Logit Arithmetic Elicits Long Reasoning Capabilities Without Training

Anonymous ACL submission

Abstract

Large reasoning models (LRMs) can do complex reasoning via long chain-of-thought (CoT) involving cognitive strategies such as backtracking and self-correction. Recent studies suggest that some models inherently possess these long reasoning abilities, which may be unlocked via extra training. Our work investigates whether we can elicit such behavior without *any* training. To that goal, we propose a decoding-time approach, **THINKLOGIT**, which utilizes logits arithmetic (Liu et al., 2024) to tune a target large LM for long reasoning using a substantially smaller model as the guider. We then show that we can further boost its performance by training the guider model with preference optimization over correct/incorrect reasoning pairs sampled from both the target and guider model—a setup we refer to as **THINKLOGIT-DPO**. Our experiments demonstrate that THINKLOGIT and THINKLOGIT-DPO achieve a relative improvement in pass@1 by 24.5% and 29.1%, respectively, over five mathematical and scientific reasoning datasets using the Qwen2.5-32B when guided by R1-Distill-Qwen-1.5B—a model 21x smaller. Ablation studies confirm that THINKLOGIT-DPO succeeds only when it couples a preference-learning objective with training pairs drawn from both the target and guider models. Our work presents a computationally-efficient method to elicit long reasoning in large models with minimal or no additional training.

1 Introduction

Large reasoning models (LRMs), such as DeepSeek-R1 (DeepSeek-AI et al., 2025), OpenAI o1 (OpenAI, 2024), and Qwen3 (Qwen Team, 2025), have significantly advanced reasoning by leveraging inference-time compute (Snell et al., 2024; Brown et al., 2024). These models generate very long chain-of-thought (CoT) traces involving planning, reflection, and self-correction (Gandhi et al., 2025). It is widely believed that such behavior requires training, either through reinforcement learning with verifiable rewards (DeepSeek-AI et al., 2025; Lambert et al., 2024; Shao et al., 2024) or supervised distillation (Muennighoff et al., 2025; Li et al., 2025b). However, this training is costly due to the length of reasoning traces and extensive sampling. While such costs are prohibitive for large models, small models can be trained with modest compute (Dang and Ngo, 2025; Luo et al., 2025). This observation motivates our central research question: ***Can a small reasoning model elicit long CoT behavior in a larger model at inference time, without training the larger model?***

We address this question with a decoding-time technique, enabling a small reasoning model to guide a target model, mainly by manipulating its logits. Specifically, we use logit arithmetic (Liu et al., 2021, 2024; Mitchell et al., 2024; Fan et al., 2024a) to combine the output distributions of both models, allowing the target model to benefit from the guider model's long-chain-of-thought capabilities, without any additional training. We call this base approach **THINKLOGIT**. Furthermore, as the output distribution of both models may substantially differ, we align them by further training the small guider model. This training process uses Direct Preference Optimization (DPO; Rafailov et al., 2023) on preference pairs sampled from the guider and target models, thereby making THINKLOGIT more *on-policy*, and then applies logit arithmetic using the fine-tuned guider. We refer to this approach as **THINKLOGIT-DPO** and show that such training can further boost performance compared to THINKLOGIT.

We evaluate our methods on five challenging benchmarks covering mathematical and scientific reasoning. Our results show that fusing the logits of a small reasoning model (DeepSeek-R1-Distill-Qwen-1.5B) with those of a large target (Qwen2.5-32B) improves pass@1 by 24.5% with

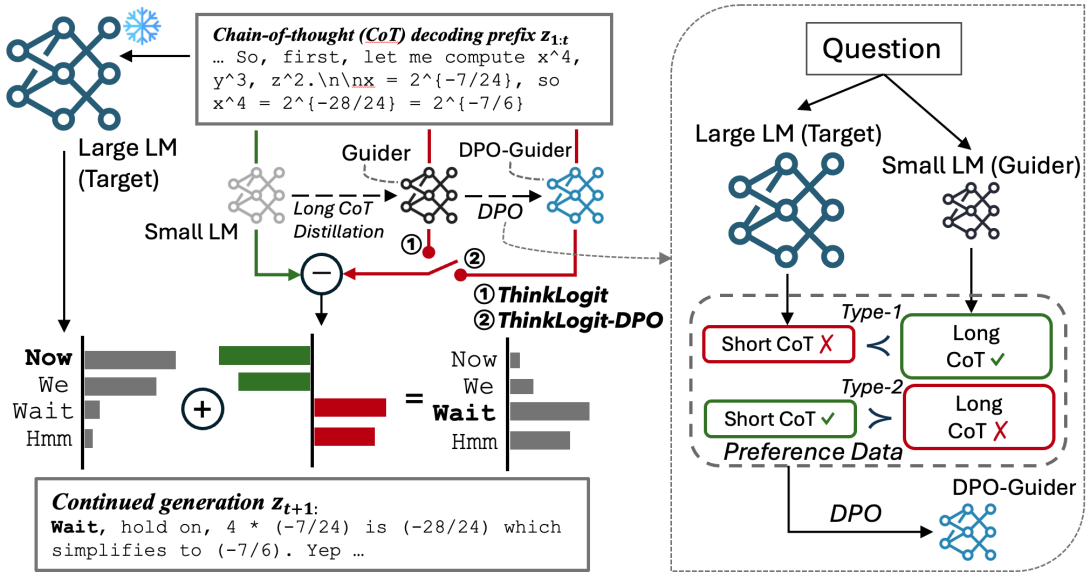


Figure 1: Overview of our THINKLOGIT and THINKLOGIT-DPO approaches to elicit long chain-of-thought reasoning from a large pre-trained model.

