Thinkless: LLM Learns When to Think

Gongfan Fang **Xinyin Ma** **Xinchao Wang***
National University of Singapore
maxinyin@u.nus.edu, gongfan@u.nus.edu, xinchao@nus.edu.sg

Abstract

Reasoning Language Models, capable of extended chain-of-thought reasoning, have demonstrated remarkable performance on tasks requiring complex logical inference. However, applying elaborate reasoning for all queries often results in substantial computational inefficiencies, particularly when many problems admit straightforward solutions. This motivates an open question: Can LLMs learn when to think? To answer this, we propose Thinkless, a learnable framework that empowers an LLM to adaptively select between short-form and long-form reasoning, based on both task complexity and the model's ability. Thinkless is trained under a reinforcement learning paradigm and employs two control tokens, <*short*> for concise responses and <*think*> for detailed reasoning. At the core of our method is a Decoupled Group Relative Policy Optimization (DeGRPO) algorithm, which decomposes the learning objective of hybrid reasoning into two components: (1) a control token loss that governs the selection of the reasoning mode, and (2) a response loss that improves the accuracy of the generated answers. This decoupled formulation enables fine-grained control over the contributions of each objective, stabilizing training and effectively preventing collapse observed in vanilla GRPO. Empirically, on several benchmarks such as Minerva Algebra, MATH-500, and GSM8K, Thinkless is able to reduce the usage of long-chain thinking by 50% - 90%, significantly improving the efficiency of Reasoning Language Models. The code is available at `https://github.com/VainF/Thinkless`

1 Introduction

Reasoning Language Models have exhibited notable effectiveness in solving complex tasks that involve multi-hop reasoning and logic-intensive inference. Their capabilities span a range of domains, such as mathematical problem solving [17, 12, 30] and agentic assistants [39, 5]. A primary factor underlying this success is their ability to perform chain-of-thought reasoning [38], wherein intermediate steps are explicitly generated before arriving at a final answer. While this approach is effective for solving challenging problems, applying it uniformly to all queries, regardless of their complexity or the model's capability can be inefficient. In particular, for questions with straightforward solutions, invoking extended reasoning results in redundant token generation, increased memory footprint, and substantially higher computational cost [11, 7].

In response to these inefficiencies, a growing body of research has sought to enhance the inference efficiency of reasoning models [11, 1, 26, 24, 9, 14, 33]. A prominent direction in this space explores *hybrid reasoning* [4, 36, 2], wherein models dynamically switch between reasoning and non-reasoning modes [41, 2]. Despite promising results, a central challenge persists: determining when a model should engage in elaborate reasoning. Many existing approaches address this by incorporating manually designed heuristics, such as fixed computational budgets [1] or prompt-level control signals like "reasoning on/off" [4, 36]. However, these strategies inherently rely on human prior knowledge and may yield suboptimal or inappropriate control decisions. This underscores a fundamental open

*Corresponding author

Preprint. Under review.

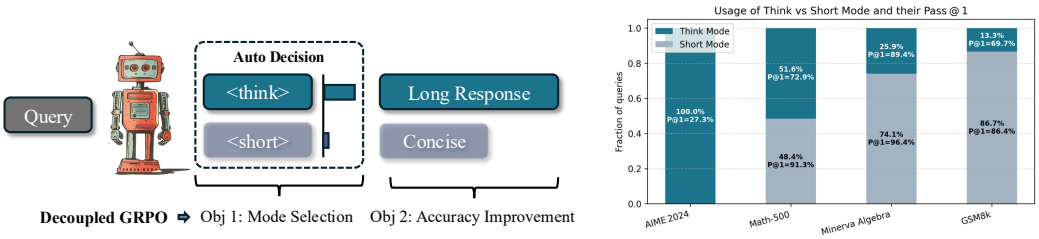


Figure 1: Thinkless learns a hybrid LLM capable of adaptively selecting between thinking and non-thinking inference modes, directed by two special tokens, <*think*> and <*short*>. At the core of our method is a Decoupled Group Relative Policy Optimization, which decomposes and balances the mode selection on the control token and accuracy improvement on the response tokens.

question: Can an LLM *learn* to decide when to think, guided by the complexity of the task and its own capability?

Motivated by this, we explore the fundamental form of hybrid reasoning, where the model is tasked with autonomously deciding whether to generate a short-form or long-form response based on the input query. This decision is guided by three core factors: (1) the complexity of the query, as simpler questions generally merit concise responses, while more intricate ones may necessitate extended reasoning; (2) the capability of the model, since more powerful models are better positioned to employ short reasoning without sacrificing accuracy, whereas less capable models may benefit from longer responses to preserve performance; and (3) the user's tolerance for the trade-off between efficiency and accuracy, which determines the acceptable level of performance degradation when opting for shorter reasoning. Naturally, reinforcement learning [30, 12, 43] offers a framework to unify these factors, as it allows the model to learn from interactions that reflect both environmental feedback and user-defined preferences. Through iterative exploration and reward-driven updates, the model progressively acquires the ability to make autonomous, context-aware decisions about its reasoning strategy, balancing accuracy and efficiency in a dynamic and data-driven manner.

Building on these insights, we propose Thinkless, a reinforcement learning framework designed to train a hybrid reasoning model capable of selecting between short-form and long-form responses. As illustrated in Figure 3, Thinkless employs two control tokens, <*think*> and <*short*>, which are generated as the first token in the model's output to signal the intended inference style. The training comprises two stages: a supervised warm-up phase followed by a reinforcement learning phase.

