Enhancing Reasoning Abilities of Small LLMs with Cognitive Alignment

Anonymous ACL submission

Abstract

The reasoning capabilities of large language reasoning models (LRMs), such as OpenAI's o1 and DeepSeek-R1, have seen substantial advancements through deep thinking. However, these enhancements come with significant resource demands, underscoring the need for training effective small reasoning models. A critical challenge is that small models possess different reasoning capacities and cognitive trajectories compared with their larger counterparts. Hence, directly distilling chain-of-thought (CoT) results from large LRMs to smaller ones can sometimes be ineffective and often requires a substantial amount of annotated data. In this paper, we first introduce a novel Critique-Rethink-Verify (CRV) system, designed for training smaller yet powerful LRMs. Our CRV system consists of multiple LLM agents, each specializing in unique abilities: (i) critiquing the CoT qualities according to the cognitive capabilities of smaller models, (ii) rethinking and refining these CoTs based on the critiques, and (iii) verifying the correctness of the refined results. Based on the CRV system, we further propose the Cognitive Preference Optimization (CogPO) algorithm to continuously enhance the reasoning abilities of smaller models by aligning their reasoning processes with their cognitive capacities. Comprehensive evaluations on challenging reasoning benchmarks demonstrate the efficacy of our CRV+CogPO framework, which outperforms other methods by a large margin. [1]

1 Introduction

The remarkable progress in language reasoning models (LRMs) has revolutionized NLP (Zhao et al., 2023). Recently, leading models such as OpenAI's o1[2] and DeepSeek-R1 (DeepSeek-AI, 2025) have leveraged slow thinking to solve complex tasks. Despite their impressive capabilities, the scale of these models results in substantial computational demands. Consequently, there is a growing need to train reasoning models with fewer parameters.

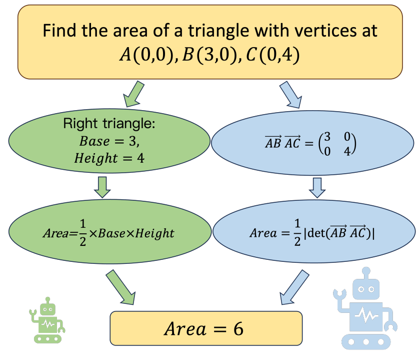


Figure 1: A motivation example. Large models (right) apply vector-based algebraic abstraction to solve the problem, while small models (left) employ simple formulaic geometric decomposition. This trajectory mismatch underscores the inefficacy of direct CoT distillation across models with substantial capacity gaps.

A straightforward approach to address this challenge is the direct distillation of Chain-of-Thought (CoT) outputs (Wei et al., 2022a) or other deep thoughts (such as Tree-of-Thought (Yao et al., 2023b)) from larger LRMs to smaller ones. This technique is widely applied to improve the capacities of smaller LRMs (Hsieh et al., 2023; Shridhar et al., 2022; Li et al., 2023; Yue et al., 2024). However, smaller models[3] inherently possess different reasoning capacities and cognitive trajectories when solving problems compared to their larger counterparts, as illustrated in Figure 1. Similar findings have also been presented in (Li et al., 2022; Zhang et al., 2024; Hu et al., 2024; Li et al., 2024). This phenomenon indicates that direct distillation of CoTs from larger models can sometimes

[1] Source codes, datasets and models will be released upon paper acceptance.

[2] https://openai.com/o1/

[3] In this work, we regard smaller LLMs as decoder-only language models typically with fewer than 10B parameters.

be ineffective due to the large capacity gap. Thus, a natural question arises: *How can we improve the reasoning abilities of smaller LRMs to align with their own cognitive capacity?*

In this paper, we introduce "Critique-Rethink-Verify" (CRV), a novel system to enhance the reasoning capabilities of smaller models. CRV leverages multiple LLM agents, each with specialized functions and working in synergy. These functions include (i) critiquing the CoT by considering the cognitive limits of smaller LRMs, (ii) rethinking and refining these CoTs, integrating the feedback received from the previous critiques, and (iii) verifying the accuracy and validity of the refined reasoning paths. Extending the direct preference optimization (DPO) technique (Rafailov et al., 2023), we further propose the cognitive preference optimization (CogPO) algorithm to align the reasoning process with the cognitive capacities of smaller LRMs on the basis of CRV system. Ultimately, the reasoning performance of smaller models can be improved effectively.

In the experiments, the effectiveness of our approach is evaluated on several challenging reasoning benchmarks that are difficult for models with limited parameter sizes, such as AIME 2024, MATH-500 (Lightman et al., 2023), GPQA-Diamond (Rein et al., 2023), and LiveCodeBench. The results indicate that the small LRMs trained using the CRV+CogPO framework achieve outstanding reasoning performance. In summary, we make the following major contributions:

- We present the CRV system for training small yet powerful LRMs, leveraging multiple LLM agents, each specializing in unique tasks.
- We propose the CogPO algorithm that continuously enhances the reasoning abilities of small models by aligning their reasoning processes with their cognitive capacities.
- Evaluations on challenging benchmarks demonstrate that the CRV+CogPO framework significantly improves the reasoning performance of small models, outperforming other popular training methods.

2 Related Work

2.1 Prompting LLMs to Reason

Prompting strategies to improve reasoning in LLMs have become a critical focus. Initial studies showed that LLMs could perform basic reasoning tasks using meticulously crafted prompts, such as linguistic analysis (Chen et al., 2021) and commonsense inference (Latcinnik and Berant, 2020; Shwartz et al., 2020). To name a few, Chain-of-Thought (CoT) (Wei et al., 2022b) prompting explicitly guides LLMs through step-by-step reasoning, enabling them to decompose complex problems into manageable intermediate reasoning steps. Tree-of-Thought (ToT) (Yao et al., 2023a) prompting introduces a hierarchical structure to reasoning trajectories, allowing models to explore multiple solution paths. Furthermore, self-refine (Shinn et al., 2023; Madaan et al., 2023) prompting incorporates verification checkpoints, where models validate intermediate results before advancing.

2.2 Reasoning LLMs

With the advancement of LLMs, model capabilities have steadily improved (Chen and Varoquaux, 2024; Bansal et al., 2024). Models with approximately 7B to 14B parameters show remarkable performance, and their fine-tuning costs have become increasingly feasible. This has led to the emergence of specialized small models tailored for mathematical and code-related reasoning tasks such as Qwen-Math[4], Qwen-Coder[5], and Macro-o1 (Zhao et al., 2024).

Recent studies (Shridhar et al., 2023; Yan et al., 2023; Liang et al., 2024; Yuan et al., 2024) have investigated fine-tuning methods to enhance the reasoning abilities of smaller models. By utilizing intermediate reasoning steps, LLMs can iteratively refine their outputs (Jiang et al., 2024; Wang et al., 2024; Chen et al., 2025). This methodology facilitates the development of small reasoning models, particularly following the release of stronger reasoning models such as DeepSeek-R1 (DeepSeek-AI, 2025) and QwQ-32B[6].