THINKLOGIT and by 29.1% with THINKLOGIT-DPO over the frozen target. We further demonstrate cross-domain (from math to scientific reasoning) and cross-model (from 32B to 72B target model) generalizability of our DPO-trained guider model. We also conduct extensive ablation studies on the preference-pair construction and alignment objectives of the guider, showing that including pairs with complementary strengths of both the guider and target, and optimizing via the DPO objective instead of supervised fine-tuning is essential for maximal performance gains. Our work sheds light on the power of small, trainable reasoning models to unlock long CoT behavior in much larger non-reasoning models at inference time, offering a practical and efficient alternative to costly model fine-tuning. Our main contributions are as follows.

- **A decoding-time approach (THINKLOGIT)** based on logit arithmetic (Liu et al., 2024) that injects long CoT guidance from a small, reasoning model into a frozen large LLM without any additional training.
- **An alignment enhancement (THINKLOGIT-DPO)** that uses Direct Preference Optimization to better match the guider's signals to the target distribution, yielding further gains.
- **Extensive ablations** on the training data and objectives in THINKLOGIT-DPO and hyperparameters in THINKLOGIT, demystifying our success recipe for long CoT elicitation and enhancement.

2 Methodology

Our goal is to elicit long CoT reasoning capabilities from a large, frozen language model without expensive training (see Figure 1). We introduce two lightweight decoding-time techniques: **THINKLOGIT**, which transfers long CoT behavior from a small guider via simple logit arithmetic (Liu et al., 2024), and **THINKLOGIT-DPO**, which further refines the guider using Direct Preference Optimization (DPO; Rafailov et al., 2023) to align its guidance with the target model.

2.1 THINKLOGIT

Let $\mathbf{z}_{1:t} = z_1, \ldots, z_t$ be the partially decoded sequence of reasoning tokens at step t. For any language model f, denote its pre-softmax logits at the next step by $\ell_{t+1}^{(f)} = f(\mathbf{z}_{1:t}) \in \mathbb{R}^{|\mathcal{V}|}$, where $\mathcal{V}$ is the vocabulary. We assume three models during inference:

- **large (target)** L, a pre-trained LLM lacking long CoT capability;
- **small base** S, a pre-trained model without reasoning fine-tuning;
- **small reasoning (guider)** $S^\star$, obtained via long CoT post-training to S.

At decoding step $t+1$, we compute $\tilde{\ell}_{t+1} = \ell_{t+1}^{(L)} + \alpha\big(\ell_{t+1}^{(S^\star)} - \ell_{t+1}^{(S)}\big)$, where $\alpha \geq 0$ controls the guidance strength. The delta term $\ell^{(S^\star)} - \ell^{(S)}$ encodes the probability shift that turns

a short-CoT model into a long-CoT one. Intuitively, adding this delta to L induces analogous long reasoning behavior without altering its weights.

Warm-up for Stable Decoding. We empirically observe that directly applying logit arithmetic at each decoding step would cause many repetitive generations in the long-CoT scenario. To stabilize generations, we defer guidance until a prefix of length T:

$$\tilde{\ell}_{t+1} = \begin{cases} \ell_{t+1}^{(L)}, & t+1 \le T, \\ \ell_{t+1}^{(L)} + \alpha\big(\ell_{t+1}^{(S^\star)} - \ell_{t+1}^{(S)}\big), & t+1 > T, \end{cases} \tag{1}$$

We set T=100 tokens in all experiments unless otherwise specified.

2.2 THINKLOGIT-DPO

The effectiveness of THINKLOGIT can be limited by mismatches between the output distributions of the guider and target models. We therefore construct preference pairs that capture complementary strengths:

Type-1: $(x,\, y^{L\checkmark},\, y^{S\times})$ — The *large* model's correct (short) CoT is preferred over the *small* model's incorrect (long) one. This encourages the guider to preserve the correctness of the target model and avoid introducing new errors.

Type-2: $(x,\, y^{S\checkmark},\, y^{L\times})$ — The *small* model's correct (long) CoT is preferred over the *large* model's incorrect (short) one, teaching the guider to be more confident at fixing the large model's reasoning errors.

We gather these pairs from training queries x by independently sampling CoTs from L and $S^\star$ and labeling correctness based on the final answer. Let θ denote the parameters of the preference-optimized guider, initialized from $S^\star$. We train θ with a DPO objective function that mixes the two pair types:

$$\mathcal{L}_{\text{DPO}}(\theta) = \lambda\, \mathbb{E}_{(x,y^{L\checkmark},y^{S\times})\sim\mathcal{D}_1}\, \ell_\theta\big(x;\, y^{L\checkmark}, y^{S\times}\big) + (1-\lambda)\, \mathbb{E}_{(x,y^{S\checkmark},y^{L\times})\sim\mathcal{D}_2}\, \ell_\theta\big(x;\, y^{S\checkmark}, y^{L\times}\big), \tag{2}$$

where $\ell_\theta\big(x;\, y^+, y^-\big) = \log\sigma\big(r_\theta(x,y^+) - r_\theta(x,y^-)\big)$, σ is the sigmoid function, $r_\theta(x,y) = \beta[\log\pi_\theta(y \mid x) - \log\pi_{\text{ref}}(y \mid x)]$ is the implicit reward of trajectory y, and $\lambda \in [0,1]$ balances the two datasets $\mathcal{D}_1$ (Type-1) and $\mathcal{D}_2$ (Type-2). We use $\lambda{=}\frac{|\mathcal{D}_1|}{|\mathcal{D}_1|+|\mathcal{D}_2|}$ by default, directly concatenating two datasets as DPO training data without further rebalancing. After fine-tuning, we replace $S^\star$ in THINKLOGIT with the optimized guider to obtain THINKLOGIT-DPO.

