PSPO*: A Dual-Dimensional Nonlinear Process-supervised Policy Optimization for Reasoning Alignment

Anonymous ACL submission

Abstract

Process supervision enhances the performance of large language models (LLMs) in reasoning tasks by providing feedback at each step of chain-of-thought reasoning. However, even advanced LLMs are prone to redundant reasoning due to the lack of effective process supervision methods. We claim that the effectiveness of process supervision significantly depends on both the accuracy and the length of reasoning chains. Moreover, we identify that these factors exhibit a nonlinear relationship with the overall reward score of the reasoning process. Based on this, we propose a dual-dimensional nonlinear process supervision method, named PSPO*, which systematically outlines the workflow from reward model training to policy optimization, and highlights the importance of nonlinear rewards in process supervision. Based on PSPO*, we develop the PSPO-WRS, which considers the number of reasoning steps in determining reward scores and utilizes an adjusted Weibull distribution for nonlinear reward shaping. Experimental results on mathematical reasoning datasets demonstrate that PSPO-WRS consistently outperforms current mainstream models.

1 Introduction

Large language models (LLMs) have shown promising development in solving tasks that require complex reasoning, particularly in mathematical problems (Shao et al., 2024; Li et al., 2024; Yang et al., 2024c). Studies have shown that an effective reasoning process can significantly improve a model's performance on downstream tasks. Conversely, the unreliable reasoning process can mislead the model and produce incorrect results (Wang et al., 2023a; Jin et al., 2024). Therefore, quantifying an accurate reasoning process is crucial for effectively addressing the complex reasoning task.

Current approaches to enhance reasoning capabilities primarily divide into supervised fine-tuning (SFT)-based and reinforcement learning (RL)-based methods. Conventional SFT methods rely on manually curated (Chern et al., 2023) or knowledge-distillation (Yang et al., 2024a) reasoning datasets, yet their effectiveness is fundamentally constrained by the quality and diversity of training data, often resulting in limited generalization capabilities. In contrast, RL-based approaches have shown remarkable success in developing advanced reasoning capabilities (Jiang et al., 2024; Min et al., 2024). The OpenAI o1 framework achieved doctoral-level mathematical reasoning through large-scale RL training (Lightman et al., 2024), while DeepSeek-R1-Zero revealed that pure RL optimization without SFT can produce competitive reasoning performance (DeepSeek-AI et al., 2025). These breakthroughs establish RL as the primary method for creating powerful reasoning capabilities in LLMs.

Recent advancements in process-supervised RL have demonstrated promising potential for achieving reasoning alignment through quantitative process evaluation (Lightman et al., 2023; Luo et al.,

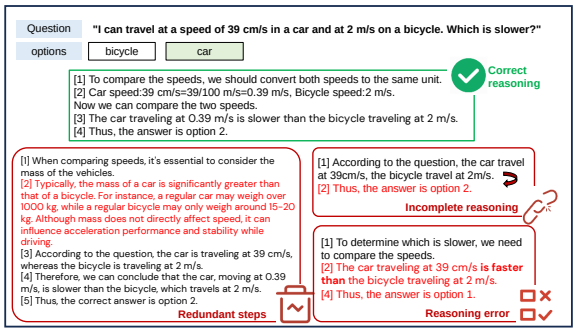


Figure 1: An example from the QQA dataset. The reasoning error solution has an error in step[2] where the model confuses the concept of time period and time point, resulting in a wrong answer. The incomplete reasoning solution simply jumps to the final answer after summarizing the problem, which is incomplete and unreasonable. And the redundant steps generate too much noise.

2023; Liang et al., 2024). This approach typically involves training a process reward model (PRM) that aggregates scores from multiple reasoning chains to produce a unified assessment (Uesato et al., 2022b; Lightman et al., 2023). However, existing process supervision methods solely focus on the accuracy dimension of reasoning chains while overlooking other critical factors such as reasoning chain-length (Lightman et al., 2023). As illustrated in Figure 1, reasoning chains that are inaccurate, redundant (i.e., excessively long), or incomplete (i.e., insufficiently short) may result in erroneous outcomes. These observations motivate the development of a multi-dimensional process supervision framework that jointly optimizes the accuracy and length of reasoning chains.

In this work, we propose Process-Supervised Policy Optimization (PSPO*), a novel method for process supervision in LLMs. In PSPO*, we innovatively propose that the accumulation function $\mathcal{F}$ needs to consider both reasoning accuracy (α) and reasoning chain length (l) dimensions to compute reward scores, i.e., $\mathcal{F} \sim (\alpha, l)$. Furthermore, our analysis reveals that reasoning steps at different positions contribute distinctly to the final reasoning outcome (e.g., initial steps that restate the problem have a limited impact), and reasoning chains typically maintain a reasonable range. We propose incorporating this prior knowledge into the final reward computation through nonlinear reward shaping.

To validate the effectiveness of the proposed methods, we instantiate the PSPO* method with adjustable Weibull distribution reward shaping (WRS), termed PSPO-WRS. In PSPO-WRS, to comprehensively consider the impact of both reasoning accuracy and chain length on the overall reward, we compute the final reward score by multiplying individual step rewards and normalizing them with respect to the number of steps (eq. 5). Furthermore, we leverage prior knowledge from process supervision to construct an adjustable Weibull distribution, which is integrated into reward shaping to enhance the nonlinear characteristics of reward scores (eq. 6). Experimental results demonstrate that PSPO-WRS achieves superior performance across various datasets, validating the effectiveness of the PSPO* paradigm.

Our main contributions are as follows:

- We propose a dual-dimensional nonlinear process-supervised policy optimization method, PSPO*, that considers both reasoning accuracy and chain length in its accumulation function for reward computation.
- We introduce nonlinear reward shaping to incorporate reasoning-related prior knowledge into reward computation, including position-dependent step importance and reasonable chain length constraints.
- We develop PSPO-WRS, a concrete implementation of PSPO* using Weibull distribution reward shaping, which achieves superior performance across multiple datasets and validates our hypothesis about reward nonlinearity in reasoning alignment.