2.3 Alignment Training

To effectively train LLMs, a reinforcement learning stage is typically employed after the supervised fine-tuning (SFT) phase, which serves to improve the model's alignment towards certain objectives. Reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) has shown effectiveness in aligning LLMs with human feedback. A potential drawback of RLHF is the ex-

[4]https://qwenlm.github.io/blog/qwen2.5-math/
[5]https://qwenlm.github.io/blog/qwen2.5-coder-family/
[6]https://qwenlm.github.io/blog/qwq-32b/

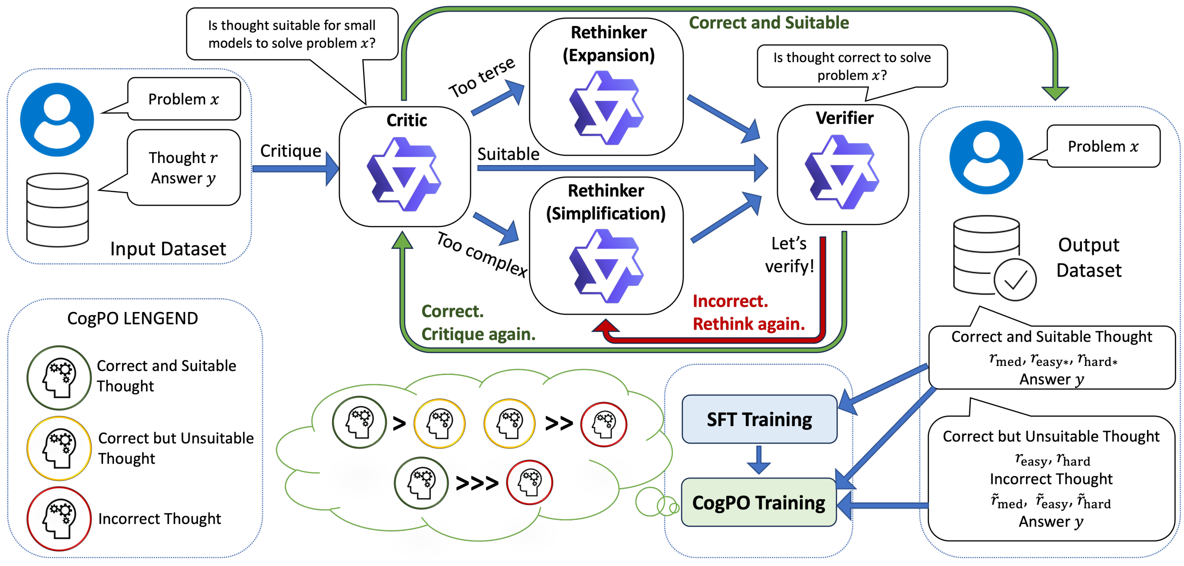


Figure 2: Overview of our CRV+CogPO framework, consisting of two synergistic phases: (1)SFT training with cognitively aligned data generated by CRV system, and (2) CogPO: dynamic β adjustment preference optimization training using cognitive reasoning pairs with different quality gaps. **Disclaimer:** We use the Qwen logo as our backbones; however, any LLMs with sufficient capabilities can serve as the agents as well.

plicit need for a reward model and the unstable RL training process. Direct preference optimization (DPO) (Rafailov et al., 2023) trains LLMs based on chosen and rejected responses. Since the introduction of DPO, several approaches have been proposed to enhance its efficacy and efficiency. For example, CPO (Xu et al., 2024) extends DPO to avoid generating adequate but not perfect machine translations. SimPO (Meng et al., 2024) simplifies DPO by eliminating the reference model. KTO (Ethayarajh et al., 2024) and NCA (Chen et al., 2024) develop novel optimization goals that leverage unpaired data for model alignment. Furthermore, SPPO (Wu et al., 2024b) employs on-policy sampling to generate preference data, outperforming off-policy DPO methods. In our work, we extend DPO to align reasoning abilities with the cognitive limits of small LLMs.

3 Proposed Approach

3.1 Overall Framework

Our framework consists of two synergistic phases: (1) SFT with cognitively aligned data generated by CRV system, and (2) CogPO with dynamic β adjustment. As illustrated in Figure 2, the CRV system first refines data tailored to the cognitive capacity of smaller LRMs for SFT training, and CogPO further aligns reasoning preferences through suitability-aware optimization using pairs with different quality gaps. This design ensures that the model initially acquires capacity-matched reasoning patterns, followed by the refinement of its decision boundaries through gap-sensitive learning.[7]

3.2 The CRV System

The CRV system employs LLM agents to construct the SFT dataset aligned with the cognitive limits of smaller models to be trained. The input to CRV system is an initial training set $\mathcal{D}_{\text{SFT}} = \{(x, y, r_{\text{orig}})\}$, where the three elements denote the problem, the correct answer, and the original reasoning process generated by any large LRMs (e.g., DeepSeek-R1), which has been validated as correct. The following provides descriptions of each agent in the CRV system.

3.2.1 Critic

An LLM agent first evaluates the appropriateness of reasoning processes for the target small LLM (denoted as π_{base}). For each $(x, y, r_{\text{orig}}) \in \mathcal{D}_{\text{SFT}}$, the Critic assesses r_{orig} using the criteria of *Cognitive Matching Degree*, where the Critic checks whether the complexity and difficulty of r_{orig} aligns with the cognitive capacity of π_{base}. Specifically, the Critic classifies the reasoning processes into three subsets: i) $\mathcal{D}_{\text{easy}} : (x, y, r_{\text{easy}})$, cases where the reasoning process is overly terse, making it difficult for π_{base} to follow; ii) $\mathcal{D}_{\text{med}} : (x, y, r_{\text{med}})$,

[7]The decision boundary refers to the model's ability to judge whether the produced CoT is correct and aligns with its own cognitive capabilities, enabling it to successfully solve problems following its CoT.

Level/Model Size	1.5B	7B	32B
Easy	195	80	19
Medium	296	389	354
Hard	9	31	127

Table 1: Complexity distributions of CoTs generated by different sizes of DeepSeek-R1-Distill-Qwen models.

cases with appropriate steps that enable successful problem solving; and iii) $\mathcal{D}_{\text{hard}} : (x, y, r_{\text{hard}})$, cases with overly redundant or excessively complex reasoning steps that exceed the comprehension of π_{base}, making it extremely prone to fail to guide π_{base} in solving x.