3 Experiments and Results

3.1 Experimental Setup

Benchmarks. We evaluate models on five widely used reasoning benchmarks. Four of them are competition math problems sources from AIME2024 (30 problems), AIME2025 (30 problems), AMC23 (40 problems), and a subset of 134 hard problems (level 5) from MATH500 (Lightman et al., 2024). We also evaluate on another scientific reasoning dataset GPQA Diamond (Rein et al., 2023), consisting of 198 PhD-level science questions in Biology, Chemistry, and Physics. For each dataset, we independently sample 8 completions and compute pass@k (Chen et al., 2021). A problem is marked as solved if any of the k sampled outputs is correct, so pass@k helps reveal a model's potential to solve a problem. We primarily use pass@1 unless otherwise specified.

Models. Our major target model is **Qwen2.5-32B** (Yang et al., 2024a). We utilize a long CoT post-trained 1.5B models as the guider which is **R1-Distill-Qwen-1.5B** (DeepSeek-AI et al., 2025), a version based on Qwen2.5-Math-1.5B (Yang et al., 2024b) that has been supervised fine-tuned on 800K long-CoT examples distilled from DeepSeek-R1. Because all three models use the identical tokenizer, their output logits are directly comparable and can be combined arithmetically.

Preference Data Construction. We use the level 4–5 subset of the MATH training set (Hendrycks et al., 2021) and independently sample 5 completions from both the guider model ($S^\star$) and the target model (L). Each completion is checked for final-answer correctness against the gold label.[1]

The target model L yields 12,412 correct completions ($y^{L\checkmark}$) and 16,448 incorrect ones ($y^{L\times}$), whereas the guider $S^\star$ produces 18,651 correct ($y^{S\checkmark}$) and 10,209 incorrect ($y^{S\times}$) completions. Forming the Cartesian product for each question gives 11,974 Type-1 preference pairs $\big(y^{L\checkmark}, y^{S\times}\big)$ and 43,209 Type-2 pairs $\big(y^{S\checkmark}, y^{L\times}\big)$, for a total of

[1] We extract answers from `\boxed{}` and compute exact match with ground-truths based on this script https://github.com/openai/prm800k/blob/main/prm800k/grading/grader.py by (Lightman et al., 2024).

Model	# Training Examples	# Trainable Params	AIME 2024	AIME 2025	AMC 23	MATH Level 5	GPQA Diamond	Average
Qwen2.5-32B (***Target***)	-	-	14.6	8.3	**57.2**	44.7	**36.9**	32.3
R1-Distill-1.5B (***Guider***)	-	-	**16.2**	**18.8**	51.2	**47.5**	28.9	**32.5**
No Fine-tuning of the Target								
Target + THINKLOGIT	0	0	**22.5**	19.2	62.2	55.3	41.8	40.2
Target + THINKLOGIT-DPO	10K	78M	22.1	**21.7**	**63.7**	**58.5**	**42.4**	**41.7**
Full Fine-tuning of the Target								
s1.1-32B	1K	32B	32.9	25.4	70.0	72.2	51.9	44.5
R1-Distill-32B	800K	32B	**45.8**	**35.0**	**76.9**	**72.7**	**55.6**	**57.2**

Table 1: Comparison of **pass@1** performance across five reasoning benchmarks. The best results in each section are marked in **bold**. Key takeaways include: (1) fusing target and guider logits (THINKLOGIT) yields substantial accuracy gains on top of both models; (2) DPO-trained guider (THINKLOGIT-DPO) adds further improvement; (3) math-only guidance alignment transfers effectively to out-of-domain scientific reasoning (GPQA Diamond); (4) THINKLOGIT-DPO partially recovers benefits of full fine-tuning with fewer trainable parameters and less training data.

55,183 pairs. We then randomly select 10K preference pairs from the total 55K pairs for DPO fine-tuning. We applied LoRA (Hu et al., 2022) with a rank size of 64 for parameter-efficient fine-tuning of the guider model. For all models, decoding is performed with a temperature of 0.6, a maximum length of 8192 tokens, and guidance strength of $\alpha = 1$. More training details are in Appendix A.1.

3.2 Main Results

Table 1 presents the pass@1 scores for all systems. We highlight three key observations. *First,* THINKLOGIT *boosts reasoning accuracy upon both target and guider model, and* THINKLOGIT-DPO *raises it further.* Combining the logits of the 32B target with those of the 1.5B guider (THINKLOGIT) raises the average pass@1 by 24.5% relative to the frozen target and by 23.7% relative to the guider. Replacing the vanilla guider with the DPO-trained guider (THINKLOGIT-DPO) brings the relative improvement to 29.1% over the target model, without any extra inference cost. The performance gains are consistent across five tested datasets.

Second, a guider trained only on mathematics problems maintains effectiveness on out-of-domain scientific reasoning. Although the DPO alignment phase trains the 1.5B guider solely to mathematics problems, it still maintains and even slightly improves performance of vanilla logit arithimetic (THINKLOGIT) from 41.8 to 42.4 on the out-of-domain GPQA Diamond benchmark, which spans biology, chemistry, and physics. This indicates that while DPO shrinks the distribution gap between guider and target outputs on maths data, it also makes the guider's token-level probabilities easier for the target to follow regardless of subject matter. A recent study by Tang et al. (2025) also reports a similar domain-general nature of long CoT based on model representation analysis, reinforcing the potential for enhanced guidance from a math-trained guider to transfer broadly across disciplines.