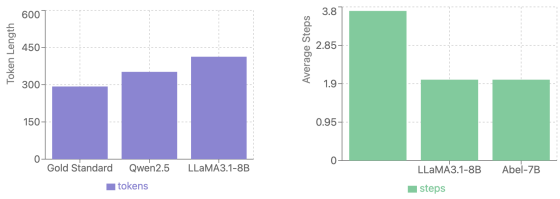
2 Related Works

2.1 LLM Alignment Techniques

LLMs have demonstrated remarkable reasoning capabilities (Yang et al., 2023; Dubey et al., 2024a; Yang et al., 2024b; DeepSeek-AI et al., 2025), yet they still face challenges such as misunderstanding instructions, making logical errors, and providing inaccurate information. This has made LLM alignment a critical research focus (Wang et al., 2023b). The traditional Reinforcement Learning from Human Feedback (RLHF) framework (Ouyang et al., 2022) involves reward learning from human feedback followed by policy optimization using PPO (Schulman et al., 2017). To address RLHF's complexity and instability, Direct Preference Optimization (DPO) (Rafailov et al., 2023) was introduced, simplifying the process through a classification loss. Recent works like Reinforced Token Optimization (RTO) (Zhong et al., 2024) have enhanced the framework with token-wise rewards, while ΨPO (Azar et al., 2024) revealed potential overfitting issues in both RLHF and DPO due to their reliance on ELo-score assumptions. In response to these limitations, we propose the PSPO* method, which incorporates step-level pointwise rewards and policy optimization for process supervision.

2.2 Process-based Reasoning Supervision

Recent advances in LLMs have shown significant improvements in multi-step reasoning tasks through approaches like Chain of Thought (CoT) and Tree of Thought (ToT) (Cobbe et al., 2021; Wei et al., 2023; Yao et al., 2023). These methods enhance reasoning abilities by decomposing complex problems into manageable steps, particularly



(a) Average token length on GSM8K.

(b) Average step count on AwpNLI.

Figure 2: Analysis of reasoning chain length on GSM8K and AwpNLI datasets. Token length analysis on the GSM8K dataset shows increased verbosity in LLM outputs compared to gold standard solutions. Analysis of reasoning steps on AwpNLI dataset revealing reduced step count in LLM solutions compared to human annotations.

effective in mathematical reasoning (Kojima et al., 2023). Research by Uesato et al. (2022a) and Lightman et al. (2023) introduced process-supervised reward models, demonstrating their necessity in ensuring correct reasoning steps and preventing false positives. Step-DPO (Lai et al., 2024) further refined this approach by optimizing individual reasoning steps, though it requires extensive supervised data. To address this limitation, Zhang et al. (2024) developed ReST-MCTS*, integrating process reward guidance with tree search for higher-quality reasoning traces. Building upon these advances, we propose PSPO-WRS, which implements nonlinear reward shaping using adjusted Weibull distribution to better identify and reinforce critical reasoning behaviors.

3 PSPO*: An Effective Method for Process Supervision

3.1 Motivation and Overview

Process supervision proposed by Lightman et al. (2023), aims to improve LLMs' reasoning capabilities by rewarding the model for generating accurate intermediate reasoning steps. However, existing methods focus exclusively on the accuracy of reasoning steps while overlooking the impact of reasoning chain length. For instance, redundant reasoning chains that repeatedly restate previous steps can still receive high rewards under current accuracy-focused approaches, despite being suboptimal. Our empirical studies demonstrate that this single-dimensional optimization causes models to generate either redundant or incomplete reasoning chains.

On the GSM8K dataset, we observe a significant increase in reasoning chain length after training Qwen2.5 and LLaMA3.1-8B using traditional PSM methods. While the gold standard solutions average 292.9 in chain length, Qwen2.5 generates longer chains averaging 351.9, and LLaMA3.1-8B produces chains with 412.9 in length, as shown in 2(a). Human analysis reveals that this increased length primarily stems from two patterns: unnecessary repetition of the problem statement and redundant restatement of previous reasoning steps.

The issue of shortened reasoning chains becomes evident in our experiments on the AwpNLI dataset. After training LLaMA3.1-8B and Abel-7B using traditional PSM methods, both models exhibit reduced reasoning steps. As illustrated in 2(b), human-annotated solutions contain an average of 3.7 reasoning steps (113 tokens), whereas LLaMA3.1-8B and Abel-7B typically generate only 2 steps (43 and 52 tokens, respectively). Human analysis shows that models often simply restate the problem and jump directly to the answer. (Detailed experimental settings are provided in Appendix B)

These findings highlight a critical limitation in current process-supervised methods: the lack of consideration for reasoning chain length results in models producing either excessively long or overly short reasoning chains. To address these issues, we propose PSPO*, which explicitly incorporates both accuracy and reasoning chain length as optimization objectives.

3.2 The PSPO* Algorithm

3.2.1 Process Supervision Preliminary

The process supervision based on human feedback primarily consists of two stages: learning the reward model and optimizing the policy based on the learned reward model (Azar et al., 2024).

In the process of training the reward model, annotators need to determine whether each reasoning step is negative, neutral, or positive, and correspondingly select from [-1, 0, 1] (Lightman et al., 2023; Ma et al., 2023). These annotated data are then used to train the reward model to accurately classify the quality of reasoning steps. These annotated data are then used to train the reward model, which will output a reward score R^k for the k-th reasoning step during the PPO training process. A detailed description of the reward model training process is provided in Appendix A.

3.2.2 Dual-dimension Accumulation Function

The score R_k for the current k-th reasoning step only reflects the quality of that individual step and not the whole reasoning process. In process supervision, the reward for the whole reasoning process can only be evaluated by accumulating the scores of all reasoning steps. We define the overall reward score for the whole reasoning process as $R(x, y)$, and let $\mathcal{F}$ be the accumulation function. In previous studies, the construction of the reward function typically only considered the impact of accuracy for the overall reward score $R(x, y)$ in the reasoning chains (Lightman et al., 2023). For instance, Lightman et al. (2023) proposed that using the product of the reward scores for each reasoning step as the accumulation function $\mathcal{F}$, thereby modeling the overall reward score for the entire reasoning process, as follows:

$$R(x, y) = \prod_{j=1}^{t} P(y^j = 1 | x^j, y^j_{pre}).$$

However, when the number of reasoning steps is not fixed, the overall reward score is influenced by the number of reasoning steps. As the correctness probability is decimal, the more steps involved in reasoning, the smaller the product of probabilities, resulting in lower rewards, which leads to a tendency for the policy to subsequently generate fewer reasoning steps.

