Beyond Step Pruning: Information Theory Based Step-level Optimization for Self-Refining Large Language Models

Jinman Zhao[1,*], Erxue Min[2], Hui Wu[3], Ziheng Li[4], Zexu Sun[2,†], Hengyi Cai[2], Shuaiqiang Wang[2], Xu Chen[5,†], Gerald Penn[1]

[1]Department of Computer Science, University of Toronto
[2]Baidu Inc.
[3]Aerospace Information Research Institute, Chinese Academy of Sciences
[4]School of Intelligence Science and Technology, Peking University
[5]Gaoling School of Artificial Intelligence, Renmin University of China
jzhao@cs.toronto.edu, sunzexu0826@gmail.com, xu.chen@ruc.edu.cn

Abstract

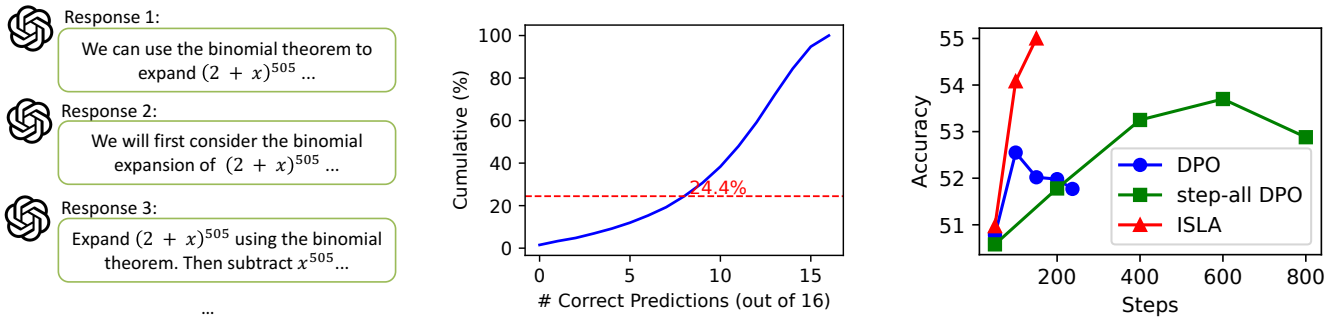
Large language models (LLMs) have shown impressive capabilities in natural language tasks, yet they continue to struggle with multi-step mathematical reasoning, where correctness depends on a precise chain of intermediate steps. Preference optimization methods such as Direct Preference Optimization (DPO) have improved answer-level alignment, but they often overlook the reasoning process itself, providing little supervision over intermediate steps that are critical for complex problem-solving. Existing fine-grained approaches typically rely on strong annotators or reward models to assess the quality of individual steps. However, reward models are vulnerable to reward hacking. We propose **ISLA**, a reward-model-free framework that constructs step-level preference data directly from SFT gold solutions. ISLA also introduces a self-improving pruning mechanism that identifies informative steps based on two signals: their marginal contribution to final accuracy (*relative accuracy*) and the model's *uncertainty*, inspired by the concept of information theorem. Empirically, ISLA achieves better performance than DPO while using only 12% of the training chosen tokens, demonstrating that careful step-level selection can significantly improve both reasoning accuracy and training efficiency.

Introduction

Recent years, one common approach to evaluate the capabilities of large language models (LLMs) (Achiam et al. 2023; Grattafiori et al. 2024; Yang et al. 2025) is to assess their performance on mathematical reasoning tasks, where models are asked to solve complex problems that require multi-step logical deduction. Among existing techniques, step-by-step prompting has proven particularly effective. For instance, Chain-of-Thought (CoT) prompting (Wei et al. 2022) enhances performance by explicitly encouraging models to verbalize intermediate steps. Building on this intuition, recent work such as Process Reward Models (PRMs) (Zhang et al. 2025b) and Long CoT (Jaech et al. 2024) further emphasize the importance of fine-grained supervision and structured reasoning trajectories, highlighting the value of guiding or evaluating LLMs at the level of individual reasoning steps. In parallel, alignment (Dong et al. 2025b) methods based on Reinforcement Learning from Human Feedback (RLHF) such as Direct Preference Optimization (DPO) (Rafailov et al. 2023), Proximal Policy Optimization (PPO) (Schulman et al. 2017), and GRPO (Shao et al. 2024) have become standard tools. However, these methods typically treat each training sample as a complete solution, implicitly assuming the model must learn the full reasoning chain holistically, without exploiting the internal structure of the reasoning process.

Despite the promise of preference-based alignment methods such as DPO, applying them directly to mathematical reasoning tasks presents several challenges. First, several DPO variants (Meng, Xia, and Chen 2024; Lai et al. 2024) rely on the model to self-sample multiple candidate responses and use a reward model to identify the best and worst completions as training pairs. However, for math problems, the model's sampled predictions often share identical or same-meaning initial reasoning steps. As shown in Figure 1a, where we visualize three sampled completions from *Qwen2.5-7B* on the same math question *"Find the coefficient of* x^{504} *in the expansion of* $(2+x)^{505}-x^{505}$*"*, the beginning tokens are highly similar. These shared prefixes can confuse the DPO loss that expects the chosen and rejected responses have large likelihood gap. This suggests that coarse-grained, solution-level preference optimization may be suboptimal for math reasoning tasks. Second, using self-sampled completions to construct training pairs introduces a distributional bias toward easier problems. It becomes difficult to find usable preference pairs for harder problems, where the model often fails to produce any valid output. To quantify this, we sampled 16 completions per question from the dataset of Lai et al. (2024) and computed how many of them were correctly sampled by LLM. As illustrated in Figure 1b, over 75% of the training math question are already solvable by the model. This lack of signal on difficult questions hampers the ability of DPO to improve reasoning where it matters most. Third, while Process Reward Models (PRMs) (Zhang et al. 2025b) have been proposed to identify critical intermediate reasoning steps, they often suffer from stability issues and reward

[*]Work done during an internship at Baidu Inc.
[†]Corresponding author



(a) DPO example. (b) Accuracy of existing training data. (c) Accuracy comparison between methods.