Remarks. An intuitive approach would be to use π_{base} itself as the Critic. However, due to its small parameter size (e.g., 7B), certain CoTs exceed π_{base}'s comprehension, rendering it incapable of reliable complexity classification. Thus, we leverage the same LLM for the Rethinker (denoted as π_{large}) to serve as the Critic, forcing it to "think" from the perspective of the small model π_{base}. A detailed analysis of the Critic choices is provided in the Experiments 4.3 and Appendix A.5.

Hypothesis Verification. To further verify that the complexity levels of CoTs are closely related to the cognitive capacities of reasoning models, we conduct an experiment in which we evaluate DeepSeek-R1-Distill-Qwen-1.5B/7B/32B on MATH500, collecting each model's outputs. We employ the Critic to rate the level of model's CoT outputs; each CoT is evaluated three times, and the final rating is determined by majority vote. For each model, we quantify the distribution of these CoTs across different complexity levels in Table 1. As shown, DeepSeek-R1-Distill-Qwen-1.5B yields the largest number of simple CoTs, while DeepSeek-R1-Distill-Qwen-32B generates the greatest number of difficult CoTs.

These findings demonstrate that the complexity of CoTs escalate as the model size increases, suggesting that larger models possess higher reasoning and cognitive capacities. Consequently, overly terse or complex CoTs may not be suitable for training models with lower cognitive abilities. It is therefore essential to use CoTs that align with the model's cognitive trajectory to improve its reasoning capabilities, a strategy akin to "teaching according to the student's ability."

3.2.2 Rethinker

An LLM agent π_{large} is tasked with rewriting reasoning processes to achieve cognitive alignment. For each $(x, y, r_{\text{easy}}) \in \mathcal{D}_{\text{easy}}$, the Rethinker expands r_{easy} by adding necessary steps for easier understanding, i.e., $r_{\text{easy*}} = \pi_{\text{large}}(x, y, r_{\text{easy}})$. Similarly, for each $(x, y, r_{\text{hard}}) \in \mathcal{D}_{\text{hard}}$, the Rethinker simplifies r_{hard} by removing redundancies or using simpler methods to solve the problem grounded in the correct answer: $r_{\text{hard*}} = \pi_{\text{large}}(x, y, r_{\text{hard}})$. Cases of the rewriting process of the Rethinker are shown in Tables 11 and 12.

3.2.3 Verifier

Finally, we leverage the LLM agent π_{base} to validate the correctness of r_{med}, $r_{\text{easy*}}$, and $r_{\text{hard*}}$ in order to preserve the high quality of the dataset. It predicts whether π_{base} can derive the correct answer y from the rewritten thoughts $r_{\text{easy*}}$ or $r_{\text{hard*}}$. Note that r_{med} has already been validated as correct in the original dataset, and we send r_{med} to the Verifier to further ensure data quality,

After verification, incorrect cases are sent back to the Rethinker to be continuously rewritten until they pass verification. In the implementation, cases that fail to pass verification after three iterations are discarded. For the cases that pass verification, we invoke the Critic to make the judgment again (please refer to Figure 2 for the algorithmic flow).

The final SFT dataset is composed of verified medium-level data: $\mathcal{D}_{\text{SFT*}} = \mathcal{D}_{\text{med}} \cup \mathcal{D}_{\text{easy*}} \cup \mathcal{D}_{\text{hard*}}$, where $\mathcal{D}_{\text{med}}$ denotes the verified medium-level data, and $\mathcal{D}_{\text{easy*}}$ and $\mathcal{D}_{\text{hard*}}$ represent the rewritten versions of $\mathcal{D}_{\text{easy}}$ and $\mathcal{D}_{\text{hard}}$ that have passed verification and have been re-rated as medium by the Critic, respectively. $\mathcal{D}_{\text{SFT*}}$ serves as the SFT training set in the CRV stage. Prompt templates used in CRV system are provided in Appendix C.

3.3 Cognitive Preference Optimization

The CogPO algorithm aligns CoT processes of smaller LLMs with their inherent cognitive capacities, following the SFT training using CRV system.

3.3.1 Preliminaries

Briefly speaking, the CogPO algorithm is extended from DPO (Rafailov et al., 2023) and its variants. Let y_w and y_l be the chosen and rejected responses for an instruction x (not restricted to reasoning problems addressed in this work), respectively. We further denote π_θ as the model to be optimized after SFT and π_{ref} as the reference model. DPO seeks to maximize the following margin: $M_\beta(x, y_w, y_l) = \beta \cdot \left(\log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right)$

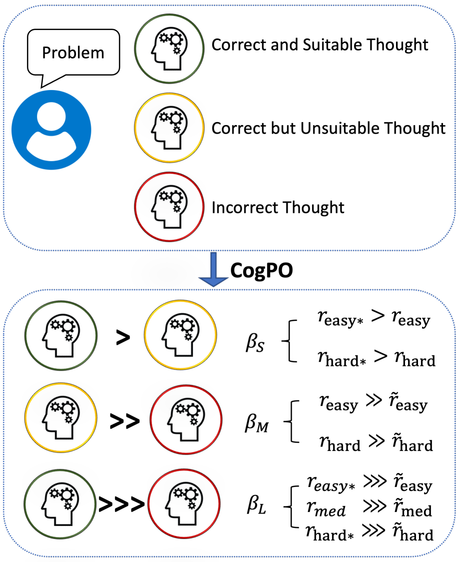


Figure 3: An illustration of CogPO, showing the different preference gaps between CoT pairs and the corresponding mini-tasks.

where β is a temperature hyperparameter. Based on $M_\beta(x, y_w, y_l)$, the DPO loss is defined as:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log \sigma(M_\beta(x, y_w, y_l)). \tag{1}$$

The settings of β are critical to the performance of DPO. β-DPO (Wu et al., 2024a) further adjusts β according to $M_\beta(x, y_w, y_l)$, either at the instance level or batch level, allowing the model to adapt β based on the reward differential of the input data.

3.3.2 Algorithmic Description

As noted, DPO and β-DPO do not require any prior knowledge of how the model learns the user's preferences. We suggest that this type of prior knowledge is critical for training better smaller reasoning models, as the cognitive trajectories of large and small models often differ (Li et al., 2022; Zhang et al., 2024; Hu et al., 2024), which may not be directly reflected in the reward differential. We propose CogPO to align reasoning preferences by encoding more prior knowledge and continuously training on a series of *mini-tasks*.