Distillation for Warm-up. In the warm-up phase, the model aligns its response style with the designated control tokens via a distillation process. Specifically, it learns to imitate the behavior of two expert models: a reasoning model and a standard instruction-following model, each conditioned on a specific control token (<*think*> or <*short*>). Additionally, the model is trained on paired long-form and short-form responses for each query, ensuring it can generate both styles with comparable likelihood. This initialization establishes a clear and robust mapping between control tokens and response formats, providing diverse outputs for subsequent reinforcement learning.

Reinforcement Learning with Decoupled GRPO. In the reinforcement learning phase, the model is optimized to select the appropriate inference mode based on performance feedback. A natural starting point for this task is the vanilla Group Relative Policy Optimization (GRPO) [12] framework. However, when applied to hybrid reasoning, vanilla GRPO treats all tokens, including the control token and the response tokens uniformly. This introduces a critical imbalance: since the response part often spans hundreds to thousands of tokens and the length of long & short responses varies significantly, the single control token may receive weak and biased gradient signals, ultimately leading to mode collapse at the early stages of training. To this end, we propose a tailored method for hybrid reasoning, termed Decoupled Group Relative Policy Optimization (DeGRPO). As illustrated in Figure 1, DeGRPO explicitly separates the hybrid reasoning objective into two components: (1) Mode Selection, which governs how quickly the policy adapts based on the model's current accuracy; and (2) Accuracy Improvement, which refines the response content to improve answer correctness under the selected reasoning mode. These two components are inherently interdependent, and effective training requires carefully balancing the learning signals for the control token and the response tokens. For instance, if the mode selection policy adapts too aggressively in favor of long-form reasoning, the

model may ignore short-form responses, resulting in insufficient exploration of the potential of short responses. To this end, DeGRPO assigns distinct weights to the control token and response tokens, promoting a more stable and balanced training dynamic. This design not only mitigates mode collapse but also enables the model to learn both accurate outputs and context-sensitive reasoning strategies. After reinforcement learning, the model learns to accurately identify simple queries and respond using the more efficient non-thinking mode. For instance, on benchmarks such as MATH-500, Minerva Algebra and GSM8K dataset, Thinkless reduces the usage of long-form reasoning by 50% - 90%. And on much more challenging tasks like AIME, the model naturally adopts a higher proportion of long-form reasoning.

In conclusion, this work demonstrates that a Reasoning Language Model can learn when to engage in reasoning before generating a response, guided by the proposed Decoupled GRPO method. This adaptive decision-making substantially reduces inference cost while preserving task performance.

2 Related Works

Efficient Reasoning Models. Reasoning models generate intermediate steps in a chain-of-thought process before producing a final answer [38, 17]. This paradigm offers significant advantages for tasks involving complex computations, logical deduction, and multi-step reasoning. However, excessively long reasoning chains can lead to substantial computational overhead [7, 9, 11]. To mitigate this, recent research has explored strategies to enhance the efficiency of reasoning models without sacrificing accuracy or generalization [26, 24, 1, 18, 13]. Several techniques, such as reinforcement learning with length penalties [1, 24], supervised fine-tuning using variable-length chain-of-thought data [26], and prompt-based methods [7, 40, 3] have been proposed to encourage concise yet effective reasoning paths. Additionally, latent reasoning techniques aim to encode reasoning steps into compact internal representations, thereby reducing token-level computation while maintaining performance [29]. Parallel efforts in knowledge distillation [16, 42, 27, 20, 6] have facilitated the transfer of reasoning capabilities from large to smaller models, while efficient decoding strategies such as predictive decoding and self-consistency optimization have also yielded notable improvements in inference speed and resource utilization [12, 19, 44].

Hybrid Reasoning. While prior work has largely focused on compressing reasoning paths to reduce token generation, an alternative path to efficiency is hybrid reasoning, which dynamically adapts the appropriate inference behaviour based on task complexity [2]. This approach allows models to flexibly alternate between short-form responses and long-chain reasoning as needed. Hybrid reasoning can be realized either through collaborative systems involving multiple models [28, 21] or within a single unified model [2, 36, 4]. In multi-model frameworks, routing mechanisms [28] or speculative decoding techniques [21] are commonly employed. For example, a lightweight model may generate a preliminary answer that a larger model verifies or refines. In contrast, unified models are trained to support both reasoning modes and can switch between them via prompt-based control. Some models adopt fixed prompt formats such as "reasoning on/off" to modulate reasoning depth [4]. However, most existing approaches depend on manually crafted heuristics to balance efficiency and performance. In this work, we explore a learning-based alternative, enabling an LLM to automatically determine its inference behavior based on inputs, without relying on manual control.

3 Method

The proposed Thinkless is implemented in two stages: (1) **Distillation for Warm-up**, where we fine-tune a pre-trained reasoning model to unify two reasoning styles, and (2) **Reinforcement Learning with DeGRPO**, where the model is trained under a decomposed objective to select the appropriate reasoning mode while improving the response quality.

3.1 Distillation for Warm-up

The first step in our framework is to construct a model π_θ capable of generating both short- and long-form responses. We leverage two pre-trained experts for distillation: a reasoning model π_{think}, trained to produce detailed chains of thought via step-by-step reasoning; and an instruction-following model π_{short}, optimized for generating concise answers aligned with user intent.