Third, our approach recovers most of the benefit of full-parameter fine-tuning while touching only a small subset of weights and using far less data. With LoRA, we adjust just 78M adapter parameters and train on 10K preference pairs, yet THINKLOGIT-DPO closes 77% of the pass@1 gap between the frozen 32B target and the fully fine-tuned s1.1-32B (Muennighoff et al., 2025), which updates all 32B parameters using 1K carefully selected long-CoT examples from a corpus of 59K examples. Our pipeline does rely on a 1.5B guider (R1-distill-1.5B) that was already fine-tuned on 800K distilled examples from DeepSeek-R1, but training this smaller model is far cheaper than R1-distill-32B, and once trained, the same guider can be reused for many other larger targets, especially those in the same model family (e.g., Qwen2.5-72B, with results shown in Figure 4), at no extra cost. Consequently, the cumulative data and compute requirements of our pipeline remain well below those of fully fine-tuning large models, while still delivering substantial accuracy gains.

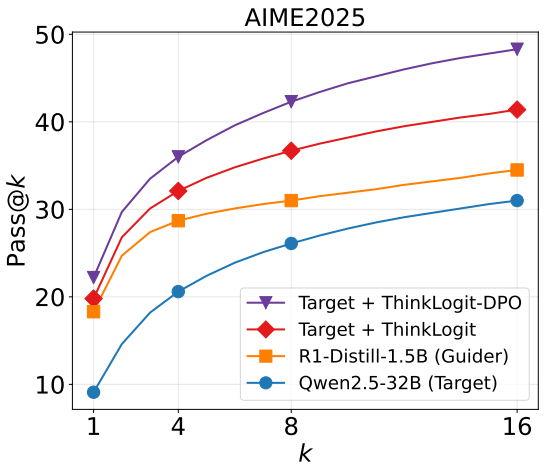


Figure 2: Test-time scaling on AIME2025. *Pass@ k* for $k = 1$–16 comparing the target, guider, their direct logit fusion (THINKLOGIT), and the DPO-aligned fusion (THINKLOGIT-DPO). *Our methods not only increase sample efficiency but also broaden the reasoning boundary of the target model.*

Figure 3: Comparison of our training-free long chain-of-thought elicitation method (THINKLOGIT) against two baselines: budget-forcing and one-shot long CoT in-context learning (ICL). The left panel shows pass@1 on AIME2025 and AMC23; the right panel shows the average chain-of-thought length in tokens. While budget-forcing and long CoT ICL increase verbosity, they degrade accuracy, whereas THINKLOGIT produces genuinely extended reasoning that boosts performance.

3.3 Test-Time Scaling Properties

Figure 2 plots *pass@k* for $k = 1$–16 on AIME2025, the dataset where the 32B target performs worst and scaling effects are therefore most visible. Both THINKLOGIT and THINKLOGIT-DPO surpass the target's pass@16 performance with only four generations, achieving a four-fold improvement in *sample efficiency*. The advantage widens as k grows: at $k = 16$ our DPO-aligned guider leads the target by roughly 17 points. Unlike the baseline's early plateau, our curve keeps rising, implying that logits guidance *broaden the reasoning boundary* (Yue et al., 2025b) rather than merely re-ranking similar completions.

3.4 Comparison with Other Training-Free Methods for Long CoT Elicitation

Figure 3 contrasts our approach against two training-free baselines for long chain-of-thought elicitation. First, the budget-forcing heuristic introduced by Muennighoff et al. (2025) replaces end-of-sentence tokens with a placeholder string like "Wait" to artificially increase output length. While this does produce longer completions, it consistently hurts performance, showing that verbosity alone does not lead to deeper reasoning.[2] Second, inserting a single long CoT example in the prompt (sampled from the s1.1-1K dataset (Muennighoff et al., 2025)) for in-context learning (ICL; Brown et al., 2020; Min et al., 2022; Dong et al., 2024) also degrades performance despite of longer outputs from the target model. In contrast, THINKLOGIT-DPO uses logit-level guidance from a small reasoning model to steer the decoding towards genuine long chain-of-thoughts, which translates into a clear uplift in downstream accuracy. This shows that our improvements stem from *the quality of the guidance* being applied, rather than *the quantity of tokens* generated.

3.5 Cross-Model Transferability of THINKLOGIT-DPO

We evaluate whether the THINKLOGIT-DPO guider, optimized for the reasoning preference of Qwen2.5-32B, can be applied off-the-shelf to a larger model (Qwen2.5-72B) in the same family. At inference time, we fuse the R1-distill-1.5B guider logits with the 72B target (THINKLOGIT) and then swap in the DPO-trained guider (THINKLOGIT-DPO), without any additional fine-tuning on the outputs from the 72B model. As highlighted in Figure 4, THINKLOGIT-DPO consistently improves upon THINKLOGIT, confirming that preference signals learned via DPO on a 32B model transfer effectively to larger scales, offering a plug-and-play mechanism to boost long-chain reasoning in even more capable LLMs within the same family.

[2] We note that while Muennighoff et al. (2025) demonstrate the effectiveness of budget-forcing on a Qwen2.5-32B-Instruct model fine-tuned on 1K long CoTs, they do not evaluate this technique directly on the untuned model.

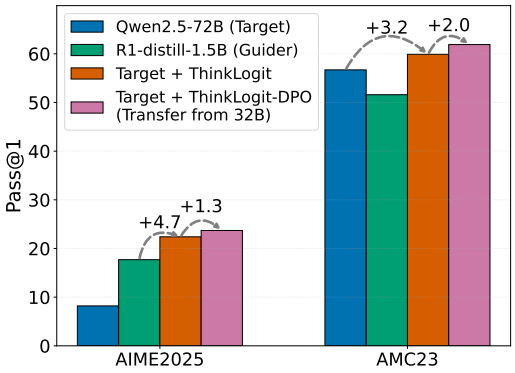


Figure 4: Pass@1 for eliciting long CoT in a 72B target model with logits arithmetic. THINKLOGIT-DPO delivers larger performance improvements on AIME2025 and AMC23 compared to THINKLOGIT, demonstrating that preference signals learned on a 32B model transfer effectively to a larger 72B model.

