

Demystifying Hybrid Thinking: Can LLMs Truly Switch Between Think and No-Think?

Shouren Wang^{1,*}, Wang Yang^{1,*}, Xianxuan Long², Debargha Ganguly¹, Qifan Wang³
Vipin Chaudhary^{1,†}, Xiaotian Han^{1,†}

¹Case Western Reserve University ²Michigan State University ³Meta AI
{sxw992,wxy320,dxg512,vipin.chaudhary,xxh584}@case.edu
longxia2@msu.edu wqfcr@meta.com

*Equal contribution †Corresponding authors

Abstract

Hybrid thinking enables LLMs to switch between reasoning and direct answering, offering a balance between efficiency and reasoning capability. Yet our experiments reveal that current hybrid thinking LLMs only achieve partial mode separation: reasoning behaviors often leak into the no-think mode. To understand and mitigate this, we analyze the factors influencing controllability and identify four that matter most: (1) larger data scale, (2) using think and no-think answers from different questions rather than the same question, (3) a moderate increase in no-think data number, and (4) a two-phase strategy that first trains reasoning ability and then applies hybrid think training. Building on these findings, we propose a practical recipe that, compared to standard training, can maintain accuracy in both modes while significantly reducing no-think output length (from 1085 to 585 on MATH500) and occurrences of reasoning-supportive tokens such as “wait” (from 5917 to 522 on MATH500). Our findings highlight the limitations of current hybrid thinking and offer directions for strengthening its controllability. The code is available at: <https://github.com/SR-A-W/demystifying-hybrid-thinking>.

1 Introduction

Hybrid thinking is a widely adopted approach in many company models, enabling control over whether the model engages in reasoning and thereby achieving a more efficient and flexible reasoning process. Examples include Gemini (Cohan et al., 2025), GPT-oss (Agarwal et al., 2025), Qwen3 (Yang et al., 2025a), and DeepSeek V3.1 (DeepSeek-AI, 2025). However, few studies have systematically investigated the capabilities of hybrid thinking models, such as the training factors that influence them and the limitations they face in practice. This paper fills this gap by 1) offering a comprehensive analysis of hybrid thinking and 2)

 Question: Please reason step by step and put your final answer within boxed(). Let triangle ABC be inscribed in a circle ω . The tangents to ω at B and C intersect at D . Let the line AD meet ω again at P . Given $AB = 5$, $BC = 9$, and $AC = 10$, suppose $AP = \frac{m}{n}$ in lowest terms. Find $m + n$.

Final Answer: 113

 **NO_THINK:**
<think>\n\n</think>\n\nWe are given triangle ABC inscribed in a circle ω ... Step 1: ... **Wait** — this is not correct ... Final Answer: 51

 **THINK:**
<think>\nOkay, so I need to solve this geometry problem. Let me try to understand it step by step ... First ... Hmm ... **Wait**, ... Thus, $AP = 100/13$, and since $m = 100$ and $n = 13$ are relatively prime, we compute $m + n = 100 + 13 = 113$

Figure 1: Case study on an AIME24 problem. We compare the responses of Qwen3-8B under no-think and think modes. In the no-think mode, Qwen3-8B still performs reasoning outside the no-think constraint (e.g., generating reasoning-supportive words such as **Wait**), indicating that its hybrid thinking ability remains imperfect and cannot achieve full control.

introducing a practical training recipe to enhance its controllability.

We begin by exploring the output differences of hybrid thinking models. Hybrid thinking is typically implemented by adding control tokens (e.g., `\no_think`, `\think`) into the prompt and applying supervised fine-tuning (SFT). Evaluating Qwen3-8B on MATH500, AIME24 and GPQA, with Qwen2.5-7B-Instruct as a pure no-think baseline, we find that hybrid models perform better in the think mode than in the no-think mode (e.g. 63% accuracy and 11,394 tokens in think vs. 24% and 4,062 in no-think), suggesting that they do learn to use control tokens to modulate outputs.

