique Model (critique): Step sentence: He needs to buy 3 pounds of butter because 6/2 = 3. Correctness of the step: Wrong Explanation: Dennis needs to make 6 dozen croissants, and he uses 1 pound of butter per dozen. Therefore, he needs 6 pounds of butter. The promotion does not change the fact that
= 3 is incorrect. He needs to buy 6 pounds, not 3. Step sentence: 3 pounds of butter would cost \$12 because 3 x 4 = 12. Correctness of the step: Wrong Explanation: Although the calculation of the cost is correct (3 pounds at \$4 per pound equals \$12), the premise is wrong. Dennis needs 6 pounds of butter, not 3. The correct calculation should be 6 pounds at \$4 per pound, which equals \$24.	The needs 6 pounds. The calculation 6/2 – 3 is interval to the problem. Step sentence: 3 pounds of butter would cost \$12 because 3 x 4 = 12. Correctness of the step: Wrong Explanation: Although the promotion is "buy one pound of butter, get one half off," Dennis needs 6 pounds of butter. The promotion applies to each pound he buys, so for every 2 pounds, he pays \$4 + 52 = 56. Since he needs 6 pounds, he can take advantage of the promotion 3 times (6 pounds / 2 pounds per promotion). Therefore, the total cost is 3 * \$6 = $\frac{6}{3}$ (9)
Step sentence: He will pay \$6 because $12/2 = 6$ . Correctness of the step: Wrong Explanation: The calculation $12/2 = 6$ is correct based on the promotion (buy one pound get one half off), but it does not apply to the correct amount of butter needed. Demnis needs 6 pounds, not 3. The correct calculation should be based on the full price of 6 pounds, which is \$24.	S16. Step sentence: He will pay \$6 because $12/2 = 6$ . Correctness of the step: Wrong Explanation: The calculation " $12/2 = 6$ " is incorrect. Based on the previous explanation, the total cost is $3 \times \$6 = \$1\$$ . Therefore, he will pay $\$1\$$ instead of $\$6$ . Correctness of the final answer: Wrong
Correctness of the final answer: Wrong Explanation: The final answer of 6 is incorrect. Based on the correct calculation. Dompis needs 6 noutles of butter at 54 per pound, which totals \$74.	Explanation: The final answer of 6 is incorrect, Given the promotion, Dennis will pay \$18 for 6 pounds of butter. Discrimination Verifier:
Actor Model (refinement response): Let's break it down step by step: To determine how much it will cost Dennis to purchase 6 pounds of butter, we need to consider the following: 1. Butter Requirement: Dennis needs to make 6 dozen croissants, and he uses 1 pound of butter per dozen. Therefore, he needs 6 pounds of butter. 2. Promotion Details: The store has a promotion where you buy one pound of butter and get the next pound at half price. 3. Cost Calculation: Full Price for 6 Pounds: Since the promotion applies to each pair of pounds, we need to calculate the cost based on the full price for 6 pounds. Cost for 6 Pounds: The cost of 6 pounds of butter at S4 per pound is S24. Therefore, the total cost for 6 pounds of butter is \$24. The pound we read to a store of the pound of soft of pounds of butter at S4 per pound is \$24. The pound we read to a store for 6 pounds of butter is \$24. The pound we read to a store of the pound of soft of pounds of butter is \$24. The pound we read to a store of the pound of soft pound is \$24. The pound we read to a store of pound of pound is \$24. The pound we read to be pound of a pound of butter is \$24. The pound we read to be pound of a pound of butter is \$24. The pound we read to be pound of butter is \$24. The pound we read to be pound of butter is \$24. The pound is \$24. The pound is the pound is \$24. The pou	<ul> <li>Actor Model (refinement response): Let's break it down step by step: To determine how much it will cost Dennis to purchase 6 pounds of butter, we need to consider the following:</li> <li>Butter Requirement: Dennis needs to make 6 dozen croissants, and he uses 1 pound of butter per dozen. Therefore, he needs 6 pounds of butter.</li> <li>Promotion Details: The promotion is \"buy one pound of butter, get one half off.\" This means for every 2 pounds of butter purchased, Dennis pays for 1 pound at full price and gets the second pound at half price.</li> <li>Cost Calculation:</li> <li>For every 2 pounds of butter, Dennis pays \$4 for the first pound and \$2 for the second pound, totaling \$6 for 2 pounds.</li> <li>Since Dennis needs 6 pounds, he can take advantage of the promotion 3 times (6 pounds / 2 pounds per promotion).</li> <li>Therefore, the total cost is 3 * \$6 = \$18.</li> </ul>
Correctness Verifier: 😣	The answer is 18 Correctness Verifier:

Figure 7: Example 2 of qualitative analysis. The actor's original response is incorrect. The model trained after Critique-RL Stage I is able to detect this error, demonstrating its discriminability. However, the model provides the actor with low-quality suggestion, causing the actor's refinement response to be incorrect. In contrast, for the same erroneous original response, model trained after Critique-RL Stage II not only detects the error but also offers a constructive suggestion, ultimately leading to the correct refinement response, demonstrating the advantage of two-stage RL process.