Model	Guider's Training Data	Pass@1
THINKLOGIT-DPO (ours)	$(x, y^{L\checkmark}, y^{S\times})$, $(x, y^{S\checkmark}, y^{L\times})$	**63.7**
THINKLOGIT-DPO		
w/o dual sources	$(x, y^{S\checkmark}, y^{S\times})$	58.8
w/o Type-1 pairs	$(x, y^{S\checkmark}, y^{L\times})$	57.2
w/o Type-2 pairs	$(x, y^{L\checkmark}, y^{S\times})$	51.9
THINKLOGIT-SFT		
learning from target	$(x, y^{L\checkmark})$	44.7
self-learning	$(x, y^{S\checkmark})$	55.6
learning from teacher	$(x, y^{R1\checkmark})$	60.9

Table 2: Pass@1 on AMC23 under ablations of guider's training data and objectives in THINKLOGIT-DPO. We compare the full DPO regime with mixed Type-1 and Type-2 pairs against single-source DPO (only guider outputs, only Type-1, only Type-2) and supervised fine-tuning variants. The dual-source, mixed-pair DPO yields the highest accuracy, demonstrating the necessity of complementary preference signals and preference-based alignment.

3.6 Ablation Study of THINKLOGIT-DPO

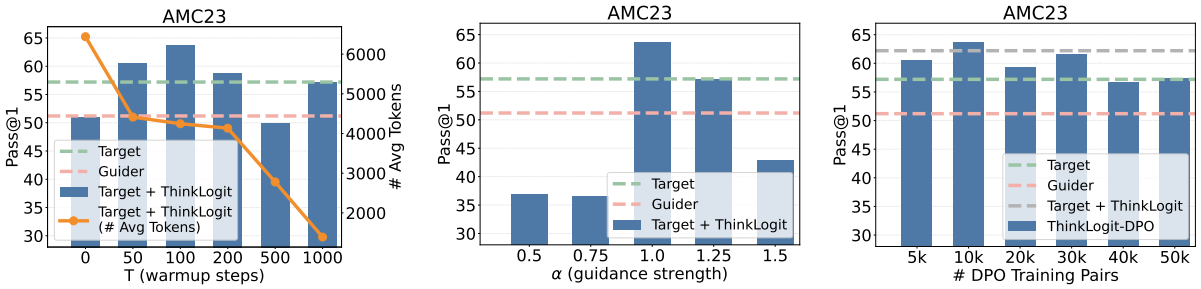
To further investigate the design choices in THINKLOGIT-DPO, we ablate both our mixed-pair data construction and preference-based learning objective (DPO) against single-source or supervised fine-tuning alternatives. Results in Table 2 answer the following research questions.

Are preference pairs sourced from both the target and the guider necessary to maximize performance? We construct the same amount of 10K preference pairs using only the guider's correct vs. incorrect outputs, i.e., $(x, y^{S\checkmark}, y^{S\times})$. DPO on this data underperforms markedly on AMC23 (58.8 vs. 63.7), confirming that mixing pairs which highlight *both* the target's and guider's strengths is crucial for maximal gains.

Is training on both types of pairs necessary for the effectiveness of THINKLOGIT-DPO? We next ablate by training on only one type of preference pairs at a time: using only Type-2 pairs $(x, y^{S\checkmark}, y^{L\times})$ (i.e., $\lambda = 0$ in Equation 2) yields a pass@1 of 57.2, while using only Type-1 pairs $(x, y^{L\checkmark}, y^{S\times})$ (i.e., $\lambda = 1$ in Equation 2) drops further to 51.9. Both are substantially below the 63.7 achieved by the full mixture, indicating that *both* Type-2 pairs (which teach the guider to correct target errors) and Type-1 pairs (which enforce preservation of correct target outputs) provide complementary signals necessary for optimal alignment.

Can supervised fine-tuning replace preference-based alignment of the guider? We evaluate standard supervised fine-tuning (SFT) against DPO by training the guider on three equally sized sets of high-quality completions: (1) the target model's correct outputs $y^{L\checkmark}$, (2) its own correct outputs $y^{S\checkmark}$ (also known as rejection-sampling fine-tuning (Yuan et al., 2023)), and (3) R1-distilled completions $y^{R1\checkmark}$. Although SFT on (1) and (2) turns the guider into a better *standalone* reasoner, none of these variants rival the performance of the DPO-aligned guider. This gap demonstrates that *optimizing with pairwise preference comparisons yields a better guider than optimizing solely for correctness*. While SFT can adapt the guider toward the target's short–CoT reasoning style in (3) and thus reduce the distributional gap, it tends to overwrite the guider's native strengths. In contrast, DPO *maintains the guider's intrinsic reasoning capabilities*—preserving its long reasoning capability—while *selectively aligning it to the target's preferences* through pairwise comparisons.

Overall, the best performance arises when the guider is aligned with the target via DPO and trained on a mixture of Type-1 and Type-2 preference pairs sourced from both models. Naive SFT—whether on the guider itself, the target's outputs, or a stronger teacher—fails to match these gains, underscoring key factors behind the effectiveness of THINKLOGIT-DPO.



(a) Varying the warm-up steps T reveals that applying guidance too early (i.e., T=0) leads to repeated generations and lower accuracy, while a moderate T improves pass@1 and stablize decoding.

(b) Sweeping the guidance strength α shows that $\alpha = 1.0$ yields the best trade-off between guider influence and target model priors.

(c) Increasing DPO training pairs from 5K to 10K improves pass@1, but further scaling leads to negative returns, likely due to redundancy and overfitting in preference data.