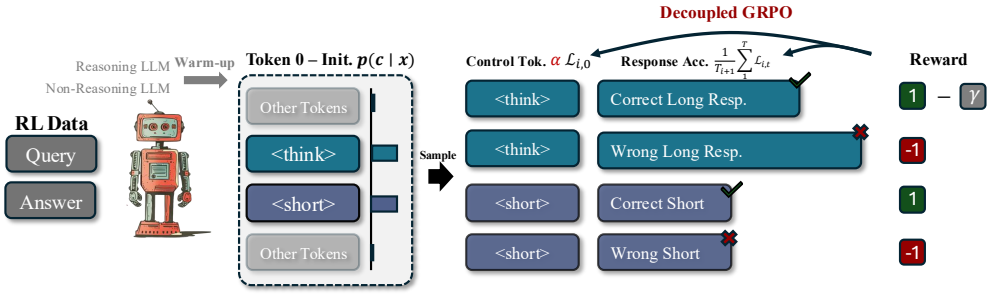


Figure 2: ThinkLess trains a hybrid model that adaptively selects reasoning modes based on task complexity and model capacity. The process begins with distillation, enabling the model to follow control tokens (<*think*> or <*short*>) for guided reasoning. This is followed by reinforcement learning using Decoupled GRPO, which separates training into two objectives: optimizing the control token for effective mode selection and refining the response to improve answer accuracy.

Given a prompt corpus $\mathcal{X} = \{x_i\}_{i=1}^{N}$, we use these models to generate a synthetic paired dataset:

$$\mathcal{D}_{\text{distill}} = \left\{ \left(x_i, \texttt{<think>}\, a_i^{\text{think}}, \texttt{<short>}\, a_i^{\text{short}}\right) \right\}_{i=1}^{N},$$

where $a_i^{\text{think}} = \pi_{\text{long}}(x_i)$ and $a_i^{\text{short}} = \pi_{\text{short}}(x_i)$. Each response is prefixed with a control token $c \in \mathcal{C} = \{\texttt{<short>}, \texttt{<think>}\}$ that conditions the model on the intended reasoning style. We then fine-tune the target reasoning model π_θ on this dataset via supervised fine-tuning (SFT). The objective is to learn a multi-style response distribution conditioned on the control token. This distillation phase ensures that the model is capable of generating both types of responses with high fidelity. Moreover, the paired construction of $\mathcal{D}_{\text{distill}}$ ensures that, the model's response distribution will be balanced. This helps the follow-up RL process to explore different solutions.

3.2 Learning When to Think via Decoupled GRPO

After the distillation phase, the model can produce both long- and short-form answers. what it still lacks is a mechanism for *deciding* which reasoning mode suits a particular input x. To supply this capability we frame mode selection as a reinforcement-learning problem and optimize a policy $\pi_\theta(c, a \mid x) = \pi_\theta(c \mid x)\, \pi_\theta(a \mid x, c)$, where the first token $c \in \mathcal{C} = \{\texttt{<short>}, \texttt{<think>}\}$ serves as a *control token* that determines the reasoning mode, and the subsequent tokens $(a_{i,1}, \ldots, a_{i,T_i})$ constitute the generated response. For notational convenience, we denote the entire sequence of the i-th sample of length $T_i + 1$ as $a_i = (a_{i,0}, \ldots, a_{i,T_i})$, where $a_{i,0} \in \mathcal{C}$ is the control token.

Reward Design. Let y^* denote the ground-truth answer corresponding to the input x. We consider a minimally designed reward function $r(a, y^*, c)$, which assigns different reward values as follows:

$$r(a, y^*, c) = \begin{cases} 1.0, & \text{if } c = \text{<\textit{short}>} \text{ and Extract-Answer}(a) = y^*, \\ 1.0 - \gamma, & \text{if } c = \text{<\textit{think}>} \text{ and Extract-Answer}(a) = y^*, \\ -1.0, & \text{if Extract-Answer}(a) \neq y^*, \end{cases}$$

where the $1 > \gamma > 0$ introduces preference to the short correct answer over long responses.

Decoupled Policy Optimization. Based on the simple reward function, we adopt a GRPO-based framework [30, 23] for training. Let $\{a_i\}_{i=1}^{G}$ denote a mini-batch of trajectories sampled from the current policy $\pi_{\theta_{\text{old}}}$. The objective is defined as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x, a_i}\left[\frac{1}{G} \sum_{i=1}^{G} \left(\frac{1}{T_i + 1} \sum_{t=0}^{T_i} \mathcal{L}_{i,t}(\theta) - \beta\, \mathbb{D}_{\text{KL}}\big[\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big] \right) \right], \tag{1}$$

where $\mathcal{L}_{i,t}(\theta)$ denotes the token-level surrogate loss formally given by:

$$\mathcal{L}_{i,t}(\theta) = \min\left(\frac{\pi_\theta(a_{i,t} \mid x, a_{i,<t})}{\pi_{\theta_{\text{old}}}(a_{i,t} \mid x, a_{i,<t})} \hat{A}_{i,t},\ \text{clip}\left(\frac{\pi_\theta(a_{i,t} \mid x, a_{i,<t})}{\pi_{\theta_{\text{old}}}(a_{i,t} \mid x, a_{i,<t})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_{i,t} \right) \tag{2}$$

In this work, we compute the relative advantage using $\hat{A}_{i,t} = r - \text{mean}(\mathbf{r})$, following [23]. This choice is motivated by the observation that our training data contains questions of varying difficulty, which can introduce bias when using standard deviation normalization.