Our contribution lies in proposing that, the accumulation function $\mathcal{F}$ should simultaneously account for both the accuracy and the length of reasoning chains in process supervision. Specifically, we define the length of reasoning chains by the number of steps in the reasoning process, then:

$$R(x, y) = \mathcal{F}(R^1, R^2, \cdots, R^t), \quad (1)$$

where t denotes the total number of reasoning steps. Through the accumulation function $\mathcal{F}$ in Equation 1, we calculate the final reward by jointly considering both the accuracy and length of the reasoning chain.

3.2.3 Non-linear Reward Shaping

The objective of process supervision is to optimize the policy function $\pi \in \Delta(x, y)$ through the overall reward score $R(x, y)$ of the reasoning process, thereby maximizing the expected reward. Simultaneously, it aims to minimize the KL divergence between π and the reference policy $\pi_{ref} \in \Delta(x, y)$:

$$J(\pi) = \mathbb{E}_\pi[R(x, y) - \beta D_{KL}(\pi \parallel \pi_{ref})], \quad (2)$$

where β is a hyperparameter used to limit the difference between the new and reference policies, balancing the exploration and exploitation of the policy.

A key contribution of our work is the introduction of nonlinear reward shaping to refine the accumulation function. In process supervision, to enable the policy to better distinguish critical behaviors, we propose to apply *nonlinear reward shaping*. Nonlinear functions allow us to assign different weights to reasoning steps based on their relative importance. For example, the first reasoning step, which typically restates the problem, may have high accuracy but contributes less to the final score, thus deserving a lower weight. Conversely, critical reasoning steps that significantly impact the final outcome should receive higher weights. The final policy optimization is:

$$J(\pi) = \mathbb{E}_\pi[R_s R(x, y) - \beta D_{KL}(\pi \parallel \pi_{ref})], \quad (3)$$

where R_s is a nonlinear function used for reward shaping. Specifically, the method proposed by Lightman et al. (2023) can be viewed as a simplified version of the PSPO* paradigm. In their method, the value for reward shaping is specifically set to 1, and the construction of the accumulation function does not take into account the impact of the length of reasoning chains on the reward score.

In the next section, we utilize the prior knowledge from process supervision to perform nonlinear reward shaping using the adjusted Weibull distribution, demonstrating the validity of this view.

4 PSPO-WRS: Process-supervised Policy Optimization with Nonlinear Reward Shaping

In process supervision, there is a nonlinear relationship between the number of reasoning steps and the overall reward score. The goal of the CoT reasoning is typically to solve the problem while minimizing computational complexity (Wei et al., 2022). Fewer reasoning steps imply higher efficiency, but this does not always correlate with higher accuracy or correctness. Conversely, a reasoning process with more steps might achieve greater accuracy, but at the cost of lower efficiency. Based on this prior knowledge, we employ the Adjusted Weibull distribution to shape the rewards for the number of reasoning steps. The reward shaping function is as follows:

$$R_s = C * \frac{k}{\lambda}(\frac{t}{\lambda})^{k-1} e^{-(t/\lambda)^k}, \quad (4)$$

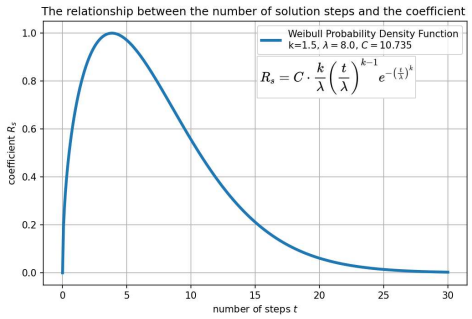


Figure 3: The adjusted Weibull distribution. Prameter settings are: $C = 10.735$, $k = 1.5$, and $\lambda = 8.0$.

where C is a constant coefficient used to adjust the overall reward score, λ is the scale parameter, which determines the spread of the distribution, and k is the shape parameter, which dictates the shape of the distribution.

Additionally, for the accumulation function $\mathcal{F}$, to eliminate the linear trend and account for the number of reasoning steps, we standardized the step count based on the method proposed by Lightman et al. (2023), specifically:

$$\mathcal{F} = [\prod_{j=1}^{t} P(y^j = 1|x^j, y^j_{pre})]^{1/t}. \quad (5)$$

Finally, we propose process supervision based on adjusted Weibull Reward Shaping (PSPO-WRS):

$$J(\pi) = \mathbb{E}_\pi[R_s\mathcal{F} - \beta D_{KL}(\pi \parallel \pi_{ref})]. \quad (6)$$

The PSPO-WRS introduces nonlinear reward shaping, integrating both the accuracy and the length of reasoning chains into process supervision. In the experimental section, we will demonstrate the effectiveness of PSPO-WRS.

5 Experimental Results

5.1 Experimental Setups

Datasets For training the Reward Model, we utilize the PRM-800K dataset (Lightman et al., 2023). Upon analysis, we observe that the dataset is predominantly composed of steps labeled as "1". To address this data imbalance, we employ an oversampling strategy where steps labeled as "0" and "-1" are duplicated 2-3 times to ensure a balanced distribution. For comprehensive evaluation of model capabilities, we employ GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), AIME24, and GPQA (Rein et al., 2023), CEval (Huang et al., 2023), MMLU (Hendrycks et al., 2021a) datasets.