Figure 5: Impact of hyperparameter tuning on THINKLOGIT (Figure 5a and 5b) and training data size on THINKLOGIT-DPO (Figure 5c).

3.7 Impact of Hyperparameter Tuning on THINKLOGIT

In THINKLOGIT, two hyperparameters play a critical role in balancing stability, accuracy, and generation efficiency: the warm-up length T and the guidance strength α (Eq. 1). We evaluate their effects on the AMC23 benchmark, which presents a suitable mix of problem difficulties and clearly exhibits both stability and guidance effects.

To assess warm-up, we vary T over $\{0, 50, 100, 200, 500, 1000\}$ with α fixed to 1. For each possible values of T, we sample 8 completions per question and compute pass@1 alongside the average number of generated tokens (Figure 5a). When $T = 0$, guidance is applied immediately from the beginning, causing repetitive loops in early decoding and yielding the lowest accuracy. Allowing 50–200 tokens of unguided generation stabilizes the chain-of-thought, improving pass@1 over both target and guider models and reducing generation length. Increasing T beyond 200 causes the model to revert to the shorter CoTs typically produced by the target model, leading to an accuracy drop and shorter outputs.

With T fixed at 100, we sweep α over $\{0.5, 0.75, 1.0, 1.25, 1.5\}$ to control how strongly the guider's delta-logits modify the target's distribution (Figure 5b). At $\alpha = 1.0$, we observe the highest pass@1 together with moderate generation length, indicating an optimal trade-off between the guider's corrective signal and the target model's own priors. Therefore, we set $\alpha = 1.0$ as the default guidance strength. Future work might explore adaptive, context-aware schedules for T and α (Fan et al., 2024a) to optimize this trade-off further.

3.8 Impact of Training Data Size on THINKLOGIT-DPO

To determine the optimal number of DPO preference pairs, we randomly sampled subsets of 5K, 10K, 20K, 30K, 40K, and 50K from our full pool of 55K Type-1 and Type-2 pairs. Figure 5c plots pass@1 against training dataset size. With 5K pairs, THINKLOGIT-DPO 's pass@1 remains lower than that of the vanilla THINKLOGIT; increasing to 10K pairs raises pass@1 above THINKLOGIT while keeping training cost moderate. Beyond 10K pairs, adding more data leads to a decline in pass@1. Since our preference set is constructed via the Cartesian product of correct and incorrect generations, we hypothesize that pairs beyond 10K primarily recombine existing chains-of-thought rather than introduce new solution patterns, resulting in redundant examples and a higher risk of overfitting.

4 Related Work

4.1 Long Chain-of-Thought (CoT) Reasoning

Large reasoning models, such as OpenAI's o1 and o3 (OpenAI, 2024, 2025), DeepSeek-R1 (DeepSeek-AI et al., 2025), and QwQ (Team, 2025), achieve state-of-the-art results on mathematical and coding benchmarks by generating CoT traces that often extend to thousands of tokens, enabling systematic backtracking, verification, and self-reflection before a final answer is produced (Gandhi et al., 2025). One way to elicit such long-form reasoning is through **reinforcement learning with verifiable rewards** (Lambert et al., 2024). Pioneered by Group-Relative Policy Optimization (GRPO) (Shao et al., 2024) and refined by more stable and token-efficient variants such as DAPO (Yu

et al., 2025) and Dr. GRPO (Liu et al., 2025), this approach optimizes outcome-based rewards for correctness; nevertheless, mounting evidence shows that it mainly re-weights reasoning patterns already latent in the base model (Liu et al., 2025; Yue et al., 2025a). A complementary line of work demonstrates that the same capability can be acquired with **data-efficient supervised fine-tuning**. Distilled long CoTs from stronger teacher models allows a student to extend its reasoning length and thus improve accuracy using only about one thousand examples (Muennighoff et al., 2025; Xu et al., 2025; Ye et al., 2025; Li et al., 2025b). Finally, **training-free** methods exploit the fact that pretrained LLMs already exhibit long-CoT behaviours (Liu et al., 2025; Gandhi et al., 2025). (Tang et al., 2025) inject contrastive long- versus short-CoT representations into hidden states via representation engineering (Zou et al., 2023), whereas (Zhao et al., 2025) amplify a handful of key neurons at inference. Both techniques, however, require domain-specific long/short traces and white-box access, limiting their applicability in out-of-domain or black-box settings. THINKLOGIT sidesteps these constraints entirely. It keeps the target LLM frozen and, at inference time, fuses its logits with those of a lightweight "guider" model trained for long reasoning. This logit-fusion strategy recovers long-CoT behaviour induced by training-based methods while introducing no additional training cost or curated long-CoT examples.