When applied to training, the objective in Equation 1 serves two purposes: (1) to learn an appropriate control token for mode selection, and (2) to improve the accuracy of the response tokens:

$$\frac{1}{T_i+1}\sum_{t=0}^{T_i}\mathcal{L}_{i,t}(\theta) = \underbrace{\frac{1}{T_i+1}\mathcal{L}_{i,0}(\theta)}_{\text{Control Token}} + \underbrace{\frac{1}{T_i+1}\sum_{t=1}^{T_i}\mathcal{L}_{i,t}(\theta)}_{\text{Response Tokens}}. \tag{3}$$

For mode selection, the response style is conditioned on the first token $a_{i,0}$, which is trained during the preceding distillation stage. As a result, adjusting the probability of this single control token is sufficient to switch between reasoning modes. Therefore, this token controls the learning of inference mode. For response accuracy, the optimization seeks to improve the generation of the remaining tokens $a_{i,1:T_i}$. However, the above Equation 3 introduces two types of imbalance during optimization: (1) *Mode-Accuracy imbalance* - each trajectory contains only one control token but T_i response tokens, disproportionately reducing the influence of the mode selection compared to the optimization of response accuracy. (2) *Think-Short imbalance* - longer sequences ($T_i^{\text{think}} \gg T_i^{\text{short}}$) further suppress the gradient contribution of the control token due to the normalization factor $1/(T_i+1)$, causing the <*think*> token to be under-optimized compared to the <*short*>. As we will show in the experiment, such imbalance may lead to severe mode-collapse at the beginning of training. To mitigate these imbalances, we propose a decoupled variant of GRPO, denoted as $\mathcal{J}_{\text{DeGRPO}}$, which separately normalizes the contributions of the control and response tokens:

$$\mathcal{J}_{\text{DeGRPO}}(\theta) = \mathbb{E}_{x,a_i}\left[\frac{1}{G}\sum_{i=1}^{G}\Big(\underbrace{\alpha\,\mathcal{L}_{i,0}(\theta)}_{\text{Control Token}} + \underbrace{\frac{1}{T_i}\sum_{t=1}^{T_i}\mathcal{L}_{i,t}(\theta)}_{\text{Response Tokens}} - \beta\,\mathbb{D}_{\text{KL}}\big[\pi_\theta(\cdot\mid x)\,\|\,\pi_{\text{ref}}(\cdot\mid x)\big]\Big)\right], \tag{4}$$

In DeGRPO, the mode selection $\mathcal{L}_{i,0}(\theta)$ and response accuracy improvement $\sum_{t=1}^{T_i}\mathcal{L}_{i,t}(\theta)$ are independently normalized. A length-independent weighting coefficient α is introduced to balance the optimization between mode selection and response generation. This formulation ensures that the control token receives a consistent gradient scale across both short and long sequences, thereby addressing the mode-mode and think-short imbalance, enabling more stable optimization of reasoning-mode selection. As will be shown in the experiments, an appropriately large α can make the mode update more efficient. In our experiment, we set $\alpha = 1/1000$ for stable training.

Method Summary. In summary, our method retains the overall structure of the standard GRPO framework. For each query, a mini-batch of samples is drawn from the current policy to estimate the token-level advantages. To address the imbalance between mode selection and response generation, we independently normalize the advantages associated with the control token and response tokens. This separation allows for explicit balancing of their contributions during optimization, leading to more stable and effective training.

4 Experiments

4.1 Experimental Setups

LLMs and Datasets. We employ `DeepSeek-R1-Distill-Qwen-1.5B` as the base model to train a hybrid reasoning policy. To construct long-short paired responses for the distillation phase, we utilize long-form data from open-source datasets [35, 10] generated by the `DeepSeek-R1-671B` model [12], which is well-suited for multi-step reasoning. The corresponding short-form answers are derived using `Qwen2.5-Math-1.5B-Instruct` [41], a compact instruction-tuned model optimized for concise mathematical responses. The hybrid model is direcytly fine-tuned on this paired dataset via supervised fine-tuning, enabling it to accommodate both long and short reasoning styles. The model is then further optimized using the Decoupled Generalized Reweighted Policy Optimization (GRPO) algorithm. For the reinforcement learning stage, we primarily use the `DeepScaleR` dataset [25], which comprises approximately 40K labeled examples. For evaluation, we mainly focus on math datasets, including AIME [37], Minerva Algebra [15], MATH-500 [22] and GSM-8K [8].

Training Details. All experiments were conducted on a single node with 4 H100 GPUs. For the warmup stage, we set the maximum context length to 16K and clip those overlong samples. We

Models	Type	AIME 2024		Minerva Algebra		Math-500		GSM8K	
		Pass@1	#Tokens (Think%)	Pass@1	#Tokens (Think%)	Pass@1	#Tokens (Think%)	Pass@1	#Tokens (Think%)
DeepSeek-R1-1.5B	Base LLM	0.2800	18063	0.9577	3029	0.8608	5675	0.8347	1919
Q-1.5B		0.0200	1300	0.7771	933	0.5168	855	0.7022	466
QMath-1.5B		0.1133	1128	0.9184	586	0.7604	721	0.8572	447
Merging-0.5 [34]	Short CoT	0.1333	8636	0.9292	834	0.7740	1524	0.8332	601
Merging-0.6 [34]		0.1733	10615	0.9321	1091	0.7900	3000	0.8381	747
Merging-0.7 [34]		0.1667	15854	0.9398	1834	0.8108	4347	0.8458	1201
CoT-Valve $\alpha = 8$ **[26]**		0.2000	10692	0.8079	1903	0.7060	3723	0.7726	773
CoT-Valve $\alpha = 6$ **[26]**		0.1933	17245	0.9468	2656	0.8024	5167	0.7970	1009
CoT-Valve $\alpha = 4$ **[26]**		0.2267	17722	0.9439	2965	0.8036	5820	0.8108	1396
Router Random	Hybrid	0.1467	8093 (56.00%)	0.9211	1736 (49.28%)	0.7608	3096 (47.92%)	0.8205	1086 (50.99%)
Router Q-7B		0.1667	9296 (46.67%)	0.9250	795 (5.64%)	0.7948	2748 (25.00%)	**0.8587**	563 (2.35%)
Thinkless		**0.2733**	7099 (100.00%)	**0.9459**	1144 (25.88%)	**0.8184**	2555 (51.56%)	0.8418	624 (13.31%)