We leverage the Rethinker in CRV to also generate incorrect reasoning processes when asked to rewrite the original thought r_{orig} (prompt template is provided in Appendix C). The incorrect thoughts are denoted as $\tilde{r}_{\text{med}}$, $\tilde{r}_{\text{easy}}$, and $\tilde{r}_{\text{hard}}$, based on their origins from $\mathcal{D}_{\text{med}}$, $\mathcal{D}_{\text{easy}}$, and $\mathcal{D}_{\text{hard}}$. These thoughts contain factual errors or invalid reasoning steps, which can mislead π_{base}, rendering it impossible to solve x. Thus, we categorize the properties of all the thoughts we have collected into the following three types: i) r_{med}, $r_{\text{easy}*}$, and $r_{\text{hard}*}$: medium-level reasoning processes that are both correct and cognitively suitable for π_{base}; ii) r_{easy} and r_{hard}: easy or hard thoughts that are correct but unsuitable for π_{base}; iii) $\tilde{r}_{\text{med}}$, $\tilde{r}_{\text{easy}}$, and $\tilde{r}_{\text{hard}}$: incorrect reasoning processes with logical flaws or invalid reasoning steps (regardless of the difficulty levels). To define the mini-tasks used for CogPO training, we consider the preference gaps in these three types of CoT pairs as follows:

1. Small Gap Mini-task: The pairs are ($r_{\text{easy}*}$, r_{easy}) and ($r_{\text{hard}*}$, r_{hard}). Both are correct but differ in complexity (suitable vs. unsuitable for π_{base}). We treat $r_{\text{easy}*}$ and $r_{\text{hard}*}$ as chosen reasoning processes (r_w), and r_{easy} and r_{hard} as rejected (r_l).

2. Medium Gap Mini-task: The pairs are (r_{easy}, $\tilde{r}_{\text{easy}}$) and (r_{hard}, $\tilde{r}_{\text{hard}}$). The former are correct but unsuitable, while the latter are completely incorrect. As correctness is more important than suitability for our model, the preference gap of this mini-task should be higher than that in the previous case. For this mini-task, r_{easy} and r_{hard} are treated as r_w, while $\tilde{r}_{\text{easy}}$ and $\tilde{r}_{\text{hard}}$ are treated as r_l.

3. Large Gap Mini-task: The pairs are (r_{med}, $\tilde{r}_{\text{med}}$), ($r_{\text{easy}*}$, $\tilde{r}_{\text{easy}}$), and ($r_{\text{hard}*}$, $\tilde{r}_{\text{hard}}$). Intuitively, the preference gaps should be the largest between suitable and correct thoughts and incorrect ones. Here, r_{med}, $r_{\text{easy}*}$, and $r_{\text{hard}*}$ are treated as r_w, while $\tilde{r}_{\text{med}}$, $\tilde{r}_{\text{easy}}$, and $\tilde{r}_{\text{hard}}$ are treated as r_l.

Following our modeling framework, each training instance (x, r_w, r_l) receives its specific β value, as illustrated in Figure 3. The CogPO objective function aggregates these preferences:

$$\mathcal{L}_{\text{CogPO}} = -\mathbb{E}_{(x,r_w,r_l)\sim\mathcal{D}} \log \sigma(M_{\beta_{\text{CogPO}}}(x, r_w, r_l)), \tag{2}$$

where β_{CogPO} is selected from $\{\beta_{\text{S}}, \beta_{\text{M}}, \beta_{\text{L}}\}$, depending on the specific types of mini-tasks (with $\beta_{\text{S}} < \beta_{\text{M}} < \beta_{\text{L}}$, corresponding to the three gaps). Overall, our CogPO algorithm enables granular preference learning: strong regularization (β_{L}) for validity discrimination, moderate guidance (β_{M}) for suitability alignment, and subtle refinement (β_{S}) for reasoning style adaptation. This design provides more control over the alignment process, leading to further improvements on the basis of SFT (using CRV system).

Remarks. CogPO can be naturally combined with β-DPO (Wu et al., 2024a). We can redefine the β values $\{\beta_{\text{S}}, \beta_{\text{M}}, \beta_{\text{L}}\}$ as follows: $\beta_i^* = \beta_i + \alpha \cdot (M_i - M_0) \cdot \beta_i$ where β_i is chosen from $\{\beta_{\text{S}}, \beta_{\text{M}}, \beta_{\text{L}}\}$ based on the corresponding gap type, M_i is the

Dataset/Model	Zero-shot	SFT	CRV+SFT	DPO	β-DPO	SimPO	CogPO
AIME2024	10.0	20.0	**26.7**	23.3	23.3	26.7	**30.0**
MATH-500	73.6	80.0	**84.0**	83.4	83.8	84.2	**84.4**
GSM8K	89.5	92.3	**92.7**	92.6	93.0	92.6	**93.3**
GPQA Diamond	33.3	37.4	**40.9**	40.0	37.4	**40.9**	**40.9**
LiveCodeBench V2	30.7	31.3	**34.4**	34.4	35.8	36.2	**36.6**
MMLU	71.9	76.1	**76.5**	76.1	76.0	**76.5**	**76.5**
OlympiadBench (math-en)	40.1	43.6	**45.8**	45.7	46.5	46.0	**46.6**

Table 2: Performance comparison of various training methods. The LLM backbone is Qwen2.5-7B-Instruct, and the training set is Bespoke-Stratos-17k. Results are shown for zero-shot (without further training), SFT, CRV+SFT, DPO, β-DPO, SimPO, and CogPO. DPO, β-DPO, SimPO, and CogPO are conducted on the same model checkpoints of CRV+SFT, using the same preference pair dataset. The metrics represent scores for these tasks, with the best results for each dataset in each group marked in bold and the second-best underlined.

Dataset/Model	LLaMA-O1	Macro-o1	Bespoke-Stratos-7B	Ours	OpenThinker-7B	Ours
Training Set Size	332K	60K	17K	17K	114K	114K
AIME2024	3.3	6.7	20.0	**30.0**	31.3	**43.3**
MATH500	28.6	38.4	82.0	**84.4**	83.0	**88.4**
GPQA Diamond	26.3	31.8	37.8	**40.9**	42.4	**42.9**
LiveCodeBench V2	1.6	24.9	36.1	**36.6**	39.9	**46.4**

Table 3: Comparison between our model and other small reasoning models in the open-source community. Specifically, we train two versions using our approach on Bespoke-Stratos-17k and OpenThoughts-114k, respectively, where the two training sets are the same with Bespoke-Stratos-7B and OpenThinker-7B, respectively.

instance-level reward differential, and M_0 is a predefined threshold as in (Wu et al., 2024a).[8]

4 Experiments

To evaluate the effectiveness of the CRV framework and the CogPO algorithm, we conduct a series of experiments on challenging reasoning benchmarks. Due to space limitation, datasets and experimental settings are shown in the Appendix A.1 and A.2.

4.1 Main Experimental Results and Ablations

We choose Bespoke-Stratos-17k as the training set. Table 2 presents the results of our CRV framework and the CogPO algorithm on various reasoning benchmarks. CRV+SFT surpasses direct SFT on all benchmarks. Building on CRV+SFT, CogPO further enhances the model's reasoning capability, surpasses other preference-optimization algorithms, and ultimately achieving the most outstanding performance, demonstrating its ability to align the model's reasoning processes with its cognitive capacities. These results reveal that our CRV+CogPO framework effectively enhances the reasoning capabilities of smaller models, outperforming other traditional methods by a large margin.