Figure 1: a) DPO preference building example for the question *"Find the coefficient of* x^{504} *in the expansion of* $(2+x)^{505} - x^{505}$*"*, solution prefix are nearly identical. b) Cumulative distribution of accuracy for the training dateset picked by Lai et al. (2024). The red dashed line at $x = 8$ indicates that only $1 - 24.4\% = 75.6\%$ questions are easy ($\geq 50\%$ accuracy). c) Accuracy of different method during training. Ours shows highest performance with minimum steps and tokens.

hacking (Wang et al. 2025b).

To address these limitations, we propose our method **I**nformation Theory Based **S**tep-**l**evel Optimiz**a**tion (ISLA) for Self-Refining LLMs. Our method leverages high-quality step by step solution already present in standard SFT dataset, particularly focusing on harder, gold-labeled reasoning steps that are more informative for model learning. Concretely, we decompose each full solution in the SFT dataset into individual reasoning steps and construct contrastive training pairs by pairing the gold next step (as the preferred choice) with a model self-generated alternative (as the rejected option), conditioned on the same partial reasoning history. This formulation follows self-play (Chen et al. 2024) style improvement without relying on external reward models, and maintaining alignment pressure at a finer granularity than traditional solution-level preference optimization. To maximize learning efficiency, we further introduce a step scoring mechanism that selects training pairs based on a combination of **model uncertainty** (via average log-probability) and **information gain** (via relative accuracy gain). This allows us to prioritize steps that are both challenging to predict and crucial for arriving at the correct final answer, reducing unnecessary training on low-impact or trivial steps.

In summary, our main contributions are:

- We introduce a novel training paradigm for math that extracts preference data directly from existing SFT dataset by contrasting gold reasoning steps with model-generated alternatives, enabling self-improvement step-level optimization without requiring external reward models or additional human labeling.
- We introduce a principled step selection strategy based on information theory, enabling the self-identification of high-impact reasoning steps. Using this metric, we retain a Step-level DPO training dataset that is just 12% the size (in chosen tokens) of standard DPO data (Figire 1c).
- We conduct extensive experiments across multiple model scales and mathematical reasoning benchmarks, showing that models trained on our filtered dataset consistently outperform those trained with standard DPO data.

Related Work

Step-Level Math Enhancement. More recently, the emergence of reasoning LLMs such as OpenAI's o1 (Jaech et al. 2024) and DeepSeek-Math R1 (DeepSeek-AI 2025) has sparked growing interest in Long Chain-of-Thought reasoning (Sun et al. 2024; Team et al. 2025; Yax, Anlló, and Palminteri 2024; Zhong et al. 2025), which extends traditional CoT prompting with more detailed, multi-hop, and reflective reasoning. Long CoT involves iterative chains of intermediate steps, often constructed through inference-time scaling (Teng et al. 2025; Li 2025). Step-DPO (Lai et al. 2024) introduced a method for step-level preference optimization, focusing on the first erroneous step in a reasoning sequence. While effective, this approach only updates one critical step and ignores the rest of the reasoning chain. Full-Step-DPO (Xu et al. 2025) proposed a new step-wise DPO loss that dynamically weights each step's gradient based on a self-supervised PRM trained without external annotations. Wu et al. (2025) employs a step-level verifier and reviser to iteratively refine reasoning chains. (Liu et al. 2025) adapts step-level fine-grained feedback for theorem proving.

Data/Token Selection. Recent work has highlighted that not all training tokens contribute equally to model finetuning. Zhou et al. (2023) demonstrates that strong instruction tuning is possible with only a few hundred carefully curated examples, emphasizing quality over quantity. training-free prompting methods can already improve sentence embeddings at inference time (Fu et al. 2025; Cheng et al. 2025). In long-context modeling, several studies (Hu et al. 2025; Wu, Zhao, and Zheng 2024; Zhu et al. 2024) show that compressing or pruning uninformative tokens leads to significant computational savings without loss in performance. In multimodal settings, several research (Li, Yang, and Lu 2025; Liu et al. 2023; Liu, Wu, and Guo 2023) observe that filtering out redundant modality-specific inputs (e.g., visual patches) can improve alignment (Dong et al. 2025a) and robustness (Dong et al. 2024). Furthermore, recent findings (Wang et al. 2025a) suggest that in GRPO (Shao et al.

2024), it is not necessary to train on all tokens: selecting only high-entropy tokens can be sufficient.

RLHF for Math Reasoning. RLHF has shown significant promise in enhancing the mathematical reasoning capabilities of LLMs. PPO (Schulman et al. 2017) became the standard in early RLHF pipelines, but its reliance on a separate critic model introduced substantial computational and memory overhead. To mitigate this, ReMax (Li et al. 2023) and GRPO (Shao et al. 2024) eliminate the need for a critic by leveraging REINFORCE-style updates with simplified baseline estimation. DAPO (Yu et al. 2025) further improves exploration by addressing entropy collapse during training. Several works have proposed structural enhancements to encourage deeper reasoning. Self-Reward methods (Xiong et al. 2025; Pang et al. 2024) integrate verification and critique into the model architecture, while RFTT (Zhang et al. 2025a) introduces symbolic control tokens to structure long chains of thought. To address the complexity of online RL, DPO (Rafailov et al. 2023) and its variants (Meng, Xia, and Chen 2024) provide an efficient alternative by directly optimizing on offline preference pairs. RLHF-style also underpins recent agentic systems such as search (Li et al. 2025).

Preliminaries

Preference Optimization for LLM

Direct Preference Optimization DPO offers a simpler and more stable alternative by eliminating the need for explicit reward modeling or reinforcement learning. Instead, it directly learns from pairwise preference data using a contrastive objective derived from the principle of maximum entropy inverse reinforcement learning.

Given a dataset of preference triplets (x, y^+, y^-), x is used to denote the input prompt, y^+ is the preferred response, and y^- is the rejected response. In the domain of mathematical reasoning, x typically corresponds to a question, while y^+ and y^- represent a preferred correct answer and a rejected incorrect answer, respectively. The DPO objective is defined as:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \Big[\log \left(\frac{\pi(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)} \right) - \log \left(\frac{\pi(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)} \right) \Big] \right), \quad (1)$$

where π is the current policy, π_{ref} is a fixed reference policy, σ is the sigmoid function, and β is a temperature parameter controlling the sharpness of preference enforcement. This loss encourages the model to increase the relative likelihood of preferred completions while regularizing deviation from the reference policy.

Information Theory

Information theory offers foundational tools for quantifying uncertainty and informativeness in probability distributions. Key concepts we rely on are entropy and information gain.