Metrics and Parameters setting We systematically evaluated the performance of our proposed approach across all benchmark datasets through the OpenCompass (Contributors, 2023) evaluation framework. We employ Llama3.1-8B (Dubey et al., 2024b), Qwen2.5-7B (Yang et al., 2024b) and DeepSeek-MATH-7b-base (Shao et al., 2024) as the backbone models. The reward model is trained on the BERT-large (Devlin et al., 2019) due to its proven efficacy in classification tasks (Gao et al., 2023). We trained the reward model over 3 epochs with a learning rate of 2e-5, a warmup rate of 0.05, and a maximum sequence length of 1024. PPO training uses Lora (Hu et al., 2022) with a learning rate of 1.41e-5 and a maximum of 1024 tokens. On 5000 entries, each epoch averages 55 hours on four NVIDIA A100 GPUs. In our PSPO-WRS method, the parameters are set as follows: $C = 10.735$, $k = 1.5$, and $\lambda = 8.0$. The function distribution is illustrated in Figure 3.

5.2 Overall Results

Main Results. We present our main experimental results in Table 1, where we evaluate our proposed PSPO-WRS method across three different backbone models (Llama3.1-8B (Dubey et al., 2024b), Qwen2.5-7B (Yang et al., 2024b) and DeepSeek-MATH-7b (Shao et al., 2024)) on six diverse datasets. The evaluation datasets consist of four mathematical reasoning benchmarks (GSM8K, MATH, GPQA, and AIME24) and two general knowledge benchmarks (CEval and MMLU).

As shown in Table 1, our PSPO-WRS method demonstrates consistent improvements, particularly on the MATH benchmark across all backbone models. Taking DeepSeek-MATH as an example, our method achieves a 16.24% absolute improvement (from 11.10% to 27.34%) on MATH compared to the backbone model. Similar substantial gains on MATH are observed with Llama3.1 (11.16% improvement) and Qwen2.5 (17.82% improvement from base model), demonstrating the effectiveness of our approach in enhancing complex mathematical reasoning capabilities.

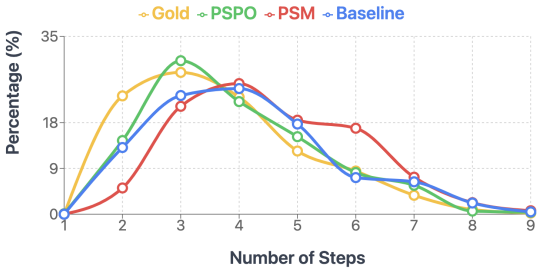
When compared to the process-supervised method (PSM) Lightman et al. (2023), PSPO-WRS shows competitive results across different benchmarks. While both methods demonstrate improvements over their respective base models, PSPO-WRS exhibits particularly strong performance on complex mathematical reasoning tasks, especially the MATH benchmark. For instance, DeepSeek-

Models	MATH	AIME24	GSM8K	GPQA	CEval	MMLU	Average
Llama3.1-8B	7.56%	0/30	56.41%	8.08%	45.5%	27.6%	24.19%
Llama3.1-PSM	16.06%	**1/30**	70.89%	9.09%	**48.6%**	**28.7%**	29.45%
Llama3.1-PSPO	**18.72%**	**1/30**	**71.19%**	**10.10%**	47.4%	27.8%	**29.76%**
Qwen2.5-7B	8.98%	3/30	**79.68%**	20.20%	**67.2%**	**61.8%**	**41.31%**
Qwen2.5-PSM	23.51%	**4/30**	74.38%	20.71%	48.2%	42.2%	37.06%
Qwen2.5-PSPO	**26.80%**	**4/30**	72.78%	**21.21%**	49.7%	44.2%	38.00%
DeepSeek-MATH-7B	11.10%	0/30	76.42%	**16.16%**	52.0%	**27.8%**	30.58%
DeepSeek-MATH-PSM	21.50%	1/30	**81.73%**	8.59%	52.0%	27.7%	32.48%
DeepSeek-MATH-PSPO	**27.34%**	**2/30**	78.17%	12.12%	**52.1%**	26.9%	**33.88%**

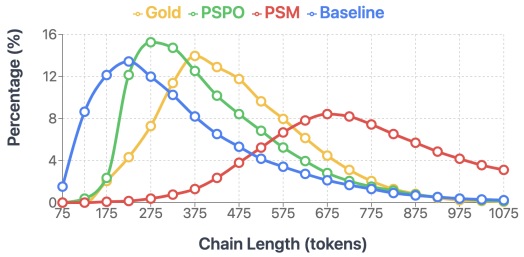
Table 1: Performance comparison on mathematical reasoning benchmarks. PSM denotes models trained with the process-supervised reinforcement learning method proposed by Lightman et al. (2023), while PSPO represents models trained with our PSPO-WRS method. Both methods are built upon their respective base models. All experimental results are obtained using OpenCompass prompts in our independent evaluation.

Method	Avg Steps (Steps)	ΔSteps (vs. Gold)	Avg Length (tokens)	ΔLength (vs. Gold)
Gold	3.7	-	292.9	-
Baseline	4.8	+1.1	351.9	+59.0
PSM	4.7	+1.0	920.6	+627.7
PSPO	4.3	+0.6	365.6	+73.0

Table 2: Comparison of reasoning chain steps and length across different method variants on GSM8K. Δ represents the absolute difference from the Gold standard. Lower values indicate better alignment with Gold.



(a) Distribution of number of reasoning steps.



(b) Distribution of reasoning chain lengths.

Figure 4: Distribution analysis of reasoning steps and reasoning chain lengths.

MATH-PSPO achieves 27.34% on MATH, outperforming DeepSeek-MATH-PSM's 21.50%. On GSM8K, while PSM shows slightly higher scores in some cases, PSPO-WRS maintains competitive performance while generating more concise and reasonable-length reasoning chains (which will be empirically demonstrated in the following experimental analysis). This suggests that our dual-dimensional optimization strategy effectively enhances mathematical reasoning capabilities while promoting more appropriate reasoning processes.

On general knowledge benchmarks (CEval and MMLU), PSPO-WRS maintains performance comparable to the baseline models, with slight variations across different backbones. This indicates that our method's focus on optimizing mathematical reasoning does not significantly impact the model's general knowledge capabilities.