Table 1: Empirical results of hybrid reasoning. For hybrid algorithms, we additionally report the proportion of queries executed in the thinking mode during evaluation.

train the `DeepSeek-R1-Distill-Qwen-1.5B` for only 1 epoch. The SFT was conducted on the Megatron framework [32]. For the reinforcement learning stage, we extend the context length to 24K. The model was trained only for 600 steps, using the AdamW optimizer with a learning rate of 1×10^{-6}, $\beta = (0.9, 0.999)$, and a weight decay of 0.01. The batch size is set to 128, with 8 responses sampled for each query, leading to 1024 data points in total. The RL experiments were implemented using the VeRL framework [31]. More details can be found in the Appendix.

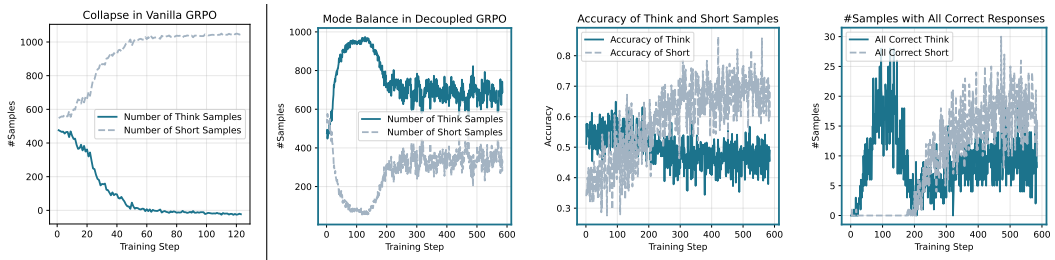
4.2 Empirical Results on Hybrid Reasoning

> ***Finding 1.*** The learned hybrid reasoning models effectively distinguish complex from simple queries, reducing the use of thinking by 50%–90%.

Table 1 presents a comparison between our method and several existing models or techniques. The first part showcases our baseline model, `DeepSeek-R1-Distill-Qwen-1.5B`, alongside two instruction-following models designed to generate concise answers. It can be observed that the number of tokens generated by reasoning models is typically 5 to 20 times higher than that of standard models. This highlights the potential for significantly improving efficiency by appropriately selecting the reasoning mode. Notably, on challenging datasets such as AIME and MATH-500, reasoning models tend to outperform others by a large margin. However, on simpler datasets like GSM-8K, extended reasoning offers no clear advantage, and the `Qwen2.5-Math-1.5B-Instruct` model with only a 4K context length achieves better results.

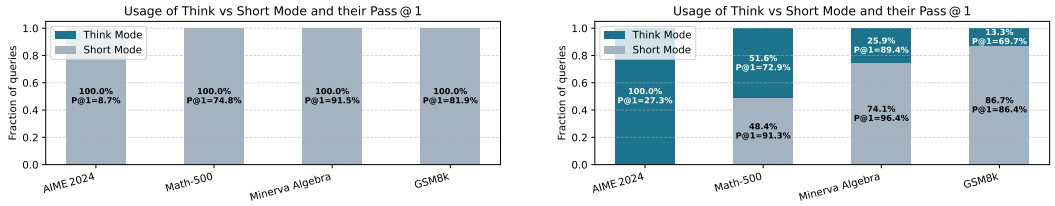
The second part of the table illustrates techniques for generating shorter chains of thought. One highly effective approach is model merging [34], where `DeepSeek-R1-Distill-Qwen-1.5B` is interpolated in parameter space with a base model, `Qwen2.5-Math-1.5B`, to obtain a more efficient model without additional training. Another method is the CoT-Valve [26] technique, which applies supervised fine-tuning (SFT) using LoRA. It allows for controllable reasoning length by adjusting the α parameter in LoRA to modulate the magnitude of parameter updates. Both methods provide mechanisms for adjusting reasoning length, such as the interpolation ratio and the LoRA α. To show their performance, we sample outputs across a range of lengths. A notable challenge with these methods is that the optimal reasoning length varies significantly across datasets. Consequently, when a good heuristic, such as merging coefficients of 0.6, is determined on one dataset to reduce token usage, it may result in unexpected performance degradation on other benchmarks, such as AIME.