4.2 Decoding Algorithms for LLM Reasoning

Decoding-time interventions offer an attractive alternative to full model fine-tuning: they can improve the reasoning capabilities of an off-the-shelf LLM with only a marginal increase in training or inference cost. The earliest line of work is Chain-of-Thought (CoT) prompting (Wei et al., 2022), which simply asks the model to "think aloud." Subsequent self-consistency decoding (Wang et al., 2023) samples a set of diverse CoTs and majority-votes over their answers, while later work shows that short CoTs can even be elicited *without* any prompting (Wang and Zhou, 2024). Crucially, these traces are usually brief: they march directly to the answer without back-tracking or verification, and therefore do not unlock the *long-form* reasoning studied in our work. A second family, **guided decoding**, biases generation toward correctness using either self-evaluation signals from the model itself (Xie et al., 2023) or an external discriminator (Khalifa et al., 2023). Accuracy is further improved by best-of-n reranking with discriminative reward models that score either the final answer or the reasoning process (Cobbe et al., 2021; Lightman et al., 2024; Wang et al., 2024). Generative reward objectives extend this idea and generalise better across tasks (Hosseini et al., 2024; Zhang et al., 2025; Wang et al., 2025; Khalifa et al., 2025). However, all of these methods depend on sampling many complete reasoning traces and scoring them *after* they are generated, which both raises costs and keeps them in the short-CoT regime. Auxiliary-model approaches modify the output of a frozen *target* model on the fly. Contrastive decoding subtracts logits from an "amateur" model or layer to suppress low-quality outputs (Li et al., 2023; Chuang et al., 2024), while speculative decoding speeds inference by letting a small draft model propose tokens that the expert later accepts or rejects (Leviathan et al., 2023; Yang et al., 2025; Liao et al., 2025). A closely related strand, **logits arithmetic**, blends the output distributions of three models token-by-token (Liu et al., 2021; Ormazabal et al., 2023; Shi et al., 2024), successfully emulating task-specific fine-tuning (Liu et al., 2024; Fan et al., 2024b), scaling laws (Mitchell et al., 2024), unlearning (Huang et al., 2025) and even overriding safety filters (Zhao et al., 2024). THINKLOGIT follows this lightweight pathway by using a compact guider model to unlock long-form reasoning in a frozen large model, while THINKLOGIT-DPO additionally aligns the guider's distribution with the target model's, delivering further gains.

5 Conclusion and Future Work

We introduce THINKLOGIT and THINKLOGIT-DPO, two decoding-time techniques that unlock long chain-of-thought (CoT) reasoning in frozen, non-reasoning LLMs. THINKLOGIT injects logits from a small, long-CoT guider, boosting accuracy by 24.5 % on five reasoning benchmarks for only a 1.1× increase in inference-time parameters, while THINKLOGIT-DPO aligns the guider with target distribution via Direct Preference Optimization for even higher gains. Together they offer a compute-efficient route to deploy long-CoT LLMs. Future work will combine heterogeneous model families, and develop context-aware guidance (e.g., adaptive strength α as in Fan et al. (2024a)) to mitigate the over-thinking problem in long reasoning (Chen et al., 2024).

Limitations

Inference-Time Overhead. Deploying THINKLOGIT requires hosting the large target model along with two smaller models—the base model S and the DPO-aligned guider $S^{\star}$. In our primary experimental setup (guiding a 32B target with a 1.5B guider), the total parameter count increases by approximately $1.1\times$ compared to using the frozen target alone. Instead of sequentially querying each model at every inference step, we implement asynchronous decoding to concurrently obtain logits from all three models. Profiling on NVIDIA L40S GPUs indicates a moderate inference slowdown (approximately 25% fewer tokens per second) compared to running only the target model. Since THINKLOGIT-DPO simply replaces the original guider model used in THINKLOGIT with a preference-optimized model of the same size, THINKLOGIT-DPO incurs no additional inference overhead beyond THINKLOGIT itself.

Same–Family Constraint. THINKLOGIT and THINKLOGIT-DPO computes *token-wise* differences between the guider's and a base model's logits and then adds that delta to the target model. Because the three models must share an identical vocabulary, we currently restrict all three to the same model family (e.g., Qwen2.5). Although Section 3.5 shows that a guider aligned on a 32B target transfers to a 72B target in the *same* family, we have not yet verified that the method generalizes to other model families. Extending the approach to heterogeneous families such as Mistral (Rastogi et al., 2025), Llama (Dubey et al., 2024), or Gemma (Kamath et al., 2025) will require a robust tokenizer alignment algorithm (Fu et al., 2023; Li et al., 2025a) to ensure delta logits remain semantically meaningful across models. We leave the design and empirical validation of such cross-family fusion to future work.

Limited Domains of Evaluation. Our experiments focus on math- and science-oriented reasoning tasks. A broader evaluation suite, including coding (Jimenez et al., 2023; Jain et al., 2025), planning (Zheng et al., 2024a; Xie et al., 2024), and tool-use (Huang et al., 2024; Patil et al., 2025), is needed to understand failure modes that may emerge in less structurally similar settings.

Offline alignment. The guider is aligned with the target via Direct Preference Optimisation (DPO) on a fixed set of preference pairs. This *offline* formulation cannot adapt once deployment uncovers new error patterns or distribution drift. Incorporating *online* reinforcement learning (Schulman et al., 2017; Shao et al., 2024) that updates the guider from streamed on-policy samples could, in principle, reduce this brittleness. However, on-policy RL introduces training efficiency and stability challenges that remain open research problems.

References

Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2024. Do NOT think that much for 2+3=? on the overthinking of o1-like llms. *CoRR*, abs/2412.21187.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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.

Quy-Anh Dang and Chris Ngo. 2025. Reinforcement learning for reasoning in small llms: What works and what doesn't. *arXiv preprint arXiv:2503.16219*.

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, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 1107–1128. 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, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Chenghao Fan, Zhenyi Lu, Wei Wei, Jie Tian, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024a. On giant's shoulders: Effortless weak to strong by dynamic logits fusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chenghao Fan, Zhenyi Lu, Wei Wei, Jie Tian, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024b. On giant's shoulders: Effortless weak to strong by dynamic logits fusion. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10421–10430. PMLR.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *CoRR*, abs/2503.01307.

Shousheng Jia Haosheng Zou, Xiaowei Lv and Xiangzheng Zhang. 2024. 360-llama-factory.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron C. Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners. *CoRR*, abs/2402.06457.

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.

James Y. Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2025. Offset unlearning for large language models. *Trans. Mach. Learn. Res.*, 2025.