Entropy. Given a discrete random variable X with probability distribution $P(X)$, the (Shannon) entropy of X is defined as:

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log P(x). \quad (2)$$

Entropy quantifies the expected amount of uncertainty or information content in the outcome of X. A higher entropy indicates a more uniform (uncertain) distribution, while lower entropy reflects greater certainty or skew.

Conditional Entropy. The uncertainty in a random variable Y given knowledge of another variable X is captured by the conditional entropy:

$$H(Y \mid X = x) = -P(y \mid x) \log P(y \mid x), \quad (3)$$

$$H(Y \mid X) = \sum_{x \in \mathcal{X}} P(x) H(Y \mid X = x) \quad (4)$$

$$= -\sum_{x \in \mathcal{X}} P(x) \sum_{y \in \mathcal{Y}} P(y \mid x) \log P(y \mid x). \quad (5)$$

Information Gain. The information gain of observing X with respect to a target variable Y is defined as the reduction in entropy of Y:

$$IG(Y; X) = H(Y) - H(Y \mid X). \quad (6)$$

This quantity measures how much knowing X reduces uncertainty about Y, and is widely used to assess the utility of features, signals, or observations.

Method

In this section, we describe how DPO and information-theoretic principles can be extended to the step level, and introduce our pipeline in Figure 2.

Step-level DPO

While standard DPO operates at the full output level that comparing entire responses y^+ and y^- for a given prompt/question x, it does not account for the structured nature of multi-step reasoning tasks. Step-level DPO extends the DPO framework by applying preference optimization at the granularity of individual reasoning steps. Specifically, instead of optimizing $\pi(y^+ \mid x)$ against $\pi(y^- \mid x)$, It compares the conditional probabilities of a step s_i given the input x and the preceding reasoning steps $s_1, \ldots, s_{i-1}$. That is, for each step position i, the model is trained to prefer the preferred step s_i^+ over a rejected alternative s_i^- under the context, the updated loss is defined as:

$$\mathcal{L}_{\text{Step-DPO}} = -\log \sigma \left\{ \beta \left[\log \left(\frac{\pi(s_i^+ \mid x, s_{<i})}{\pi_{\text{ref}}(s_i^+ \mid x, s_{<i})} \right) - \log \left(\frac{\pi(s_i^- \mid x, s_{<i})}{\pi_{\text{ref}}(s_i^- \mid x, s_{<i})} \right) \right] \right\}, \quad (7)$$

Here, $s_{<i} = s_1, \ldots, s_{i-1}$ denotes the preceding reasoning trace up to step $i-1$, and π is optimized to prefer the correct

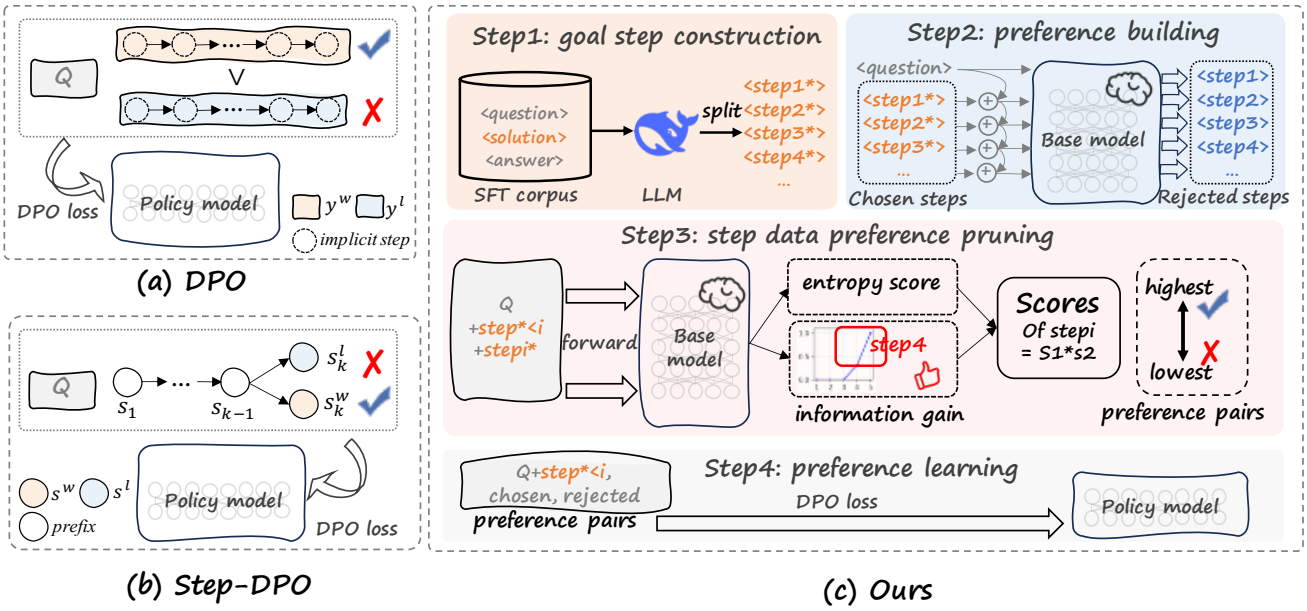


Figure 2: Overview of our ILSA. We also compare our ILSA with DPO and Step-DPO.

continuation s_i^+ over the incorrect s_i^-, conditioned on this context. By shifting the preference signal from full-sequence completions to individual reasoning steps, Step-level DPO enables more fine-grained control and supervision during alignment, especially for math that requiring precise, interpretable reasoning.

Step-level Preference Data Preparation

Self-guided Question Filtering. To ensure the constructed preference data focuses on non-trivial and informative reasoning problems, we first apply a question filtering phase based on model performance. For each question in the original SFT dataset, we prompt a base model (e.g., *Qwen2.5-7B*) to generate 8 independent completions then apply a solution verifier to evaluate whether the final answer in each completion is correct. Questions for which the model fails to produce any correct solution across all 8 attempts are retained. This heuristic aims to select questions that are challenging for the model, thereby increasing the likelihood that their intermediate reasoning steps are also non-obvious and pedagogically valuable.