However, comparison with Qwen2.5-7B-Instruct shows that thinking mode switch is incomplete: Qwen2.5-7B-Instruct generates short outputs without reasoning. But, in the “*no_think*” mode,

Table 1: Performance on MATH500, AIME24, and GPQA under think and no-think modes. We report accuracy (Acc.), output length (Len.), #Wait count (i.e., occurrences of reflection words such as “wait”, “hmm”, “alternatively”, aggregated over all responses in the evaluation), and **no-think differences** (Δ) between Hybrid (Qwen3-8B) and Instruct (Qwen2.5-7B-Instruct). Notably, Hybrid model in the no-think mode produce much longer outputs than the Instruct model, and even generate reflection words.

Model	Acc. (%)			Len.			#Wait		
	Think	No-think	Δ	Think	No-think	Δ	Think	No-think	Δ
MATH500									
Instruct	–	59.94		–	703.11		–	0	
Hybrid	92.82	82.90	+22.96	4384.35	958.31	+255.20	83703	646	+646
AIME24									
Instruct	–	6.67		–	1729.22		–	0	
Hybrid	63.33	24.00	+17.33	11394.54	4061.67	+2332.45	12184	184	+184
GPQA									
Instruct	–	30.15		–	775.07		–	0	
Hybrid	59.14	47.93	+17.78	7451.07	1364.96	+589.89	66152	616	+616

Qwen3-8B still generate outputs with reasoning, which leaks reasoning words (e.g., “wait”, “hmm”) outside empty “<think>” blocks (Figure 1). Therefore, hybrid thinking affords only limited control, motivating further study of training strategies and their trade-offs.

To investigate which factors affect the controllability of hybrid thinking—i.e., the balance between think and no-think mode—we systematically analyze and obtain the following findings:

1. *Large-scale data enables effective hybrid switching.* A sufficiently large amount of hybrid thinking data (e.g., 140k samples) is required for stable control.
2. *No-think control improves with non-paired data.* Using think and no-think answers from different questions (*no-pairs*) yields stronger controllability than paired settings (think and no-think answers from the same questions).
3. *A moderate rise in no-think data strengthens control.* Appropriately increasing the no-think data proportion reduces output length in the no-think mode while maintaining accuracy.
4. *Two-phase training enhances no-think control.* First training on pure think data, then applying hybrid training, further improves the controllability of no-think mode.

Finally, we compare hybrid thinking models with those trained purely on think or no-think data,

under carefully matched training settings. Our results show that hybrid thinking cannot achieve complete mode separation, as the no-think mode remains influenced by think data. Based on our analysis of training factors, we propose a practical recipe to improve controllability in hybrid thinking, which maintains accuracy in both two modes while substantially reducing no-think verbosity. For example, on MATH500, the average no-think output length decreases from 1085 to 585 tokens, and the number of “wait” occurrences is reduced from 5917 to 522. These results demonstrate that our recipe effectively improves controllability without sacrificing performance, highlighting both the limitations of current hybrid thinking and concrete directions for its advancement.

We thus advocate that future hybrid thinking training should deliberately allocate appropriately more no-think data and adopt structured strategies such as two-phase training, which we believe will be crucial for advancing the development of controllable and efficient reasoning models.

2 Motivations: No-Think Mode is Still Thinking

The implementation of hybrid thinking is typically achieved by adding control tokens (e.g., “\no_think”, “\think”) into prompts and applying supervised fine-tuning (SFT), enabling models to seemingly decide whether to reason. To examine whether this provides true controllabil-

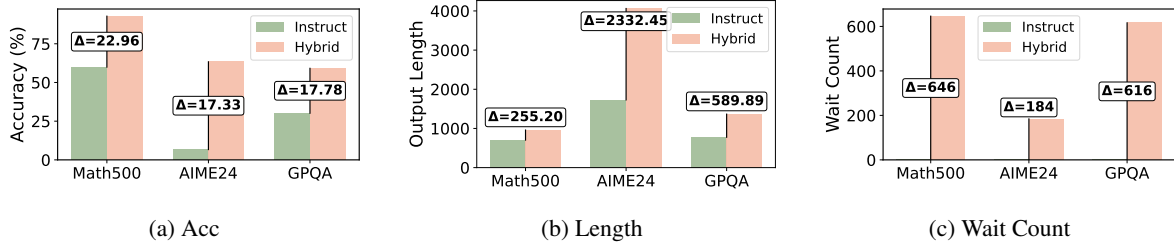


Figure 2: Comparison of the performance of **no-think** mode of Hybrid model (Qwen3-8B) and Instruct model (Qwen2.5-7B-Instruct) on MATH500, AIME24, and GPQA. We show bar plots of accuracy, average output length and wait count across these three datasets.