Analysis of Reasoning Chain Length. To analyze whether PSPO-WRS helps models generate more reasonable-length reasoning chains, we examine the reasoning chains produced by Qwen2.5-7B (baseline), Qwen2.5-PSM (PSM), and Qwen2.5-PSPO (PSPO) on the GSM8K dataset [1]. We compare their reasoning chains with the standard solutions from the dataset in terms of the number of reasoning steps and chain length.

As shown in Figure 4(a), the distribution of reasoning steps reveals that PSPO generates solutions with step counts more closely aligned with the standard solutions. The baseline and PSM models both show tendencies toward generating redundant reasoning chains, though to different degrees.

[1]The standard solution is provided in the GSM8K dataset from OpenCompass.

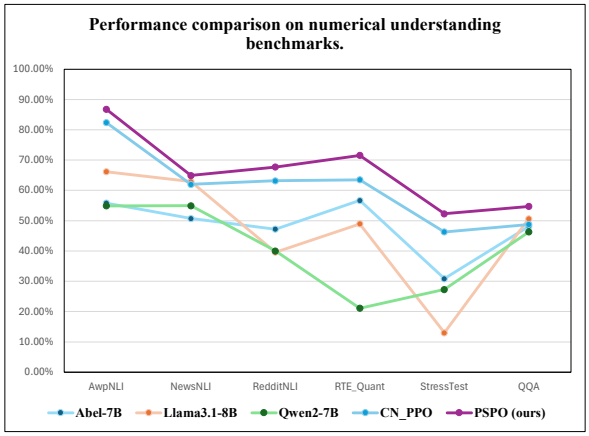


Figure 5: Performance comparison on numerical understanding benchmarks. Our PSPO-WRS method, built upon Abel-7B, consistently outperforms baseline models across all six datasets that test LLMs' numerical sensitivity. See Appendix C.3 for detailed numerical results.

Figure 4(b) demonstrates the length differences between generated chains and standard solutions, where PSPO shows the most similar distribution to the standard solutions, indicating its generated chain lengths consistently match the standard solutions more closely across all problems.

The quantitative results in Table 2 further support these observations. PSPO generates chains averaging 4.3 steps, differing by only 0.6 steps from the gold standard solutions (3.7 steps), while baseline and PSM models show larger deviations of 1.1 and 1.0 steps, respectively. In terms of chain length, PSPO's reasoning chains (365.6 tokens) show a moderate deviation of 73.0 tokens from gold standard solutions (292.9 tokens), significantly better than PSM which deviates by 627.7 tokens, and comparable to the baseline's deviation of 59.0 tokens. Notably, although the baseline model appears to have a closer average length to the gold standard, its length distribution tends to skew towards shorter chains, indicating that the baseline model is more prone to generating incomplete reasoning steps.

These comprehensive analyses confirm that PSPO effectively guides the model to generate reasoning chains with more reasonable lengths.

5.3 Extension to Numerical Reasoning

To further validate the effectiveness of PSPO-WRS beyond mathematical reasoning tasks, we extend our evaluation to a broader range of tasks that require numerical understanding. We conduct experiments on six datasets that focus on numerical sensitivity in natural language understanding proposed by Chen et al. (2023): AwpNLI, NewsNLI, RedditNLI, RTE-Quant, StressTest, and QQA. Detailed descriptions of these datasets and experimental settings can be found in Appendix C.

Figure 5 presents the performance comparison between our PSPO-WRS and several strong baseline models. Building upon the Abel-7B (Chern et al., 2023), PSPO demonstrates superior performance on AwpNLI, NewsNLI, RedditNLI, RTE-Quant, StressTest, and QQA benchmarks. The consistent improvements in these six datasets demonstrate that our process-supervised optimization approach is effective not only in mathematical logical reasoning tasks but also in numerical sensitivity related reasoning tasks.

These results demonstrate that PSPO's benefits extend beyond traditional mathematical reasoning tasks to broader numerical understanding scenarios, suggesting its potential as a general approach for enhancing models' numerical reasoning capabilities.

PSPO-WRS exhibits exceptional performance even when compared with ultra LLMs. We conducted a comparative analysis of PSPO-WRS against mainstream ultra LLMs, as detailed in Figure 6. Across all evaluated datasets, the PSPO-WRS significantly outperformed GPT-3.5 (Ouyang et al., 2022). Relative to GLM4 (Zeng et al., 2024), our model showed slightly weaker performance on the NewsNLI dataset, yet exhibited superior performance on other datasets. Against the more robust reasoning capabilities of Qwen2-72B (Yang et al., 2024a), PSPO-WRS also showed its strengths in the AWPNLI dataset and demonstrated comparable performance on additional datasets. Notably, although the baseline model Abel-7B (Chern et al., 2023) of PSPO-WRS is far outperformed by these ultra LLMs in terms of raw performance, our process supervision method effectively bridges this gap, showcasing its efficacy.

5.4 Ablation Analysis

It is necessary to incorporate the length of reasoning chains into process supervision through nonlinear rewards. Our ablation study confirms that process supervision depends not only on the accuracy of the reasoning chain but also on its length, and introduces nonlinear rewards accordingly. Further analysis of Figure 7 and Figure 8 reveals that without nonlinear rewards, the probability of the

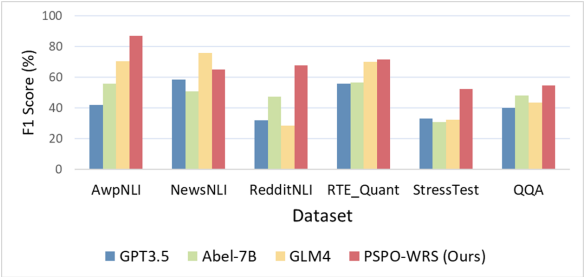


Figure 6: The results compared with ultra LLMs. It is noteworthy that our model outperforms ultra LLMs in most scenarios with only 7B parameters.