Shijue Huang, Wanjun Zhong, Jianqiao Lu, Qi Zhu, Jiahui Gao, Weiwen Liu, Yutai Hou, Xingshan Zeng, Yasheng Wang, Lifeng Shang, Xin Jiang, Ruifeng Xu, and Qun Liu. 2024. Planning, creation, usage: Benchmarking llms for comprehensive tool utilization in real-world complex scenarios. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 4363–4400. Association for Computational Linguistics.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *CoRR*, abs/2310.06770.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, and 79 others. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *CoRR*, abs/2504.16828.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. GRACE: discriminator-guided chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 15299–15328. Association for Computational Linguistics.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Chong Li, Jiajun Zhang, and Chengqing Zong. 2025a. Tokalign: Efficient vocabulary adaptation via token alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 4109–4126. Association for Computational Linguistics.

Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhamaneshi, Shishir G. Patil, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025b. Llms can easily learn to reason from demonstrations structure, not content, is what matters! *CoRR*, abs/2502.07374.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. 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 12286–12312. Association for Computational Linguistics.

Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. 2025. Reward-guided speculative decoding for efficient LLM reasoning. *CoRR*, abs/2501.19324.

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.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024. Tuning language models by proxy. volume abs/2401.08565.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6691–6706. Association for Computational Linguistics.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog. Accessed 2025.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics.

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D. Manning. 2024. An emulator for fine-tuning large language models using small language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *CoRR*, abs/2501.19393.

OpenAI. 2024. Learning to reason with LLMs.

OpenAI. 2025. OpenAI o3 and o4-mini System Card. `https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf`. Accessed: 2025-07-05.

Aitor Ormazabal, Mikel Artetxe, and Eneko Agirre. 2023. Comblm: Adapting black-box language models through small fine-tuned models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2961–2974. Association for Computational Linguistics.

Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*.

Qwen Team. 2025. Qwen3: Think Deeper, Act Faster | Qwen.

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*.

Abhinav Rastogi, Albert Q. Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, Léonard Blier, Lucile Saulnier, Matthieu Dinot, Maxime Darrin, Neha Gupta, Roman Soletskyi, Sagar Vaze, Teven Le Scao, Yihan Wang, and 80 others. 2025. Magistral. *CoRR*, abs/2506.10910.

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.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Short Papers, NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 783–791. Association for Computational Linguistics.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314.

Xinyu Tang, Xiaolei Wang, Zhihao Lv, Yingqian Min, Wayne Xin Zhao, Binbin Hu, Ziqi Liu, and Zhiqiang Zhang. 2025. Unlocking general long chain-of-thought reasoning capabilities of large language models via representation engineering. *arXiv preprint arXiv:2503.11314*.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Chenglong Wang, Yang Gan, Yifu Huo, Yongyu Mu, Qiaozhi He, Murun Yang, Bei Li, Tong Xiao, Chunliang Zhang, Tongran Liu, and 1 others. 2025. Gram: A generative foundation reward model for reward generalization. *arXiv preprint arXiv:2506.14175*.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9426–9439. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

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*.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning. 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*.

Haotian Xu, Xing Wu, Weinong Wang, Zhongzhi Li, Da Zheng, Boyuan Chen, Yi Hu, Shijia Kang, Jiaming Ji, Yingying Zhang, Zhijiang Guo, Yaodong Yang, Muhan Zhang, and Debing Zhang. 2025. Redstar: Does scaling long-cot data unlock better slow-reasoning systems? *CoRR*, abs/2501.11284.

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, and 22 others. 2024a. 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. 2024b. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122.

Wang Yang, Xiang Yue, Vipin Chaudhary, and Xiaotian Han. 2025. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time. *CoRR*, abs/2504.12329.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. LIMO: less is more for reasoning. *CoRR*, abs/2502.03387.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *CoRR*, abs/2308.01825.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025a. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025b. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *CoRR*, abs/2504.13837.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025. Generative verifiers: Reward modeling as next-token prediction. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Xuandong Zhao, Xianjun Yang, Tianyu Pang, Chao Du, Lei Li, Yu-Xiang Wang, and William Yang Wang. 2024. Weak-to-strong jailbreaking on large language models. *CoRR*, abs/2401.17256.

Zekai Zhao, Qi Liu, Kun Zhou, Zihan Liu, Yifei Shao, Zhiting Hu, and Biwei Huang. 2025. Activation control for efficiently eliciting long chain-of-thought ability of language models. *arXiv preprint arXiv:2505.17697*.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, and Denny Zhou. 2024a. NATURAL PLAN: benchmarking llms on natural language planning. *CoRR*, abs/2406.04520.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024b. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Li Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405.

A Technical Details

A.1 Training Details

Environment. All experiments were conducted using NVIDIA A40/L40S GPUs with 48GB memory. The software environment was configured as follows:

- `360-LLaMA-Factory` (Haosheng Zou and Zhang, 2024) (A long-CoT adapted version of `LLaMA-Factory` 0.9.1 (Zheng et al., 2024b))
- `torch` 2.7.0
- `transformers` 4.51.3
- `accelerate` 1.0.1
- `datasets` 3.1.0
- `trl` 0.9.6
- `peft` 0.12.0
- `deepspeed` 0.14.4

LoRA Configuration. We applied LoRA (Hu et al., 2022) for parameter-efficient fine-tuning of the guider model:

- Rank: 64
- α_{LoRA}: 128
- Target modules: `q_proj`, `k_proj`, `v_proj`, `o_proj`
- Bias: None

DPO Training. For preference optimization with DPO, we used the following settings:

- Batch size: 32 (4 GPUs * 8 Gradient Accumulation)
- Epoch: 1
- Learning rate: 5e-6
- Optimizer: AdamW
- Learning rate scheduler: cosine with warmup
- Warmup ratio: 0.1
- β (reward scaling): 0.1
- Cutoff length: 8192