Table 2: Model-based judge validation of #Refl./Ans. as a genuine (not surface-level) leakage signal, on off-the-shelf Qwen3 hybrid baselines. Judge: GPT-OSS-20B, three-way no/implicit/explicit verdict.

Benchmark	Model	Mode	Output Len.	#Refl.	Judge verdict (agreement)
MATH500	Qwen3-4B	No-think	969	210	Implicit reasoning (100%)
	Qwen3-4B	Think	4750	163160	Explicit reasoning (98%)
	Qwen3-8B	No-think	1044	113	Implicit reasoning (92%)
	Qwen3-8B	Think	4419	154670	Explicit reasoning (98%)
	Qwen3-14B	No-think	885	131	Implicit reasoning (98%)
	Qwen3-14B	Think	4468	127342	Explicit reasoning (98%)
AIME24	Qwen3-4B	No-think	5022	672	Implicit reasoning (94%)
	Qwen3-4B	Think	11410	21481	Explicit reasoning (94%)
	Qwen3-8B	No-think	4677	24	Implicit reasoning (98%)
	Qwen3-8B	Think	11578	23222	Explicit reasoning (96%)
	Qwen3-14B	No-think	4013	74	Implicit reasoning (98%)
	Qwen3-14B	Think	11341	18814	Explicit reasoning (98%)
GPQA-Diamond	Qwen3-4B	No-think	1504	150	Implicit reasoning (100%)
	Qwen3-4B	Think	7615	132680	Explicit reasoning (100%)
	Qwen3-8B	No-think	1245	196	Implicit reasoning (96%)
	Qwen3-8B	Think	7481	131701	Explicit reasoning (98%)
	Qwen3-14B	No-think	1294	181	Implicit reasoning (98%)
	Qwen3-14B	Think	7246	105493	Explicit reasoning (96%)

ity, we evaluate Qwen3-8B (Yang et al., 2025a) on MATH500 (Lightman et al., 2024), AIME24, and GPQA (Rein et al., 2024), with Qwen2.5-7B-Instruct serving as a pure no-think baseline. Evaluation results in Table 1 show that hybrid models consistently perform better in the think mode than in the no-think mode (e.g., Qwen3-8B achieves 63% accuracy with an average length of 11,394 tokens in think mode, compared to 24% and 4,062 tokens in no-think mode on AIME24), confirming that they learn to use control tokens to modulate output style and length.

However, comparison with Qwen2.5-7B-Instruct reveals that mode separation remains incomplete. As shown in Figure 2 and Table 1, Qwen3-8B pro-

duces much longer no-think outputs (e.g., 4,062 tokens on AIME24 task, compared to Qwen2.5-7B-Instruct’s 1,729 tokens length), and **leaks reasoning-like behaviors** in its no-think responses (e.g., generating 646 “wait” tokens on AIME24 task, in contrast to Qwen2.5-7B-Instruct which operates strictly without reasoning traces).

As further illustrated in Figure 1, when answering an AIME24 problem, Qwen3-8B, despite leaving the “<think>” reasoning block empty in no-think mode, inserted reasoning phrases such as “wait” when summarizing before the final answer. To confirm that such reflection markers indicate genuine reasoning rather than surface artifacts, and that the pattern extends beyond this