In the final part of our study, we examine hybrid reasoning strategies, focusing on a comparison with router-based approaches. These methods employ a separate large language model (LLM) to assess query difficulty and dispatch inputs to either a reasoning model or a standard model. While effective to some extent, router models are typically independent and lack comprehensive awareness of the target reasoning models, limiting their ability to make confidence-informed decisions. For instance, on the challenging AIME dataset, where even the reasoning model achieves only 28% accuracy, the router struggles to recognize the its difficulty. In contrast, our method jointly considers both input complexity and model capability, dynamically refining the dispatch strategy through direct interaction with real examples. As a result, it achieves efficient and adaptive reasoning without manual tuning. On the Minerva Algebra dataset, our approach activates the reasoning mode for only 25% of the samples, reducing token usage to one-third of the original, while maintaining performance within a 1% margin. In addition, we found that the RL will also compress the length of long responses,



(a) Vanilla GRPO

(b) The proposed Decoupled GRPO, with a U-shape learning curve.



(c) Collapsed Policy

(d) Learned Policy of DeGRPO

Figure 3: Policy–training comparison between vanilla GRPO and decoupled GRPO.

since the algorithm will encourage short and correct answers, producing a gradient towards a more compact response.

4.3 Training Dynamics in RL

> ***Finding 2.*** Policy may collapse due to imbalanced update of control tokens in Vanilla GRPO.

Mode Collapse in RL. To further analyze how the model learns a reasonable policy, we visualize the training process of RL. Figure 3 (a) illustrates the *Mode Collapse* issue in standard GRPO, where the model develops an excessive preference for either long or short outputs during training. In conventional GRPO, the gradient on the control token is normalized by the total length of the response, which introduces an imbalance between long and short outputs. Specifically, long-chain samples, due to having more tokens, receive slower updates on the `<think>` token, while samples encouraging `<short>` dominate the updates. This imbalance causes the model to collapse rapidly, as shown in Figure 3 (a), the number of generated long-chain responses drops below 10 within just 120 update steps, making it difficult for the model to learn the correct policy. Furthermore, as shown in Figure 3 (c), the model fails to correctly differentiate between samples of varying difficulty, consistently opting for the short-chain reasoning mode.

> ***Finding 3.*** The U-shape learning curve: The proportion of short-chain samples first drops due to low initial accuracy, then rises after accuracy improvement and mode selection take effect.

The U-Shape Learning Curve. To mitigate the collapse issue, we propose a Decoupled GRPO algorithm. Figure 3 (b) illustrates the impact of this decoupling on the training process. We observe a characteristic U-shaped curve in the RL process: the proportion of long-chain outputs initially increases and then gradually decreases. After properly balancing the update weights between the control token and response tokens, the model shows a preference for long-chain reasoning in the early stages of training, primarily because long chains tend to yield higher accuracy. As training progresses, we observe an improvement in the accuracy of short-chain responses. This is driven by two factors: (1) reinforcement learning enhances the generation quality, and (2) the model learns to assign simpler queries to the short-chain reasoning mode. As a result, short-chain responses receive increasingly higher rewards, encouraging the model to further explore the feasibility of short reasoning. This behavior is manifested in the rising proportion of short-chain outputs over time, with many short responses achieving perfect accuracy in the later stages of training. Additionally, we observe a decline

Model	Mode & Teacher	AIME 2024		Minerva Algebra		Math-500		GSM8K	
		Pass@1	#Tokens	Pass@1	#Tokens	Pass@1	#Tokens	Pass@1	#Tokens
Qwen2.5-1.5B-Instruct	Short (Base)	0.0200	1300	0.7771	933	0.5168	855	0.7022	466
Qwen2.5-Math-1.5B-Instruct	Short (Base)	0.1133	1128	0.9184	586	0.7604	721	0.8572	447
DeepSeek-R1-1.5B	Long (Base)	0.2800	18063	0.9577	3029	0.8608	5675	0.8347	1919
R1-1.5B + OpenR1-97K	♦ Long (R1-671B)	0.2933	18880	0.9456	3239	0.8336	6780	**0.8382**	3423
	◊ Short (QMath-1.5B)	0.0733	3023	0.9001	642	0.7212	1004	0.7985	433
R1-1.5B + OpenThoughts-114K	♦ Long (R1-671B)	0.2600	17964	0.9506	3122	0.8280	6430	0.8039	2427
	◊ Short (QMath-1.5B)	0.0800	3807	0.8947	702	0.7136	1004	0.7886	449
R1-1.5B + OpenThoughts-1M	♦ Long (R1-671B)	**0.3067**	18685	**0.9530**	3064	**0.8360**	6209	0.8253	2363
	◊ Short (QMath-1.5B)	**0.0800**	3977	**0.9051**	690	**0.7276**	989	**0.8202**	441

Table 2: The effectiveness of different SFT datasets during the warm-up stage. Since these models have not yet been optimized via reinforcement learning, the control tokens <*think*>and <*short*> are manually inserted to elicit the desired response patterns.

in the accuracy of long-chain responses during the latter half of training. This phenomenon is not due to a degradation of the model's reasoning ability but rather a shift in task allocation: more difficult queries, which cannot be solved via short reasoning, are assigned to the long-chain mode, thereby lowering its average accuracy.

Finding 4. The weight α of the control token governs the learning speed of mode selection.

The Influence of Decoupling. Building upon the above analysis, we further visualize the effect of decoupling on model behavior. In Figure 4, we show the number of samples that are correctly answered with short responses across all sampled trajectories. We compare the results under two settings: a high control token update weight (0.5) and the weight used in our method (0.001). We observe that with a higher weight on the control token, the all-correct short samples emerge earlier in training. This is because the control token is updated more aggressively, allowing the model to focus more on learning the mode selection. However, excessively fast policy updates can be problematic. For example, some samples may initially yield low accuracy under short responses, but reinforcement learning could eventually improve their short-mode performance. If the model's strategy updates too quickly, it tends to assign such samples prematurely to the long-chain mode, degenerating the algorithm to a simple binary classifier based on initial accuracy rather than a collaborative learning of mode selection and accuracy improvement.