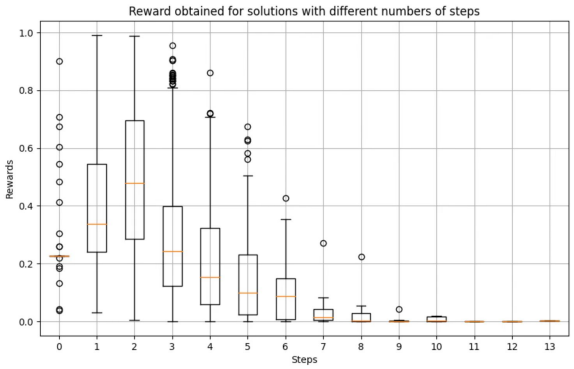


Figure 7: The relationship between the length of reasoning chains and rewards when nonlinearity is not incorporated into the reward scores of the reasoning process.

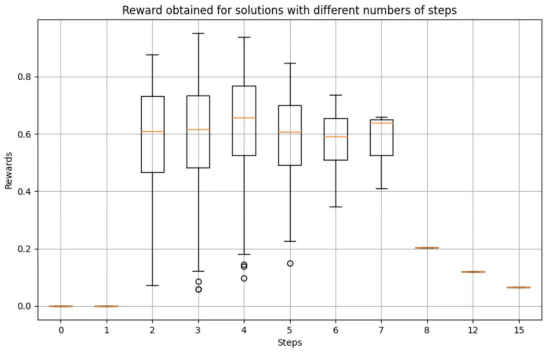


Figure 8: The relationship between the length of reasoning chains and rewards when nonlinearity is incorporated into the reward scores of the reasoning process.

policy generating a particular reasoning process significantly decreases as the number of steps increases. Additionally, the reward scores for higher reasoning steps also diminish. However, the incorporation of nonlinear rewards mitigates this phenomenon.

Figure 7 demonstrates that without nonlinear rewards, the average reward score for reasoning chains declines once the number of steps exceeds three. This decline suggests that overlooking step count in process supervision can reduce overall reward scores, even when each step is accurately executed, due to simple multiplicative effects. Consequently, models may favor generating shorter reasoning processes.

Conversely, as shown in Figure 8, after introducing nonlinear rewards, although the model still tends to generate multiple three-step reasoning processes, the proportion of reasoning processes with more steps has significantly increased. This phenomenon aligns with the prior knowledge incorporated during the reward shaping process. Furthermore, the model consistently yields high-scoring reasoning processes across different step counts, demonstrating its adaptability to tasks with variable reasoning lengths.

Process supervision is nonlinear. To understand the impact of our nonlinearity module, we conducted an ablation study by comparing PSPO with process-supervised RL, as shown in Table 1. Note that PSPO degenerates to PRM when the nonlinearity module is removed since process-supervised RL (Lightman et al., 2023) can be viewed as a special case of our method. The comparison demonstrates that incorporating nonlinear rewards consistently improves performance across all evaluated datasets. For instance, on the MATH dataset, the nonlinearity module brings improvements of 2.66%, 3.29%, and 5.84% for Llama3.1, Qwen2.5, and DeepSeek-MATH respectively. These results highlight that nonlinear reward modeling plays a crucial role in effective process supervision.

6 Conclusion

In this paper, we substantiate the critical role of accuracy and length of reasoning chains in enhancing process supervision and show that reasoning-related prior knowledge can benefit the reasoning chains. Inspired by these insights, we propose a novel process supervision method, PSPO*, which incorporates both accuracy and length into process supervision computation through accumulation functions, while leveraging nonlinear reward shaping to encode reasoning-related prior knowledge. As an instantiation of the PSPO* method, we introduce PSPO-WRS, which leverages an adjusted Weibull distribution for nonlinear reward shaping. The experimental results confirm our hypothesis and demonstrate that our method enables various LLMs to generate more accurate reasoning chains with appropriate lengths and shows consistent effectiveness across different reasoning tasks.

Limitations and Future Works

Alternative Constructions of Nonlinear Functions The current implementation of PSPO utilizes an adjusted Weibull distribution function as its instantiation, which has demonstrated promising results across various reasoning tasks. However, this represents only one possible formulation among numerous potential mathematical functions. The choice of instantiation function is critical as it directly influences the optimization dynamics and the resulting reasoning behavior. Future research could systematically explore alternative functional forms to potentially discover more optimal implementations of PSPO.

Automatic Data-Driven Prior Knowledge Modeling Another limitation lies in our current approach to incorporating prior knowledge about reasoning step importance. While we recognize that reasoning steps at different positions contribute differently to the final outcome (e.g., problem restatement steps having a limited impact) and that there exists a reasonable range for chain length, our current method relies on manually designed prior distributions. This manual design process may not generalize well across diverse reasoning datasets, as different tasks may exhibit distinct patterns in terms of optimal reasoning step distribution and importance. Future work should explore automated methods to learn and model these prior distributions directly from specific datasets to capture dataset-specific reasoning patterns more effectively.

Ethics Statement

This work focuses on improving the reasoning process of large language models through process supervision and does not present any increased risks of harm beyond the existing norms of language model research. The associated risks include the potential for models to generate inaccurate reasoning chains, which we explicitly address through our multi-dimensional supervision approach. While our method aims to enhance reasoning capabilities, we acknowledge that the underlying language models may still contain inherent biases from their pre-training data. However, such concerns are mitigated in our work as we primarily focus on quantitative reasoning tasks with verifiable solutions, rather than open-ended generation tasks that could potentially produce harmful content.

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

Ethan Chern, Haoyang Zou, Xuefeng Li, Jiewen Hu, Kehua Feng, Junlong Li, and Pengfei Liu. 2023. Generative ai for math: Abel. https://github.com/GAIR-NLP/abel.

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.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao

Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, 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, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel 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, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Hol-

land, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024a. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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. 2024b. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. 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.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

Jinhao Jiang, Zhipeng Chen, Yingqian Min, Jie Chen, Xiaoxue Cheng, Jiapeng Wang, Yiru Tang, Haoxiang Sun, Jia Deng, Wayne Xin Zhao, Zheng Liu, Dong Yan, Jian Xie, Zhongyuan Wang, and Ji-Rong Wen. 2024. Technical report: Enhancing LLM reasoning with reward-guided tree search. *CoRR*, abs/2411.11694.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. *CoRR*, abs/2401.04925.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *CoRR*, abs/2406.18629.

Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, , Bin Xu, Ren Bowen, Chong Feng, Yang Gao, and Heyan Huang. 2024. Fundamental capabilities of large language models and their applications in domain scenarios: A survey. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11116–11141, Bangkok, Thailand. Association for Computational Linguistics.

Xinyue Liang, Jiawei Li, Yizhe Yang, and Yang Gao. 2024. Bit_numeval at SemEval-2024 task 7: Enhance numerical sensitivity and reasoning completeness for quantitative understanding. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1830–1841, Mexico City, Mexico. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison 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.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *CoRR*, abs/2308.09583.

Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. 2023. Let's reward step by step: Step-level reward model as the navigators for reasoning. *CoRR*, abs/2310.10080.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *CoRR*, abs/2412.09413.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. 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 15991–16111. Association for Computational Linguistics.

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*.

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, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

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.

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 fine-tuned chat models. *CoRR*, abs/2307.09288.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022a. Solving math word problems with process- and outcome-based feedback. *Preprint*, arXiv:2211.14275.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022b. Solving math word problems with process- and outcome-based feedback. *CoRR*, abs/2211.14275.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. 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.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023b. Aligning large language models with human: A survey. *CoRR*, abs/2307.12966.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. 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, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024c. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122.

Yizhe Yang, Huashan Sun, Jiawei Li, Runheng Liu, Yinghao Li, Yuhang Liu, Heyan Huang, and Yang Gao. 2023. Mindllm: Pre-training lightweight large language model from scratch, evaluations and domain applications. *CoRR*, abs/2310.15777.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. 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, New Orleans, LA, USA, December 10 - 16, 2023*.

Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from GLM-130B to GLM-4 all tools. *CoRR*, abs/2406.12793.

Dan Zhang, Sining Zhoubian, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: LLM self-training via process reward guided tree search. *CoRR*, abs/2406.03816.

Han Zhong, Guhao Feng, Wei Xiong, Li Zhao, Di He, Jiang Bian, and Liwei Wang. 2024. DPO meets PPO: reinforced token optimization for RLHF. *CoRR*, abs/2404.18922.

A Detailed Process of Reward Model Training

In the process of training the outcome-supervised reward models (ORMs), annotators are required to distinguish between human-preferred and non-preferred responses in the candidate responses for a given input (Ouyang et al., 2022). Based on this annotated data, researchers typically employ the Bradley-Terry model to construct a classification model and subsequently train the reward model using pairwise loss (Wang et al., 2023b; Rafailov et al., 2023). For a given context x and action y, the Bradley-Terry model represents the preference function $p(y_w \succ y_l)$ as a sigmoid of the difference of rewards:

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l)),$$

where $\sigma(\cdot)$ denotes the sigmoid function and plays the role of normalization, $r(x, y)$ denotes the pointwise reward of y given x, y_w denotes the human-preferred responses and y_l denotes the non-preferred responses. Given the dataset $\mathcal{D} = (x_i, y_{w,i} \succ y_{l,i})_{i=1}^{N}$ one can learn the reward function by optimizing the following logistic regression loss:

$$\mathcal{L}(x) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[log(p(y_w \succ y_l \mid x))].$$

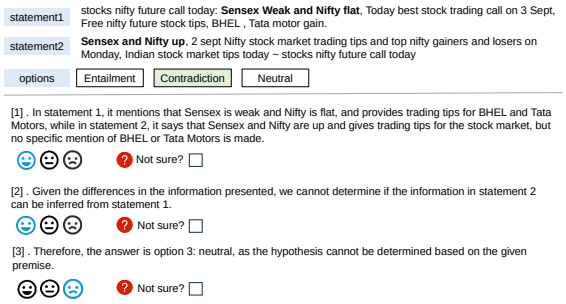


Figure 9: The data annotation approach for PRM. Unlike ORM, the annotation approach of PRM cannot generate pairwise preference data, thus precluding the use of the Bradley-Terry method for training the reward model.

In the process of training the PRMs, annotators are required to assess the correctness of each step in the model-generated solutions. Specifically, as illustrated in Figure 9, annotators typically need to determine whether the current reasoning step is negative, neutral, or positive, and correspondingly select from [-1, 0, 1] (Lightman et al., 2023; Ma et al., 2023). These annotated data are subsequently used to train the reward model, thereby enhancing its capability to distinguish and classify negative, neutral, and positive steps. However, due to the absence of pairwise comparison data regarding human preferences in this process, the Bradley-Terry model cannot be employed to construct a classification model. Here we redefine the training process of PRM.

In light of the coherence of reasoning steps, evaluating the accuracy of the k-th reasoning step y^k necessitates the simultaneous consideration of the input x and the preceding k reasoning steps y^k_{pre} as context. The reward model maps these inputs to an n-dimensional vector z, which encompasses the scores or raw outputs for each category. Formally, this can be represented as:

$$z = r(x, y^k_{pre}, y^k; \theta), \tag{7}$$

where θ denotes the parameters of the reward model, and $r(\cdot)$ denotes the reward model. We employ an activation function to transform the model

Models	AwpNLI	NewsNLI	RedditNLI	RTE-Quant	StressTest	QQA
Llama2-7B (Touvron et al., 2023)	1.47%	0.47%	0.40%	0.86%	1.36%	3.70%
BLOOMZ (Muennighoff et al., 2023)	48.04%	54.46%	37.20%	47.64%	31.22%	51.85%
Abel-7B (Chern et al., 2023)	55.82%	50.75%	47.20%	56.67%	30.87%	48.14%
Llama3.1-8B (Dubey et al., 2024a)	66.18%	62.91%	39.60%	48.93%	13.04%	50.62%
Qwen2-7B-chat (Yang et al., 2024a)	54.90%	54.93%	40.00%	21.13%	27.32%	46.30%
CN-PPO (Liang et al., 2024)	82.35%	61.97%	63.20%	63.52%	46.30%	48.77%
PSPO (Ours)	**86.76%**	**64.91%**	**67.60%**	**71.57%**	**52.29%**	**54.70%**