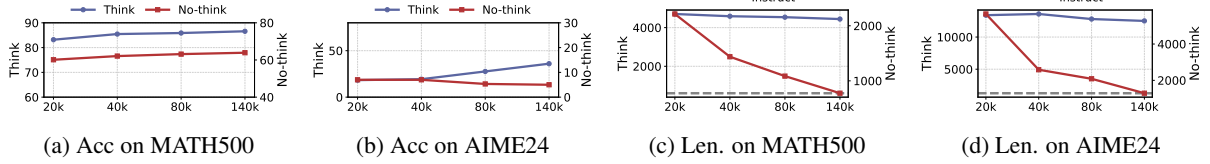


Figure 3: Line plots of Qwen2.5-7B-Instruct performance with different training data scales (20k/40k/80k/140k) on MATH500, AIME24, and GPQA. We report accuracy (top) and average output length (bottom) under both think and no-think modes. The results show that while accuracy and output length in the think mode, as well as accuracy in the no-think mode, remain relatively stable across scales, the no-think output length decreases substantially with larger training data scales as expected.

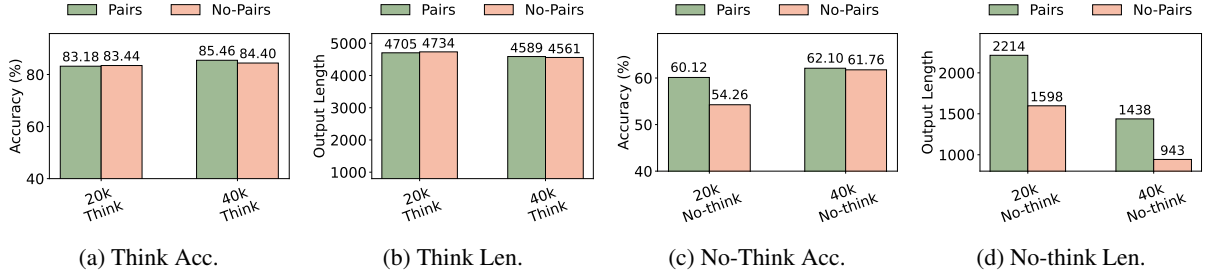


Figure 4: Comparison of accuracy and output lengths under think and no-think modes on MATH500. The bar charts illustrate model behavior under different training settings (pairs vs. no-pairs), highlighting how no-pairs could reduce output length in no-think mode.

single example, we further evaluate no-think outputs with an external LLM judge (GPT-OSS-20B) that classifies each response as *no reasoning*, *implicit reasoning* (the `<think></think>` block is empty but chain-of-thought reasoning or reflection is present in the visible output), or *explicit reasoning* (chain-of-thought is properly contained within the `<think></think>` block). As shown in Table 2, across Qwen3-4B/8B/14B on MATH500, AIME24, and GPQA, the judge labels 92–100% of no-think outputs as containing implicit reasoning, while think-mode outputs are labeled explicit reasoning 94–100% of the time.

Together, these findings indicate that hybrid models can only partially suppress reasoning and fail to achieve a clean separation between think and no-think modes. Motivated by this limitation, our work focuses on systematically exploring training strategies and trade-offs to strengthen mode controllability, aiming to achieve finer and more reliable separation between think and no-think behaviors.

3 What Factors Affect Mode Control in Hybrid Thinking Model Training

This section investigates training factors influencing the controllability of think and no-think mode within hybrid thinking. (1) *Data Scale*: effective control requires sufficiently large data (e.g., 140k samples). (2) *Pairing*: avoiding paired

think/no-think samples—i.e., using answers from different questions—slightly improves no-think controllability compared to paired data. (3) *Data Ratio*: while models easily learn the think mode, no-think is harder; moderately increasing its proportion enhances controllability. (4) *Two-Phase Training*: training first on pure think data and then applying think-mode fusion yields stronger no-think control than mixed training.

All experiments use the OpenR1-Math (Hugging Face, 2025) dataset, with no-think data from Numina-Math (LI et al., 2024) and think data generated by DeepSeek-R1 (Guo et al., 2025).

3.1 Large-Scale Data Facilitates Hybrid Thinking Mode Control

We first explore the impact of *data scale* on training hybrid thinking models. Specifically, we collect 20k, 40k, 80k, and 140k examples, ensuring that the ratio between think and no-think data is approximately 1:1. For each question, we include both a think-mode and a no-think-mode response. Comparative results for Qwen2.5-7B-Instruct are provided in Figure 3. For comprehensive and detailed experimental results, please refer to Tables 7 to 9 for Qwen2.5-7B-Instruct, and Tables 10 to 12 for Llama3.1-8B-Instruct in Appendix A.1.