Step Data Construction. Existing Step data preparation pipeline (Lai et al. 2024; Xu et al. 2025) typically construct training pairs by generating both preferred and rejected steps from the same model, without effectively leveraging existing high-quality sft chain of thought based data. This leads to suboptimal use of annotated reasoning traces, especially in domains like mathematics where ground-truth solutions are available and informative. To this end, we propose a data construction method that utilizes existing SFT datasets containing detailed reasoning traces.

Formally, for each reasoning instance, let the input question be denoted by x, and let the gold solution consist of a sequence of reasoning steps $(s_1^*, s_2^*, \ldots, s_T^*)$. At each position i, the gold next-step s_i^+ is defined as s_i^*, and the conditioning context is given by $(x, s_1^*, \ldots, s_{i-1}^*)$. To construct a contrastive pair, we generate a negative step s_i^- by prompting a language model with prompt:

$$x,\ s_1^*, \ldots, s_{i-1}^*,\ \texttt{What is the next step?}$$

The output of the model is then used as s_i^-. This process simulates a realistic rejection sample in the context of multi-step reasoning, where the model must distinguish between the correct next step and its own plausible but incorrect continuation. By applying this procedure across all valid positions $i \in \{1, \ldots, T\}$ for each example in the SFT corpus, we obtain a dataset of step-level preference triples $(x, s_{<i}^*, s_i^+, s_i^-)$ that used to train Step-level DPO models.

Self-selective Step Data Pruning

To reduce training cost while preserving the most informative supervision signals, we introduce a step-level data pruning strategy. Our goal is to identify and retain only those reasoning steps that are both critical for producing a correct final answer and challenging for the model to predict. This strategy is grounded in the information-theoretic notion of *information gain*, which measures how much a new piece of information reduces uncertainty about a target variable.

We define two complementary metrics to quantify the importance of each step.

Information Gain and Relative Accuracy. From an information-theoretic perspective, the utility of a reasoning step s_i^* can be measured by how much it reduces the uncertainty about the final answer y. Formally, we define the information gain of a step as:

$$\mathrm{IG}(s_i) = H(y \mid x, s_{<i}^*) - H(y \mid x, s_{\leq i}^*), \tag{8}$$

where $H(y \mid \cdot)$ denotes the entropy of the model's predictive distribution over final answers conditioned on the reasoning prefix. A larger $\mathrm{IG}(s_i)$ implies that observing s_i^* makes the final answer more certain.

However, exact entropy computation is intractable. To approximate $\mathrm{IG}(s_i)$, we define a model-based proxy called *Relative Accuracy (ReAcc)*. For each step position i, we compare the model's accuracy in generating the final answer before and after including the step s_i^*. Concretely, we construct two contexts: the partial trace $(x, s_1^*, \ldots, s_{i-1}^*)$ and the extended trace $(x, s_1^*, \ldots, s_i^*)$. We then compute:

$$\mathrm{ReAcc}(s_i) = \mathrm{Accuracy}(x, s_{\leq i}^*) - \mathrm{Accuracy}(x, s_{<i}^*). \tag{9}$$

This provides a practical estimate of a step's informativeness by evaluating how much it improves the likelihood of producing the correct final answer.

Token-level Uncertainty via Log-Probability. Information theory quantifies uncertainty using entropy. For a reasoning step s_i^*, the token-level entropy under prefix $(x, s_1^*, \ldots, s_{i-1}^*)$ is defined as:

$$p_t(v) = \pi(v \mid x, s_{<i}^*, s_i^*[< t]), \tag{10}$$

$$\overline{H}(s_i) = \frac{1}{|s_i^*|} \sum_{t=1}^{|s_i^*|} \left(- \sum_{v \in \mathcal{V}} p_t(v) \log p_t(v) \right). \tag{11}$$

This captures the model's average uncertainty over the vocabulary at each token position. However, exact entropy computation requires marginalizing over all possible tokens and is computationally expensive in practice.

To approximate this uncertainty more efficiently, we use the token-level average log-probability of the gold step s_i^*:

$$\overline{\log p}(s_i) = \frac{1}{|s_i^*|} \sum_{t=1}^{|s_i^*|} \log \pi(s_i^*[t] \mid x, s_{<i}^*, s_i^*[< t]). \tag{12}$$

Lower $\overline{\log p}(s_i)$ implies higher uncertainty, analogous to higher entropy. To validate this relationship, we randomly sampled 6,000 reasoning steps from our training corpus and computed both the token-level average log-probability $\overline{\log p}(s_i)$ and the corresponding entropy $\overline{H}(s_i)$ for each step. We observed a strong empirical correlation between the two metrics (Pearson: 0.90, Spearman: 0.91), supporting the use of $\overline{\log p}$ as a reliable and efficient proxy for entropy in evaluating model uncertainty. This finding justifies our use of $\overline{\log p}$ in step selection, as it avoids the costly marginalization over vocabulary required for exact entropy computation while preserving meaningful uncertainty signals.

Taxonomy of Reasoning Behaviors. Different combinations of above metrics reveal distinct reasoning behaviors:

1. $ReAcc \uparrow$, $\overline{logp} \downarrow$: The *ideal* steps that the model finds difficult yet are highly beneficial. These are promising candidates for focused training.
2. $ReAcc \uparrow$, $\overline{logp} \uparrow$: Useful steps that the model already handles with confidence. These are less urgent for training but still desirable.
3. $ReAcc \downarrow$, $\overline{logp} \downarrow$: Difficult steps that do not improve the answer. These may reflect noise, hallucination, or poorly grounded reasoning.
4. $ReAcc \downarrow$, $\overline{logp} \uparrow$: Steps that are both unhelpful and easy for the model. These offer limited training value.

Combined Score. Each metric alone is insufficient; their combination more accurately identifies steps that are both impactful and difficult. To jointly capture step gain and uncertainty, we initially define a composite score as:

$$\mathrm{Score}(s_i) = \mathrm{ReAcc}(s_i) \cdot \left(-\overline{\log p}(s_i)\right). \tag{13}$$

This formulation prioritizes steps that are both difficult for the model (low log-probability) and highly beneficial for final answer accuracy (high ReAcc). However, we observe that $-\overline{\log p}(s_i)$ exhibits a long-tailed distribution: a small number of extremely low-probability steps dominate the score due to noise, including hallucinations, off-topic predictions, or syntax errors.