Table 3: Performance comparison on numerical understanding benchmarks. Our PSPO method, built upon Abel-7B, consistently outperforms baseline models across all six datasets that test LLMs' numerical sensitivity.

outputs into a probability distribution:

$$p(z_i) = \sigma(z_i), \tag{8}$$

where $p(z_i)$ denotes represents the probability that the current step belongs to category i, and $\sigma(\cdot)$ denotes the activation function, which is typically the softmax function:

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}, \tag{9}$$

where n denotes the total number of categories. The training of reward models typically employs the cross-entropy loss function to quantify the divergence between the predicted probability distribution and the true labels. Let $\mathbf{z}$ denote the one-hot encoded vector of the true labels. The training process is then formulated as follows:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}}[\sum_{i=1}^{N} \mathbf{z} log(p(z_i))]. \tag{10}$$

Ultimately, the reward model $r(x, y^k_{pre}, y^k)$ predicts the probabilities of the current reasoning step y belonging to various categories. The probability assigned to the positive category is then used as the reward score for the current reasoning step, as follows:

$$R^k_s = p(z_{i=1}), \tag{11}$$

where R^k_s denotes the reward score of the k-th reasoning step.

B Detailed Analysis on Reasoning Chain Length

For GSM8K experiments, we trained Llama3.1-8B (Dubey et al., 2024b), Qwen2.5-7B (Yang et al., 2024b) using the process supervision method proposed by Ma et al. (2023) with the PRM-800K dataset. The models were then evaluated on the GSM8K dataset for reasoning chain generation. We used the original step-by-step solutions provided in the GSM8K dataset as our ground truth reference for length comparison. The reported lengths are averaged across all samples in the test set.

For AwpNLI experiments, we trained the Abel-7B (Chern et al., 2023) using the process supervision method from Ma et al. (2023) with the annotated data detailed in Appendix B. To facilitate the quantitative analysis of reasoning steps, we required the model to explicitly number each generated reasoning step (e.g., [1], [2], ...). The standard solutions were generated by GPT-4 and underwent manual verification of reasoning steps.

C Experimental Details on Numerical Reasoning

C.1 Human-data Collection for Training Reward Model

Training a robust reward model requires a balanced label distribution. While the steps generated by GPT-3.5 predominantly feature positive labels, we included additional reasoning step candidates from other LLMs, such as Abel-7b, to provide more negative examples and achieve label balance. Human labelers would evaluate the given steps by their correctness, and correct answers to the question are provided as a reference. The statistics of datasets are shown in Table 4.

Step Labelling Criteria Each reasoning step is evaluated and assigned a label based on its correctness: 'positive' (score of '1'), 'neutral' (score of '0'), and 'negative' (score of '-1'). A step receives a positive score if it accurately meets logical and computational requirements, correctly interprets the task, and contributes to deriving the correct answer. A neutral score is awarded if the step is correct but does not aid in reaching the correct conclusion. Conversely, steps that contain logical, computational, or factual inaccuracies, or are irrele-

Datasets	Cases	Human labeled			
		Pos.	Neu.	Neg.	Steps
AwpNLI	1622	4334	822	1669	7109
NewsNLI	1643	3358	910	2870	7502
RedditNLI	1152	3074	507	958	4674
RTE_Quant	1324	3363	290	914	4817
StressTest	1369	2598	723	1921	5696
QQA	1394	3937	184	1778	6424

Table 4: The step data labeled by human annotators. "Cases" is the number of solutions generated by models, "Pos.", "Neu.", and "Neg." are the number of positive, neutral, and negative labels after labeling, respectively, "Steps" is the total number of reasoning steps taken to solve all the questions in the dataset.

vant to the given context and question, are assigned a negative score of -1.

C.2 Experimental Setups

Datasets We adopt the MATH dataset, which includes AwpNLI, NewsNLI, RedditNLI, RTE-Quant, StressTest, and QQA datasets as reported by Chen et al. (2023). These datasets are further expanded using the GPT-3.5 API, as detailed in Table 4. The training dataset for the reward model is primarily composed of data labeled as '1'. To ensure a balanced dataset, steps labeled '0' and '-1' are replicated 2-3 times, yielding a final count of 16,587 positive, 11,072 neutral, and 16,236 negative steps. For evaluation, 20% of the dataset is designated as test sets.

Metrics and Parameters setting The evaluation metric utilized is the average micro-F1 score on the test dataset because it balances precision and recall, providing a more comprehensive measure of model performance. We employ Abel-7B as the baseline model, which has been fine-tuned on a substantial portion of the MATH dataset for generating chain-of-thought reasoning in mathematical problem-solving (Chern et al., 2023). The reward model is trained on the BERT-large (Devlin et al., 2019) due to its proven efficacy in classification tasks (Gao et al., 2023). We trained the reward model over 10 epochs with a learning rate of 2e-5, a warmup rate of 0.05, and a maximum sequence length of 256.PPO training uses Lora (Hu et al., 2022) with a learning rate of 1.41e-5 and a maximum of 512 tokens. On a dataset of 5470 entries, each epoch averages 55 hours on four NVIDIA A100 GPUs. In our PSPO-WRS method, the parameters are set as follows: $C = 10.735$, $k = 1.5$, and $\lambda = 8.0$.

C.3 Overall Results

Table 3 presents the performance comparison between our PSPO and several strong baseline models. Building upon the Abel-7B (Chern et al., 2023), PSPO consistently outperforms all baselines across all six datasets. Notably, on the AwpNLI dataset, PSPO-WRS achieves 86.76% accuracy, surpassing the previous best result (82.35% by CN-PPO (Liang et al., 2024)) by 4.41%. Similar improvements are observed across other datasets, with particularly substantial gains on RTE-Quant (71.57% vs. 63.52%) and RedditNLI (67.60% vs. 63.20%).