4.2 Comparison Against Other Models

We compare our trained 7B model with other models released in the open-source community. We consider two reasoning LLMs available before the launch of DeepSeek-R1, namely Macro-o1 (Zhao et al., 2024) and LLaMA-O1[9]. We also compare other models trained on datasets distilled from DeepSeek-R1, including Bespoke-Stratos-7B[10] and OpenThinker-7B[11]. Using our CRV+CogPO framework, we also train two models on the Bespoke-Stratos-17k and OpenThoughts-114k training sets, respectively. Thus, it is fair to compare our method against those of Bespoke-Stratos-7B and OpenThinker-7B. The results, along with the sizes of the training sets, are shown in Table 3. It can be observed that employing DeepSeek-R1-generated CoT data yields superior results. At the algorithmic level, both Bespoke-Stratos-7B and our model are trained on the 17K CoTs from DeepSeek-R1. Under identical data conditions, our model significantly outperforms Bespoke-Stratos-7B across all benchmarks and achieves performance comparable to OpenThinker-7B, which is trained on 114K CoTs from DeepSeek-R1. Moreover, when trained on the same dataset

[8]In our experiment, the combination does not yield substantial improvements, as prior knowledge is more important for our task. Hence, we stick to the usage of $\mathcal{L}_{\text{CogPO}}$.

[9]https://huggingface.co/SimpleBerry/LLaMA-O1-Supervised-1129

[10]https://huggingface.co/bespokelabs/Bespoke-Stratos-7B

[11]https://huggingface.co/open-thoughts/OpenThinker-7B

Model Backbone (The Critic)	AIME2024	MATH-500	GPQA-D	GSM8K	LCB V2	OlympiadBench
Qwen2.5-7B-Instruct	13.3	80.2	**40.9**	92.3	30.5	43.9
Qwen2.5-32B-Instruct	23.3	82.2	39.9	92.6	33.3	45.1
Qwen2.5-72B-Instruct	20.0	81.8	36.4	**92.7**	30.5	42.0
DeepSeek-R1-Distill-Qwen-32B	**26.7**	**84.0**	**40.9**	**92.7**	**34.4**	**45.8**

Table 4: Comparison using different backbones as the Critic. All the results are produced using CRV+SFT without CogPO on Bespoke-Stratos-17k.

Dataset/Model	Easy	Medium	Hard
AIME2024	13.3	23.3	16.7
MATH500	75.4	82.8	78.2
GPQA-D	34.3	37.4	33.3
LCB V2	31.9	36.2	32.5

Table 5: Experimental results on training data of different complexity levels.

Dataset/Model	SFT	w. C	w. CR	w. CRV
AIME2024	20.0	23.3	26.7	26.7
MATH500	80.0	83.4	83.2	84.0
GPQA-D	37.4	38.4	39.9	40.9
LCB V2	31.3	34.3	34.1	34.4

Table 6: Ablation results on the CRV system.

as OpenThinker-7B, our model substantially surpasses OpenThinker-7B on all benchmarks. These findings demonstrate that, given the same data, our CRV + CogPO training framework exhibits superior performance, confirming its effectiveness.

4.3 Study on Choices of the Critic

In the previous section, we claimed that using the small target LLM π_{base} as the Critic does not necessarily produce satisfactory results due to its limited parameter size. In contrast, larger LLMs π_{large} can "think like small models" better. The results of using different backbones as the Critic are shown in Table 4, with the backbones for the Rethinker and the Verifier unchanged. From the results, we can see that they confirm our findings, as larger models consistently perform better than the 7B model in almost all tasks. Among the three large agents, DeepSeek-R1-Distill-Qwen-32B exhibits the best performance based on majority voting across all testing sets. A detailed and in-depth analysis of the selection of the Critic is provided in Appendix A.5.

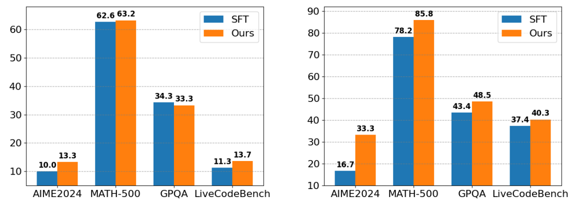
4.4 Training with CoT Datasets of Different Complexity Levels

To further investigate whether medium-level data are indeed the most suitable for base model, we conduct experiments on the OpenThoughts-114K dataset. We used the Critic to rate all CoTs in the dataset, then randomly sampled 10K CoTs from each of the derived easy, medium, and hard subsets to construct three training sets. We then perform SFT with Qwen2.5-7B-Instruct on these three training sets under identical configurations. The results are shown in Table 5, indicating that when the number of training data is the same, the model trained on the medium subset achieves the highest scores, fully supporting our hypothesis. The CoTs in the easy and hard sets are either too terse or overly complex, preventing the base model from effectively comprehending all CoTs in those sets. In contrast, the medium subset data align with the model's cognitive capabilities and thus yield the best results.

4.5 Study on Effectiveness of Critic, Rethinker and Verifier

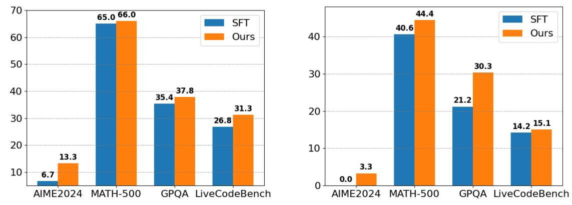
To further explore the collaborative mechanism within the CRV system and the individual roles and contributions of each module, we conduct extensive ablation experiments on the Bespoke-Stratos-17k dataset. Table 6 presents our ablation results. The "SFT" row reports results from directly performing SFT on the original dataset without any CRV intervention; the "w. C" row shows performance when only the Critic is applied before SFT, using only the traces rated as medium by the Critic for SFT; the "w. CR" row indicates results when both the Critic and the Rethinker participate prior to SFT, utilizing the medium-rated traces and the refined easy/hard traces that have not yet been verified; the "w. CRV" row reflects outcomes when the Critic, Rethinker, and Verifier are all applied.

As the Critic, Rethinker, and Verifier participate sequentially, the model's reasoning ability exhibits a progressively improving trend, which clearly illustrates the role of each component. Notably, "w. CR" experiences a performance drop on MATH500 and LCB V2, indicating that omitting the Verifier after the Rethinker's refinement could impair the model's reasoning ability. Therefore, each component of the CRV system plays an indispensable



(a) Qwen2.5-3B-Instruct (b) Qwen2.5-14B-Instruct

Figure 4: Experimental results of different sizes of Qwen2.5 models on AIME2024, MATH500, GPQA Diamond, and LiveCodeBench V2.