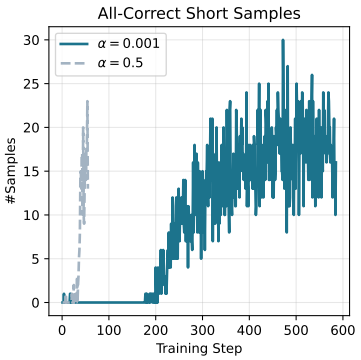


Figure 4: A large token loss coefficient α accelerates the shift in reasoning behavior, leading to the rapid emergence of all-correct short-mode samples.

4.4 Details of Warm-up Distillation

Finding 5. Reasoning LLMs can be a good short response learner.

In this work, knowledge distillation is deployed for warm-up, serving as a critical step to equip the model with the basic ability to generate both long chains and short responses. In this section, we provide additional implementation details. Specifically, we consider multiple datasets for distillation. We compare three datasets of increasing scale and domain coverage: (1) OpenR1, a mathematics-only dataset with rigorously verified solutions; (2) OpenThoughts-114K, a compact yet multi-domain dataset labeled by DeepSeek-R1-67B, covering mathematics, science, and programming; and (3) OpenThoughts-1M, a large-scale and diverse collection that subsumes the former two. Table 2 presents the training results on these datasets. Notably, we find that generating short responses using a long-chain reasoning model is relatively straightforward; even with the smallest dataset, OpenR1-97K, the target model successfully learns to produce short outputs. However, this distillation process may incur some performance degradation. For instance, on the Math-500 benchmark, the accuracy of the distilled hybrid model is slightly lower than that of the original model, which may affect the downstream performance during the RL phase. For the distillation stage, larger and more

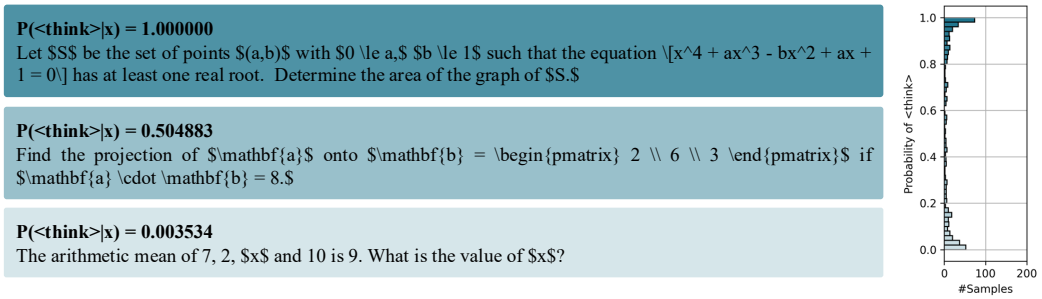


Figure 5: Distribution of the model's probability of emitting <*think*> on MATH-500. The samples with the highest, medium, and lowest probabilities are highlighted. The example with almost 0 thinking score mainly involves straightforward computation, and the query with 1.0 probability relies more on understanding and logical reasoning. More examples and LLM responses can be found in the appendix.

comprehensive datasets can lead to improved performance. However, the marginal gains diminish as the dataset size increases. For example, expanding the dataset from 114K to 1M results in only a 1% improvement in long-chain accuracy on the Math-500 benchmark. This work provides a preliminary validation of the effectiveness of simple distillation, and we leave the construction of stronger initial hybrid models as an important direction for future research.

4.5 Case Study

Figure 5 presents a case study on the model's predicted probability of selecting the `<think>` token across the MATH-500 dataset. The distribution reveals that the model makes smooth and hierarchical predictions for queries of different difficulty. In addition, we highlight representative examples corresponding to high, medium, and low confidence levels to illustrate the model's decision behavior. It can be observed that samples assigned to the short reasoning mode are typically simple arithmetic problems that do not require deep or complex reasoning. In contrast, questions routed to the thinking mode tend to be more complex, involving multiple conditions and concepts. Overall, the results reflect a well-calibrated policy that adapts reasoning depth based on task complexity.

5 Limitations and Future Works

This work presents an effective reinforcement learning framework that enables a hybrid model to adapt its inference mode based on both problem complexity and its own capabilities. However, several limitations remain. For instance, during the warm-up phase, we only validate a simple supervised fine-tuning (SFT) approach without extensive parameter tuning to achieve optimal performance, which results in a slight performance drop in the initial model for reinforcement learning. Exploring better strategies for constructing the hybrid model, such as merging techniques or lightweight fine-tuning methods like LoRA to mitigate catastrophic forgetting, could further enhance the overall performance. In addition, while our algorithm has been validated on DeepScaleR, a dataset containing 40K mathematical problems, future work could expand to a broader range of datasets, incorporating more diverse domains to enable more general and practical hybrid reasoning capabilities.

6 Conclusion

This paper proposes a reinforcement learning framework for building a hybrid reasoning model. It autonomously decides whether to generate a short response or engage in long-form reasoning based on the complexity of the input. The core of our approach is a Decoupled GRPO algorithm, which separates the reinforcement learning objective into two components: mode selection on the control token and accuracy improvement on the response tokens. This decoupling enables a more balanced contribution between the two learning objectives. Our method effectively reduces unnecessary long-form reasoning, thereby lowering overall system cost and improving user latency.