To mitigate this issue, we apply a clipping function to suppress outliers in the uncertainty term. Specifically, we analyze the empirical CDF of $\overline{\log p}(s_i)$ and identify the point where the slope begins to rise steeply—indicating the boundary between the noisy tail and the meaningful distribution. Letting x_i denote sorted log-probability values and $\mathrm{CDF}(x)$ the cumulative proportion, we define the lower bound τ as:

$$\tau = \min \left\{ x_i \;\middle|\; \frac{d\,\mathrm{CDF}(x)}{dx}\bigg|_{x=x_i} > 5 \cdot \mathrm{median}\left(\frac{d\,\mathrm{CDF}}{dx}\right) \right\}, \tag{14}$$

We then set the clipping range to $[\tau, 0]$ to ensure pathological low-probability steps do not dominate. The final score becomes:

$$\mathrm{Score}(s_i) = \mathrm{ReAcc}(s_i) \cdot \mathrm{clip}\left(-\overline{\log p}(s_i),\ \tau,\ 0\right). \tag{15}$$

This principled modification retains the desirable properties of the original score while preventing overemphasis on spurious or structurally invalid reasoning steps. We rank all steps by $\mathrm{Score}(s_i)$ and retain the top-k based on chosen steps for Step-level DPO training.

Experimental Setup

In this section, we describe the experimental configurations.

Model Selection

We conduct our experiments using the Qwen2.5 model family (Qwen et al. 2025), selecting three sizes to evaluate scalability and generalization under varying capacity constraints: *Qwen2.5-1.5B*, *Qwen2.5-3B*, and *Qwen2.5-7B*. The Qwen2.5 series is a strong open-source language model with competitive reasoning abilities. For reference model to select data, we pick *Qwen2.5-7B*.

Benchmark Selection

We select a diverse suite of benchmarks that vary in difficulty and domain. Specifically, we use **GSM8K** (Cobbe et al. 2021), **MATH500**, **Minerva-MATH** (Lewkowycz et al. 2022), **Gaokao23-Math** (Liao et al. 2024), **OlympiadBench** (He et al. 2024) and **CollegeMath** (Tang et al. 2024).

Method	Token %	GSM8K	MATH500	Minerva Math	Gaokao 2023	Olympiad	CollegeMath	Avg
Qwen2.5-1.5B								
base		3.2	3.3	2.2	4.7	1.8	3.8	3.2
DPO	100	63.2$_{\uparrow 60.0}$	42.4$_{\uparrow 39.1}$	9.2$_{\uparrow 7.0}$	37.1$_{\uparrow 32.4}$	14.5$_{\uparrow 12.7}$	31.7$_{\uparrow 27.9}$	33.0$_{\uparrow 29.8}$
Step all DPO	100	69.5$_{\uparrow 66.3}$	49.6$_{\uparrow 46.3}$	14.0$_{\uparrow 11.8}$	**44.2**$_{\uparrow 39.5}$	17.1$_{\uparrow 15.3}$	35.7$_{\uparrow 31.9}$	38.4$_{\uparrow 35.2}$
+logp	20	66.6$_{\uparrow 63.4}$	46.3$_{\uparrow 43.0}$	14.0$_{\uparrow 11.8}$	43.4$_{\uparrow 38.7}$	17.2$_{\uparrow 15.4}$	32.3$_{\uparrow 28.5}$	36.6$_{\uparrow 33.4}$
+relative acc	20	68.5$_{\uparrow 65.3}$	49.4$_{\uparrow 46.1}$	16.2$_{\uparrow 14.0}$	42.9$_{\uparrow 38.2}$	**18.1**$_{\uparrow 16.3}$	37.5$_{\uparrow 33.7}$	38.8$_{\uparrow 35.6}$
+random	20	69.4$_{\uparrow 66.2}$	47.8$_{\uparrow 44.5}$	12.5$_{\uparrow 10.3}$	43.7$_{\uparrow 39.0}$	17.0$_{\uparrow 15.2}$	35.5$_{\uparrow 31.7}$	37.7$_{\uparrow 34.5}$
ISLA	20	**70.1**$_{\uparrow 66.9}$	**50.2**$_{\uparrow 46.9}$	**16.5**$_{\uparrow 14.3}$	43.6$_{\uparrow 38.9}$	16.7$_{\uparrow 14.9}$	**37.9**$_{\uparrow 34.1}$	**39.2**$_{\uparrow 36.0}$
Qwen2.5-3B								
base		72.9	50.8	17.0	48.3	17.0	35.5	40.3
DPO	100.0	76.6$_{\uparrow 3.7}$	58.0$_{\uparrow 7.2}$	**24.3**$_{\uparrow 7.3}$	49.6$_{\uparrow 1.3}$	21.5$_{\uparrow 4.5}$	37.2$_{\uparrow 1.7}$	44.5$_{\uparrow 4.2}$
Step all DPO	100.0	**81.0**$_{\uparrow 8.1}$	59.2$_{\uparrow 8.4}$	22.1$_{\uparrow 5.1}$	**50.6**$_{\uparrow 2.3}$	21.3$_{\uparrow 4.3}$	39.0$_{\uparrow 3.5}$	45.5$_{\uparrow 5.2}$
+logp	12.0	77.3$_{\uparrow 4.4}$	58.0$_{\uparrow 7.2}$	23.5$_{\uparrow 6.5}$	50.4$_{\uparrow 2.1}$	21.3$_{\uparrow 4.3}$	38.0$_{\uparrow 2.5}$	44.8$_{\uparrow 4.5}$
+relative acc	12.0	80.0$_{\uparrow 7.1}$	58.0$_{\uparrow 7.2}$	23.5$_{\uparrow 6.5}$	48.8$_{\uparrow 0.5}$	21.4$_{\uparrow 4.4}$	38.4$_{\uparrow 2.9}$	45.0$_{\uparrow 4.7}$
+random	12.0	73.8$_{\uparrow 0.9}$	55.0$_{\uparrow 4.2}$	23.9$_{\uparrow 6.9}$	44.4$_{\uparrow -3.9}$	21.5$_{\uparrow 4.5}$	37.2$_{\uparrow 1.7}$	42.6$_{\uparrow 2.3}$
ISLA	12.0	79.1$_{\uparrow 6.2}$	**61.4**$_{\uparrow 10.6}$	23.9$_{\uparrow 6.9}$	50.4$_{\uparrow 2.1}$	**22.5**$_{\uparrow 5.5}$	**39.4**$_{\uparrow 3.9}$	**46.1**$_{\uparrow 5.8}$
Qwen2.5-7B								
base		87.3	61.4	21.4	55.6	23.9	41.9	48.6
DPO	100.0	87.3$_{\uparrow 0.0}$	66.6$_{\uparrow 5.2}$	27.9$_{\uparrow 6.5}$	57.9$_{\uparrow 2.3}$	31.6$_{\uparrow 7.7}$	43.6$_{\uparrow 1.7}$	52.5$_{\uparrow 3.9}$
Step all DPO	100.0	89.5$_{\uparrow 2.2}$	68.0$_{\uparrow 6.6}$	28.2$_{\uparrow 6.8}$	60.5$_{\uparrow 4.9}$	31.0$_{\uparrow 7.1}$	44.9$_{\uparrow 3.0}$	53.7$_{\uparrow 5.1}$
+logp	12.0	88.4$_{\uparrow 1.1}$	64.4$_{\uparrow 3.0}$	28.7$_{\uparrow 7.3}$	59.2$_{\uparrow 3.6}$	31.6$_{\uparrow 7.7}$	45.1$_{\uparrow 3.2}$	52.9$_{\uparrow 4.3}$
+relative acc	12.0	89.1$_{\uparrow 1.8}$	**69.6**$_{\uparrow 8.2}$	32.0$_{\uparrow 10.6}$	59.0$_{\uparrow 3.4}$	31.0$_{\uparrow 7.1}$	44.5$_{\uparrow 2.6}$	54.2$_{\uparrow 5.6}$
+random	12.0	87.8$_{\uparrow 0.5}$	65.4$_{\uparrow 4.0}$	26.8$_{\uparrow 5.4}$	59.0$_{\uparrow 3.4}$	28.7$_{\uparrow 4.8}$	44.3$_{\uparrow 2.4}$	52.0$_{\uparrow 3.4}$
ISLA	12.0	**89.7**$_{\uparrow 2.4}$	67.6$_{\uparrow 6.2}$	**34.2**$_{\uparrow 12.8}$	**61.0**$_{\uparrow 5.4}$	**32.4**$_{\uparrow 8.5}$	**45.0**$_{\uparrow 3.1}$	**55.0**$_{\uparrow 6.4}$