(a) Llama3.1-8B-Instruct (b) Mistral-7B-V0.3

Figure 5: Experimental results of different model series (Llama3.1-8B-Instruct, Mistral-7B-V0.3) other than Qwen2.5 on AIME2024, MATH500, GPQA Diamond, and LiveCodeBench V2.

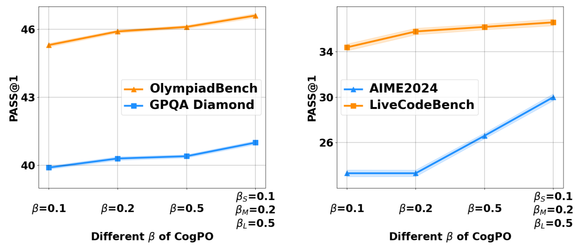


Figure 6: Impact of different β on AIME2024, GPQA Diamond, LiveCodeBench V2 and OlympiadBench.

role. To achieve optimal performance, we recommend processing the data using the complete CRV system.

4.6 Study on Model Scales

To study the effectiveness of different parameter sizes on the student models, we further report the performance of Qwen2.5-3B-Instruct and Qwen2.5-14B-Instruct. The experimental settings are identical to those of Qwen2.5-7B-Instruct. The results are presented in Figure 4. We observe that our method is also effective across different model scales. An interesting observation is that the improvement is more significant in Qwen2.5-14B-Instruct compared to Qwen2.5-3B-Instruct. This is because, even when we leverage the CRV system to rewrite the CoTs, the large capacity gap between the teacher and student models makes it more challenging for Qwen2.5-3B-Instruct to capture the CoTs through SFT. This finding is also consistent with the recently discovered "distillation scaling law" (Busbridge et al., 2025).

4.7 Study on Other Model Backbones

To evaluate the universality of the proposed approach, we perform additional experiments on multiple backbones beyond the Qwen2.5 series on Bespoke-Stratos-17k dataset. Figure 5 demonstrates that, for both LLaMA and Mistral series, our approach achieves notable performance gains over the direct SFT baseline across diverse mathematical and coding tasks. These results indicate that the CRV+CogPO framework enables seamless adaptation to other backbones, demonstrating the universality of our approach on various LLM backbones, which also shows the potential of our work to produce stronger models based on other LLMs.

4.8 Hyper-parameter Analysis

To evaluate the impact of β values in CogPO, we perform a series of experiments with varying β values to assess the algorithm's effectiveness. As shown in Figure 6, the highest performance is attained when assigning tailored β values to samples based on their specific gaps, which is a core principle of the CogPO algorithm.

4.9 Case Studies

Due to space limitations, case studies are shown in the appendix. They clearly show how our approach can effectively expand or simplify the reasoning processes based on the Critic.

5 Conclusion and Future Work

In this paper, we present the CRV framework where we leverage the strengths of LLM agents to critique, refine, and verify CoT outputs for optimizing CoT training sets. The CogPO algorithm further aligns model outputs with their inherent cognitive capacities, improving the performance on several challenging reasoning tasks. In the future, we will i) train and release stronger small models using larger CoT datasets; ii) improve the effectiveness of the CRV framework, especially for much smaller models; and iii) investigate our approach for other domain-specific applications, such as medical diagnosis and legal reasoning.

Limitations

While our proposed framework shows promising results in enhancing the reasoning capabilities of smaller LLMs, several limitations still remain. The CRV framework relies heavily on the contributions of larger models refining the CoT output. This dependency may create challenges in situations where access to larger models is restricted, or these larger models generate incorrect results. In addition, although our framework is designed for smaller LLMs, there remains a ceiling on their performance. By nature, smaller models inherently have reduced capacity to encode complex information and handle nuanced reasoning tasks, which may limit their effectiveness in certain scenarios.

Ethical Considerations

Our work is fully methodological; hence, there are no direct ethical issues. However, smaller models trained on data distilled from larger ones might inherit or exacerbate biased outputs, which can still influence outcomes. We suggest that continuous evaluation of trained LLMs based on ethical guidelines is indispensable.

References

Hritik Bansal, Arian Hosseini, Rishabh Agarwal, Vinh Q. Tran, and Mehran Kazemi. 2024. Smaller, weaker, yet better: Training LLM reasoners via compute-optimal sampling. *CoRR*, abs/2408.16737.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russ Webb. 2025. Distillation scaling laws. *CoRR*, abs/2502.08606.

Huayu Chen, Guande He, Hang Su, and Jun Zhu. 2024. Noise contrastive alignment of language models with explicit rewards. *CoRR*, abs/2402.05369.

Kaiyuan Chen, Jin Wang, and Xuejie Zhang. 2025. Learning to reason via self-iterative process feedback for small language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3027–3042. Association for Computational Linguistics.

Lihu Chen and Gaël Varoquaux. 2024. What is the role of small models in the LLM era: A survey. *CoRR*, abs/2409.06857.

Zeming Chen, Qiyue Gao, and Lawrence S. Moss. 2021. NeuralLog: Natural language inference with joint neural and logical reasoning. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 78–88, Online. 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.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

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, 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.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: model alignment as prospect theoretic optimization. *CoRR*, abs/2402.01306.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 3828–3850. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations*. OpenReview.net.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017. Association for Computational Linguistics.

Linmei Hu, Hongyu He, Duokang Wang, Ziwang Zhao, Yingxia Shao, and Liqiang Nie. 2024. LLM vs small model? large language model based text augmentation enhanced personality detection model. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 18234–18242. AAAI Press.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *CoRR*, abs/2403.07974.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, 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.

Weisen Jiang, Han Shi, Longhui Yu, Zhengying Liu, Yu Zhang, Zhenguo Li, and James T. Kwok. 2024. Forward-backward reasoning in large language models for mathematical verification. In *Findings of the Association for Computational Linguistics, ACL 2024*, pages 6647–6661. Association for Computational Linguistics.

Veronica Latcinnik and Jonathan Berant. 2020. Explaining question answering models through text generation. *arXiv preprint arXiv:2004.05569*.

Chenglin Li, Qianglong Chen, Caiyu Wang, and Yin Zhang. 2023. Mixed distillation helps smaller language model better reasoning. *CoRR*, abs/2312.10730.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022. Explanations from large language models make small reasoners better. *CoRR*, abs/2210.06726.

Zhiming Li, Yushi Cao, Xiufeng Xu, Junzhe Jiang, Xu Liu, Yon Shin Teo, Shang-Wei Lin, and Yang Liu. 2024. Llms for relational reasoning: How far are we? In *LLM4CODE@ICSE*, pages 119–126.