References

[1] Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

[2] Anthropic. Claude 3.7 Sonnet. `https://www.anthropic.com/claude/sonnet`, 2025. Accessed: 2025-05-10.

[3] Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. *arXiv preprint arXiv:2503.05179*, 2025.

[4] Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, Yian Zhang, Tugrul Konuk, Gerald Shen, Ameya Sunil Mahabaleshwarkar, Bilal Kartal, Yoshi Suhara, Olivier Delalleau, Zijia Chen, Zhilin Wang, David Mosallanezhad, Adi Renduchintala, Haifeng Qian, Dima Rekesh, Fei Jia, Somshubra Majumdar, Vahid Noroozi, Wasi Uddin Ahmad, Sean Narenthiran, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Siddhartha Jain, Igor Gitman, Ivan Moshkov, Wei Du, Shubham Toshniwal, George Armstrong, Branislav Kisacanin, Matvei Novikov, Daria Gitman, Evelina Bakhturina, Jane Polak Scowcroft, John Kamalu, Dan Su, Kezhi Kong, Markus Kliegl, Rabeeh Karimi, Ying Lin, Sanjeev Satheesh, Jupinder Parmar, Pritam Gundecha, Brandon Norick, Joseph Jennings, Shrimai Prabhumoye, Syeda Nahida Akter, Mostofa Patwary, Abhinav Khattar, Deepak Narayanan, Roger Waleffe, Jimmy Zhang, Bor-Yiing Su, Guyue Huang, Terry Kong, Parth Chadha, Sahil Jain, Christine Harvey, Elad Segal, Jining Huang, Sergey Kashirsky, Robert McQueen, Izzy Putterman, George Lam, Arun Venkatesan, Sherry Wu, Vinh Nguyen, Manoj Kilaru, Andrew Wang, Anna Warno, Abhilash Somasamudramath, Sandip Bhaskar, Maka Dong, Nave Assaf, Shahar Mor, Omer Ullman Argov, Scot Junkin, Oleksandr Romanenko, Pedro Larroy, Monika Katariya, Marco Rovinelli, Viji Balas, Nicholas Edelman, Anahita Bhiwandiwalla, Muthu Subramaniam, Smita Ithape, Karthik Ramamoorthy, Yuting Wu, Suguna Varshini Velury, Omri Almog, Joyjit Daw, Denys Fridman, Erick Galinkin, Michael Evans, Katherine Luna, Leon Derczynski, Nikki Pope, Eileen Long, Seth Schneider, Guillermo Siman, Tomasz Grzegorzek, Pablo Ribalta, Monika Katariya, Joey Conway, Trisha Saar, Ann Guan, Krzysztof Pawelec, Shyamala Prayaga, Oleksii Kuchaiev, Boris Ginsburg, Oluwatobi Olabiyi, Kari Briski, Jonathan Cohen, Bryan Catanzaro, Jonah Alben, Yonatan Geifman, Eric Chung, and Chris Alexiuk. Llama-nemotron: Efficient reasoning models, 2025.

[5] Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, et al. Research: Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025.

[6] Xiaoshu Chen, Sihang Zhou, Ke Liang, and Xinwang Liu. Distilling reasoning ability from large language models with adaptive thinking. *arXiv preprint arXiv:2404.09170*, 2024.

[7] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

[8] 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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.

[10] Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025.

[11] Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*, 2025.

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[13] Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*, 2024.

[14] Masoud Hashemi, Oluwanifemi Bamgbose, Sathwik Tejaswi Madhusudhan, Jishnu Sethumadhavan Nair, Aman Tiwari, and Vikas Yadav. Dna bench: When silence is smarter–benchmarking over-reasoning in reasoning llms. *arXiv preprint arXiv:2503.15793*, 2025.

[15] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[17] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[18] Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. C3ot: Generating shorter chain-of-thought without compromising effectiveness. *arXiv preprint arXiv:2412.11664*, 2024.

[19] Chenglin Li, Qianglong Chen, Liangyue Li, Caiyu Wang, Yicheng Li, Zulong Chen, and Yin Zhang. Mixed distillation helps smaller language model better reasoning. *arXiv preprint arXiv:2312.10730*, 2023.

[20] Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*, 2025.

[21] Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. Reward-guided speculative decoding for efficient llm reasoning. *arXiv preprint arXiv:2501.19324*, 2025.

[22] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

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

[24] Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.

[25] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 2025.

[26] Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*, 2025.

[27] Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*, 2022.

[28] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

[29] Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J Reddi. Reasoning with latent thoughts: On the power of looped transformers. 2025.

[30] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[31] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[32] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[33] Gaurav Srivastava, Shuxiang Cao, and Xuan Wang. Towards reasoning ability of small language models. *arXiv preprint arXiv:2502.11569*, 2025.

[34] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[35] Open Thoughts Team. Open thoughts, January 2025.

[36] Qwen Team. Qwen3, April 2025.

[37] Hemish Veeraboina. Aime problem set 1983-2024, 2023.

[38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[39] Junde Wu, Jiayuan Zhu, and Yuyuan Liu. Agentic reasoning: Reasoning llms with tools for the deep research. *arXiv preprint arXiv:2502.04644*, 2025.

[40] Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025.

[41] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[42] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*, 2024.

[43] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

[44] Xunyu Zhu, Jian Li, Can Ma, and Weiping Wang. Improving mathematical reasoning capabilities of small language models via feedback-driven distillation. *arXiv preprint arXiv:2411.14698*, 2024.