Table 1: Results of DPO, our step-level DPO before pruning and different step selection strategies on the step-all setting. Step all means all steps are used for training. Random samples steps uniformly, while logp, relative acc, and ISLA select top-k steps based on different metrics, respectively.

Dataset Selection

We select DeepMath-103K (He et al. 2025), a large-scale, decontaminated SFT dataset with a strong focus on higher-difficulty questions as our training dataset. We first filter out 10k problems that are difficult, i.e., those that reference model fails to produce a correct answer. Then annotate step-by-step reasoning traces using DeepSeek-V3. While we employed an LLM-based labeler during preprocessing, it was only used for segmenting step-wise traces. Step segmentation is relatively shallow and does not rely on semantic judgment, avoiding issues like reward hacking or annotation bias.

Baseline Selection

We consider two categories of baselines for comparison:

- We adopt the standard **solution-level DPO** setup, where the *chosen* response is the SFT full solution, and the *rejected* response is an incorrect solution sampled from the reference model.
- We further evaluate **step-level DPO** baselines with all available reasoning steps and selecting the top data based on different criteria.

Results

In this section, we present the experimental results and provide analyses of our method.

Main Results

Table 1 presents the performance comparison of our method against several baselines on the various Qwen2.5 models. We observe that our method consistently outperforms all other approaches across most metrics. Compared to standard solution-level DPO, step-level DPO provides a noticeable gain, indicating the benefit of finer-grained supervision. Among the step selection strategies, using logp, relative accuracy, or random for top-k filtering yields incremental improvements over the full step set, with relative accuracy performing best among the three.

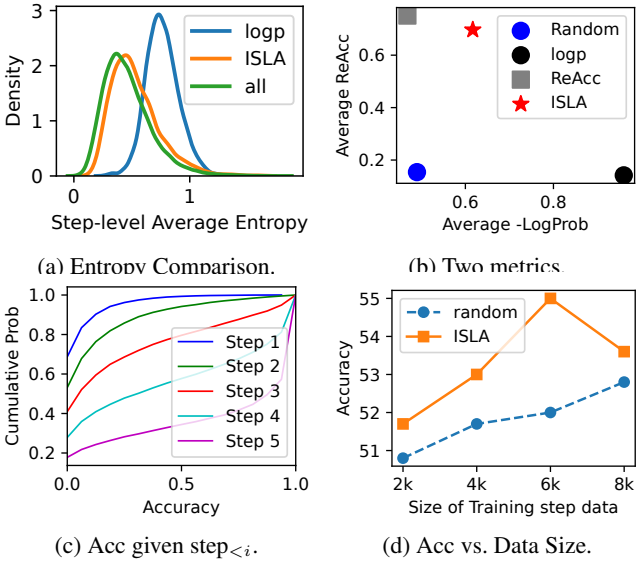
Our method achieves the highest performance on most of the benchmarks and average accuracy. Notably, despite using only 12% of the step data, our approach surpasses full-step DPO and other selection strategies, demonstrating the effectiveness of our step scoring mechanism in identifying informative reasoning steps. This highlights that careful step selection can lead to better supervision signals and improved reasoning quality, even with significantly less training data.

Other Study

Entropy Analysis of Selected Steps. We analyze the average token-level entropy (using *Qwen2.5-7B*) of steps selected by different criteria. As shown in Figure 3a, steps selected using logp-only exhibit the highest entropy, indicating high model uncertainty. However, such steps often include hallucinations or structurally invalid reasoning. In contrast,

Method	GSM8K	MATH500	Minerva Math	Gaokao 2023	Olympiad	CollegeMath	Avg
Qwen2.5-Math-7B-PRM800K	88.1	65.6	32.7	60.5	**32.6**	45.0	54.1
Llama-PRM800K	88.4	66.8	32.4	57.7	32.4	**45.1**	53.8
ISLA	**89.7**	**67.6**	**34.2**	**61.0**	32.4	45.0	**55.0**

Table 2: Comparison with different PRMs. Trained on *Qwen2.5-7B*.



(a) Entropy Comparison. (b) Two metrics.