Zhenwen Liang, Ye Liu, Tong Niu, Xiangliang Zhang, Yingbo Zhou, and Semih Yavuz. 2024. Improving LLM reasoning through scaling inference computation with collaborative verification. *CoRR*, abs/2410.05318.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *CoRR*, abs/2305.20050.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *CoRR*, abs/2303.17651.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *CoRR*, abs/2405.14734.

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*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*.

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

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *CoRR*, abs/2303.11366.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2022. Distilling multi-step reasoning capabilities of large language models into smaller models via semantic decompositions. *CoRR*, abs/2212.00193.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073. Association for Computational Linguistics.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online. Association for Computational Linguistics.

Weiqi Wang, Tianqing Fang, Chunyang Li, Haochen Shi, Wenxuan Ding, Baixuan Xu, Zhaowei Wang, Jiaxin Bai, Xin Liu, Cheng Jiayang, Chunkit Chan, and Yangqiu Song. 2024. CANDLE: iterative conceptualization and instantiation distillation from large language models for commonsense reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 2351–2374. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022a. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024a. β-dpo: Direct preference optimization with dynamic β. *CoRR*, abs/2407.08639.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2024b. Self-play preference optimization for language model alignment. *CoRR*, abs/2405.00675.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *Forty-first International Conference on Machine Learning*. OpenReview.net.

Junbing Yan, Chengyu Wang, Taolin Zhang, Xiaofeng He, Jun Huang, and Wei Zhang. 2023. From complex to simple: Unraveling the cognitive tree for reasoning with small language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12413–12425. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *CoRR*, abs/2305.10601.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023b. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. 2024. Advancing LLM reasoning generalists with preference trees. *CoRR*, abs/2404.02078.

Yuanhao Yue, Chengyu Wang, Jun Huang, and Peng Wang. 2024. Distilling instruction-following abilities of large language models with task-aware curriculum planning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6030–6054. Association for Computational Linguistics.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. 2024. When scaling meets LLM finetuning: The effect of data, model and finetuning method. In *The Twelfth International Conference on Learning Representations*. OpenReview.net.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024. Marco-o1: Towards open reasoning models for open-ended solutions. *CoRR*, abs/2411.14405.

Dataset	Size
AIME2024	30
MATH-500	500
GSM8K	1319
GPQA Diamond	198
LiveCodeBench V2	511
MMLU	14042
OlympiadBench (math-en)	674

Table 7: Testing set statistics.

A Supplementary Experiments

A.1 Datasets

In our experiments, we evaluate our work on several benchmarks, including AIME2024[12], MATH-500 (Lightman et al., 2023), GSM8K (Cobbe et al., 2021), GPQA Diamond (Rein et al., 2023), LiveCodeBench V2 (Jain et al., 2024), MMLU (Hendrycks et al., 2021), and OlympiadBench (math-en) (He et al., 2024). The sizes of our testing sets are summarized in Table 7.

For our training set $\mathcal{D}_{\text{SFT}^*}$, we leverage Bespoke-Stratos-17k[13], which contains 17K tuples of questions, reasoning processes, and answers directly distilled from DeepSeek-R1 (DeepSeek-AI, 2025). We also utilize two released CoT datasets to conduct supplementary experiments. The first one is Sky-T1-data-17k[14], which is distilled from QwQ-32B-Preview, whose reasoning abilities are reported to be weaker than those of DeepSeek-R1. The second one is OpenThoughts-114k[15], which is distilled from DeepSeek-R1 and verified using a data curation recipe. We have chosen not to use some previously released CoT datasets (e.g., OpenLongCoT-SFT[16]) due to their significantly weaker reasoning abilities, while some benchmarks (e.g., AIME2024, OlympiadBench) are extremely challenging.

A.2 Experimental Details

In our work, we utilize Qwen2.5-7B-Instruct as the default model backbone and extend our evaluation to Llama3.1-8B-Instruct (Dubey et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), along with other sizes of Qwen2.5 models, to validate the generalizability of our algorithm across diverse model architectures and sizes. We first establish a baseline by assessing the model's zero-shot capabilities. Subsequent experiments leverage this result to quantify the performance improvements attributable to CRV and CogPO. During the CRV phase, the same generation hyperparameters are applied to the Critic, Rethinker, and Verifier for inference: temperature $T = 0.7$, top_p $= 0.9$, and top_k $= 50$. The default backbone is DeepSeek-R1-Distill-Qwen-32B, while we test other backbone choices in the experiments. For CogPO training, the default β settings are: $\beta_{\text{S}} = 0.1$, $\beta_{\text{M}} = 0.2$, and $\beta_{\text{L}} = 0.5$. Training details for model training and baselines are shown in Table 8.

Hyperparameter	Value
CRV Stage	
Batch size	96
Learning rate	1e-5
Learning epoch	3.0
CogPO Stage	
Batch size	96
Learning rate	5e-7
Learning epoch	1.0
SFT (Baseline)	
Batch size	96
Learning rate	1e-5
Learning epoch	3.0
DPO (Baseline)	
Batch size	96
Learning rate	5e-7
Learning epoch	1.0
β	0.1
SimPO (Baseline)	
Batch size	96
Learning rate	5e-7
Learning epoch	1.0
β	2.0
γ	0.3

Table 8: Training hyperparameters.

On the Bespoke-Stratos-17k dataset, for the 3B model, we use a single node with 8 A800 GPUs (80GB), with an approximate training time of 4 hours. For the 7B model, we use a single node with 8 A800 GPUs (80GB), with a training time of about 5 hours. For the 14B model, we use 4 nodes, each with 8 A800 GPUs (80G), resulting in a training time of approximately 14 hours.

[12]https://maa.org/math-competitions/american-invitational-mathematics-examination-aime

[13]https://huggingface.co/datasets/bespokelabs/Bespoke-Stratos-17k

[14]https://github.com/NovaSky-AI/SkyThought

[15]https://huggingface.co/datasets/open-thoughts/OpenThoughts-114k

[16]https://huggingface.co/datasets/SimpleBerry/OpenLongCoT-SFT

Dataset/Model	Zero-shot	SFT	Ours
AIME2024	10.0	<u>16.7</u>	**20.0**
MATH-500	<u>73.6</u>	73.2	**77.0**
GPQA Diamond	<u>33.3</u>	28.8	**36.9**
LiveCodeBench V2	<u>30.7</u>	20.9	**33.3**

Table 9: Performance comparison using Sky-T1-data-17k as the training set.

Dataset/Model	Zero-shot	SFT	Ours
AIME2024	10.0	<u>31.3</u>	**43.3**
MATH-500	73.6	<u>83.0</u>	**88.4**
GPQA Diamond	33.3	<u>42.4</u>	**42.9**
LiveCodeBench V2	30.7	<u>39.9</u>	**46.4**

Table 10: Performance comparison using OpenThoughts-114k as the training set.