(c) Acc given step$_{<i}$. (d) Acc vs. Data Size.

Figure 3: Analysis of our method. (a) Density of entropy for different data pruning strategy. (b) Strategy comparison in ReAcc vs. $-\overline{\log p}$ space. (c) Accuracy for including step $_{<i}$. (d) Accuracy vs. data size under different pruning methods.

our method selects steps with moderate-to-high entropy (Figire 3b), indicating a better trade-off between model uncertainty and learning potential. This distribution is more selective than step-all while avoiding the over-selection of pathological steps seen in logp-only. These results confirm that entropy alone is not sufficient, and incorporating task-specific impact (ReAcc) is crucial for effective supervision.

As shown in Figure 3a, each point represents a data selection strategy characterized by its average difficulty (negative log-probability) and information gain (relative accuracy). Our method lies in the top-right corner, indicating that it selects steps that are both challenging for the model and highly beneficial to final answer accuracy.

Training Data Hardness. Figure 3c shows the CDF of accuracy for the *Qwen2.5-7B* when reasoning over prompts containing the question plus the first $i-1$ steps. For each prompt, we sample 16 responses and compute the empirical accuracy, which is then aggregated into the CDF curves. It is worth emphasizing that our dataset spans a wide range of difficulties, from straightforward prompts to highly challenging prompts. This stands in contrast to datasets like Lai et al. (2024) that primarily consist of relatively simple problems. The inclusion of diverse and harder questions in our data allows for a more robust step selection strategy.

Effect of the Numbers Training data. Figure 3d illustrates how the performance of *Qwen2.5-7B* varies with the number of training steps used. We compare our method against the random selection baseline at four data sizes: 2k, 4k, 6k, and 8k. Across all settings, our method consistently outperforms random selection, indicating that selecting higher-quality reasoning steps is more beneficial than simply increasing data quantity. Interestingly, we observe that our method achieves the best accuracy at 6k steps, after which performance slightly drops at 8k. This suggests that indiscriminately adding more steps, even if scored high, is not always optimal, and that harder or noisy reasoning steps may degrade performance.

Comparison with PRM. PRMs are often trained to assist reasoning LLMs by scoring or ranking intermediate steps. PRMs can provide useful signals and are also used for step selection (Xu et al. 2025). But they are known to suffer from stability issues and are prone to reward hacking (Wang et al. 2025b), making them less reliable in practice compared to rule-based or symbolic verifiers.

To evaluate the effectiveness of PRM-based selection strategy, we experimented with two widely used PRMs: *Qwen2.5-Math-7B-PRM800K* (Zhang et al. 2025c) and *LLaMA-PRM800K*[1]. For each PRM, we use the generated scores to rank all candidate reasoning steps and selected the top 6k for training. We then fine-tune the *Qwen2.5-7B* model on these selected steps. As shown in Table 2, our method outperforms both PRM-based selections on the majority of benchmarks. This suggests that our scoring mechanism offers more robust and informative supervision.

Conclusions

We present ISLA, an efficient and reward-model-free framework for fine-grained alignment of LLMs in multi-step mathematical reasoning. By reformulating step selection as an information-theoretic data pruning problem, ISLA identifies reasoning steps that are both impactful and uncertain steps that the model most needs to learn from. ISLA supports self-improvement by using the model's own behavior to guide data pruning, enabling scalable preference optimization even under tight resource budgets. Empirically, we demonstrate that ISLA can outperform standard DPO using only a fraction of the training tokens, highlighting the benefits of selective supervision.

[1]https://huggingface.co/UW-Madison-Lee-Lab/Llama-PRM800K

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S. et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chen, Z.; Deng, Y.; Yuan, H.; Ji, K.; and Gu, Q. 2024. Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 6621–6642. PMLR.

Cheng, Z.; Wang, Z.; Fu, Y.; Jiang, Z.; Yin, Y.; Wang, C.; and Gu, Q. 2025. Contrastive Prompting Enhances Sentence Embeddings in LLMs through Inference-Time Steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, 3475–3487.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J. et al. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Dong, J.; Chen, J.; Xie, X.; Lai, J.; and Chen, H. 2024. Survey on adversarial attack and defense for medical image analysis: Methods and challenges. *ACM Computing Surveys*, 57(3): 1–38.

Dong, J.; Koniusz, P.; Zhang, Y.; Zhu, H.; Liu, W.; Qu, X.; and Ong, Y.-S. 2025a. Improving Zero-Shot Adversarial Robustness in Vision-Language Models by Closed-form Alignment of Adversarial Path Simplices. In *Forty-second International Conference on Machine Learning*.

Dong, J.; Zhang, C.; Qu, X.; Ma, Z.; Koniusz, P.; and Ong, Y.-S. 2025b. Robust SuperAlignment: Weak-to-Strong Robustness Generalization for Vision-Language Models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Fu, Y.; Cheng, Z.; Jiang, Z.; Wang, Z.; Yin, Y.; Li, Z.; and Gu, Q. 2025. Token Prepending: A Training-Free Approach for Eliciting Better Sentence Embeddings from LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, 3168–3181.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A. et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

He, C.; Luo, R.; Bai, Y.; Hu, S.; Thai, Z. L.; Shen, J.; Hu, J.; Han, X.; Huang, Y. et al. 2024. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. arXiv:2402.14008.

He, Z.; Liang, T.; Xu, J.; Liu, Q.; Chen, X.; Wang, Y.; Song, L.; Yu, D.; Liang, Z. et al. 2025. DeepMath-103K: A Large-Scale, Challenging, Decontaminated, and Verifiable Mathematical Dataset for Advancing Reasoning. arXiv:2504.11456.

Hu, Z.; Liu, Y.; Zhao, J.; Wang, S.; WangYan, W.; Shen, W.; Gu, Q.; Luu, A. T.; Ng, S.-K. et al. 2025. LongRecipe: Recipe for Efficient Long Context Generalization in Large Language Models. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11857–11870. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A. et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Lai, X.; Tian, Z.; Chen, Y.; Yang, S.; Peng, X.; and Jia, J. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*.

Lewkowycz, A.; Andreassen, A. J.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V. V.; Slone, A.; Anil, C.; Schlag, I. et al. 2022. Solving Quantitative Reasoning Problems with Language Models. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Li, D.; Yang, Z.; and Lu, S. 2025. ToDRE: Visual Token Pruning via Diversity and Task Awareness for Efficient Large Vision-Language Models. arXiv:2505.18757.