A.3 Results on Weaker CoT Dataset

To demonstrate that our approach is truly superior to vanilla SFT over CoT datasets, we conduct an experiment on the Sky-T1 dataset, which is relatively weaker than Bespoke-Stratos-17k due to the choice of the teacher model (i.e., QwQ-32B-Preview) and the data curation pipeline. The results are presented in Table 9. As shown, in some cases, the SFT baseline cannot even beat the zero-shot performance. This negative finding is also consistent with their own blog regarding the model size and data quality[17]. Nonetheless, by comparing our method with the SFT baseline, we can observe clear improvement, which demonstrates that our approach has the efficacy to enhance the reasoning abilities of small models in various scenarios.

A.4 Results on Larger CoT Dataset

We further evaluate the performance of our method using OpenThoughts-114k as the training set, which is much larger than other training sets. This dataset is distilled from DeepSeek-R1 and goes through several quality verification steps. The results are presented in Table 10. It can be seen that our method ultimately exhibits exceptionally strong reasoning performance, significantly surpassing SFT on all benchmarks. This underscores the scalability and generalizability of our CRV+CogPO framework to larger datasets.

A.5 Design Choice of the Critic

An initial, straightforward approach is to employ π_{base} as the Critic. However, owing to the small model's limited reasoning capability, it consistently faces difficulties in distinguishing the difficulty levels of CoTs effectively within our datasets. Note that for CoTs rated as “easy” or “hard”, the CoT is either overly concise (omitting necessary steps) or excessively complex, rendering it unintelligible to the small model and preventing it from following the chain to arrive at the correct answer. Under these circumstances, it is clearly unreasonable to require the small model to classify the CoT difficulty that it cannot comprehend effectively.

Another intuitive CoT evaluation approach is to input the problem and its corresponding CoT into the small model and then verify whether the model can arrive at the correct answer. However, applying this method directly would only partition CoT processes into “correct” or “incorrect” categories. For incorrect CoTs, this binary classification fails to distinguish the root cause of errors (i.e., whether the CoT is overly simplified or overly complex), which is critical for determining appropriate refinement strategies (e.g., expansion for overly simplified processes vs. simplification for overly complex ones).

Consequently, we utilize the larger and stronger LLM used in both the Rethinker and the Verifier (referred to as π_{large}) to act as the Critic. This involves guiding the large model to simulate the cognitive approach of the smaller model, π_{base}. The prompt template of the Critic is shown in Table 13. This setting is akin to educational practices, where professors, instead of students, customarily curate academic content across a spectrum of difficulty levels due to their broader knowledge base. As shown in Table 4, the experiments clearly demonstrate the superior evaluative proficiency of the large model, confirming its advantage in categorizing CoT complexity from the perspective of the smaller model efficiently.

B Case Studies

Case studies are presented in Table 11 and Table 12.

C Prompt Templates

Prompt templates of the Critic, the Rethinker and the Verifier in our CRV system are shown in Table 13.

[17] https://novasky-ai.github.io/posts/sky-t1/

Problem	Find the inverse of matrix $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$
Answer	$A^{-1} = \frac{1}{3}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$
Original reasoning process (correct but unsuitable)	Calculate determinant $\det(A) = 3$, thus $A^{-1} = \frac{1}{3}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$
Extended reasoning process (correct and suitable)	Compute determinant: $2 \times 2 - 1 \times 1 = 3$ Construct adjugate: $\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$ Normalize: $A^{-1} = \frac{1}{3}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$
Incorrect reasoning process	Swap diagonal elements: $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$

Table 11: Case study of how the reasoning process is extended.

Problem	Find the area of a triangle with vertices at $(0, 0)$, $(3, 0)$, and $(0, 4)$
Answer	6
Original reasoning process (correct but unsuitable)	Vector Representation: $\vec{AB} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$, $\vec{AC} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}$ Determinant Method: Area $= \frac{1}{2}\left\|\begin{vmatrix} 3 & 0 \\ 0 & 4 \end{vmatrix}\right\| = \frac{1}{2}(12) = 6$
Simplified reasoning process (correct and suitable)	Recognize right-angled triangle: Base $= 3$, Height $= 4$ Apply elementary formula: Area $= \frac{1}{2} \times$ Base $\times$ Height $= \frac{1}{2} \times 3 \times 4 = 6$
Incorrect reasoning process	Area $= \frac{1}{2}$(Sum of sides) $= \frac{1}{2}(3 + 4 + 5) = 6$

Table 12: Case study of how the reasoning process is simplified.

Role	Prompt Template
Critic	You are a highly capable evaluator. Your task is to assess the given reasoning process from the perspective of a small language model (e.g., 7B). Specifically, determine whether the reasoning process provides sufficient detail for a small model to solve the problem, or whether it is too terse (i.e., lacking critical details) or too complex (i.e., containing unnecessary or confusing steps). Complexity Definitions (from the perspective of a small model): - Easy: The reasoning process is overly terse; it omits essential details that a small model needs to solve the problem. - Medium: The reasoning process is appropriately balanced, offering enough detailed guidance. - Hard: The reasoning process is overly complex, with extraneous or convoluted steps that could hinder a small model to follow it. Output Format: You must output exactly one word: easy, medium, or hard.
Rethinker (easy)	You are a helpful assistant who is highly skilled at extending reasoning processes. Given a problem ,its correct answer and its terse reasoning process, your task is to extend the reasoning process by adding necessary details and intermediate steps so that a small language model (e.g., a 7B model) can follow the extended reasoning process to solve the problem. You should add necessary steps and details based on the correct answer. You must output ONLY the extended reasoning process with no additional explanation or commentary.
Rethinker (hard)	You are a helpful assistant who is highly skilled at simplifying reasoning processes. Given a problem, its correct answer and its overly complex reasoning process, your task is to simplify the reasoning process so that a small language model (e.g., a 7B model) can reliably follow the steps to solve the problem. You should remove redundancies or use simpler method on the basis of correct answer. You must output ONLY the simplified reasoning process with no additional explanation or commentary.
Verifier	You are a highly capable Verifier. Your task is to assess a given reasoning process based on a problem and its correct answer. Specifically, determine whether the reasoning process is sufficient and accurate for you to reach the correct answer. If the reasoning process correctly guides you to derive the the correct answer, output YES. If the reasoning process fails to guide you to the correct answer, output NO. You must output exactly one word: YES or NO.
Rethinker (incorrect thought)	You are an assistant who is skilled at converting correct reasoning processes to incorrect reasoning processes. Given a problem ,its answer and its correct reasoning process, your task is to corrupt the correct reasoning process by introducing logical fallacies and misleading steps, so that a small language model (e.g., a 7B model) cannot follow the incorrect reasoning process to solve the problem. You must output ONLY the incorrect reasoning process with no additional explanation or commentary.

Table 13: Prompt templates for the CRV+CogPO framework.