Li, X. 2025. A Survey on LLM Test-Time Compute via Search: Tasks, LLM Profiling, Search Algorithms, and Relevant Frameworks. *Transactions on Machine Learning Research*.

Li, Y.; Cai, H.; Kong, R.; Chen, X.; Chen, J.; Yang, J.; Zhang, H.; Li, J.; Wu, J. et al. 2025. Towards AI Search Paradigm. arXiv:2506.17188.

Li, Z.; Xu, T.; Zhang, Y.; Lin, Z.; Yu, Y.; Sun, R.; and Luo, Z.-Q. 2023. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*.

Liao, M.; Li, C.; Luo, W.; Jing, W.; and Fan, K. 2024. MARIO: MAth Reasoning with code Interpreter Output - A Reproducible Pipeline. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 905–924. Bangkok, Thailand: Association for Computational Linguistics.

Liu, C.; Yuan, Y.; Yin, Y.; Xu, Y.; Xu, X.; Chen, Z.; Wang, Y.; Shang, L.; Liu, Q. et al. 2025. Safe: Enhancing Mathematical Reasoning in Large Language Models via Retrospective Step-aware Formal Verification. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12171–12186. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Liu, X.; Wu, T.; and Guo, G. 2023. Adaptive Sparse ViT: Towards Learnable Adaptive Token Pruning by Fully Exploiting Self-Attention. In Elkind, E., ed., *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 1222–1230. International Joint

Conferences on Artificial Intelligence Organization. Main Track.

Liu, Y.; Gehrig, M.; Messikommer, N.; Cannici, M.; and Scaramuzza, D. 2023. Revisiting Token Pruning for Object Detection and Instance Segmentation. arXiv:2306.07050.

Meng, Y.; Xia, M.; and Chen, D. 2024. SimPO: Simple Preference Optimization with a Reference-Free Reward. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Pang, J.-C.; Wang, P.; Li, K.; Chen, X.-H.; Xu, J.; Zhang, Z.; and Yu, Y. 2024. Language Model Self-improvement by Reinforcement Learning Contemplation. In *The Twelfth International Conference on Learning Representations*.

Qwen; :; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C. et al. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sun, J.; Zheng, C.; Xie, E.; Liu, Z.; Chu, R.; Qiu, J.; Xu, J.; Ding, M.; Li, H. et al. 2024. A Survey of Reasoning with Foundation Models. arXiv:2312.11562.

Tang, Z.; Zhang, X.; Wang, B.; and Wei, F. 2024. MathScale: Scaling Instruction Tuning for Mathematical Reasoning. arXiv:2403.02884.

Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C. et al. 2025. Kimi k1.5: Scaling Reinforcement Learning with LLMs. arXiv:2501.12599.

Teng, F.; Yu, Z.; Shi, Q.; Zhang, J.; Wu, C.; and Luo, Y. 2025. Atom of Thoughts for Markov LLM Test-Time Scaling. arXiv:2502.12018.

Wang, S.; Yu, L.; Gao, C.; Zheng, C.; Liu, S.; Lu, R.; Dang, K.; Chen, X.; Yang, J. et al. 2025a. Beyond the 80/20 Rule: High-Entropy Minority Tokens Drive Effective Reinforcement Learning for LLM Reasoning. arXiv:2506.01939.

Wang, T.; Jiang, Z.; He, Z.; Tong, S.; Yang, W.; Zheng, Y.; Li, Z.; He, Z.; and Gong, H. 2025b. Towards Hierarchical Multi-Step Reward Models for Enhanced Reasoning in Large Language Models. arXiv:2503.13551.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; ichter, b.; Xia, F.; Chi, E.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 24824–24837. Curran Associates, Inc.

Wu, T.; Zhao, Y.; and Zheng, Z. 2024. An Efficient Recipe for Long Context Extension via Middle-Focused Positional Encoding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Wu, Z.; Zeng, Q.; Zhang, Z.; Tan, Z.; Shen, C.; and Jiang, M. 2025. Enhancing Mathematical Reasoning in LLMs by Stepwise Correction. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 21602–21623. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Xiong, W.; Zhang, H.; Ye, C.; Chen, L.; Jiang, N.; and Zhang, T. 2025. Self-rewarding correction for mathematical reasoning. arXiv:2502.19613.

Xu, H.; Mao, X.; Li, F.-L.; Wu, X.; Chen, W.; Zhang, W.; and Luu, A. T. 2025. Full-Step-DPO: Self-Supervised Preference Optimization with Step-wise Rewards for Mathematical Reasoning. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 24343–24356. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C. et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yax, N.; Anlló, H.; and Palminteri, S. 2024. Studying and improving reasoning in humans and machines. *Communications Psychology*, 2(1): 51.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G. et al. 2025. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. arXiv:2503.14476.

Zhang, K.; Yao, Q.; Lai, B.; Huang, J.; Fang, W.; Tao, D.; Song, M.; and Liu, S. 2025a. Reasoning with Reinforced Functional Token Tuning. arXiv:2502.13389.

Zhang, Z.; Zheng, C.; Wu, Y.; Zhang, B.; Lin, R.; Yu, B.; Liu, D.; Zhou, J.; and Lin, J. 2025b. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*.

Zhang, Z.; Zheng, C.; Wu, Y.; Zhang, B.; Lin, R.; Yu, B.; Liu, D.; Zhou, J.; and Lin, J. 2025c. The Lessons of Developing Process Reward Models in Mathematical Reasoning. *arXiv preprint arXiv:2501.07301*.

Zhong, T.; Liu, Z.; Pan, Y.; Zhang, Y.; Zhou, Y.; Liang, S.; Wu, Z.; Lyu, Y.; Shu, P. et al. 2025. Evaluation of OpenAI o1: Opportunities and Challenges of AGI. arXiv:2409.18486.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P. et al. 2023. LIMA: Less Is More for Alignment. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 55006–55021. Curran Associates, Inc.

Zhu, D.; Yang, N.; Wang, L.; Song, Y.; Wu, W.; Wei, F.; and Li, S. 2024. PoSE: Efficient Context Window Extension of LLMs via Positional Skip-wise Training. In *The Twelfth International Conference on Learning Representations*.