We evaluate the model on MATH500, AIME24, and GPQA under both think and no-think modes.

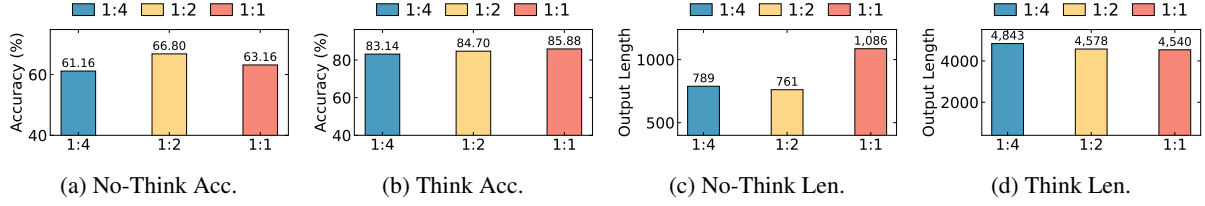


Figure 5: Comparison of output lengths under think and no-think modes on MATH500. The bar charts illustrate model behavior under different think-to-no-think data ratios, showing that increasing the proportion of no-think data reduces output length in the no-think mode.

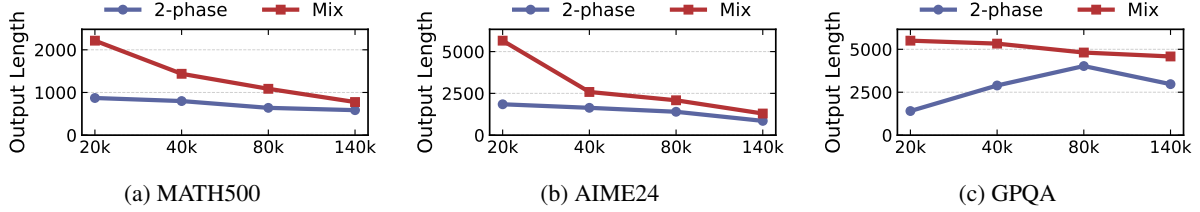


Figure 6: Average output length of models trained with 2-phase and mix strategies across three benchmarks (MATH500, AIME24, and GPQA) under different data scales with no-think mode.

The results show that as the training data scale increases, the accuracy in the think mode steadily improves, while the output length remains roughly unchanged. In contrast, the accuracy in the no-think mode remains relatively stable, but the output length decreases significantly. At 140k scale, the no-think output length is the shortest, approaching that of the baseline. For example, on MATH500, after training with 140k data, the model’s no-think output length is reduced to 775.67, which is nearly comparable to the baseline length of 608.85. This indicates that training with 140k examples serves as an effective data scale for enabling the model to achieve hybrid thinking.

3.2 Using Different Questions for Think and No-Think Improves Mode Control

We further investigate the impact of *With-Think vs No-Think Pairs* on hybrid thinking training. Using 20k and 40k training examples, we compare two settings: one where each question is paired with both a think-mode and a no-think-mode response, and another where the think and no-think responses are not paired to the same question.

As shown in Figure 4, consistent with our broader experimental results, we find that under the *no-pairs* setting, models produce substantially shorter outputs in the no-think mode, while maintaining almost the same accuracy in both think and no-think modes. For example, at the 40k scale, the think accuracy of pairs and no-pairs is 85 and 84 respectively, and the no-think accuracy is 62% and

61%. However, the no-think output length of the no-pairs setting is only 942, significantly shorter than that of the pairs setting. This suggests that when constructing datasets, it is preferable to use non-paired settings—i.e., ensuring that think and no-think responses do not originate from the same question. For comprehensive experimental results, please refer to Tables 13 to 16 for Qwen2.5-7B-Instruct and Llama3.1-8B-Instruct in Appendix A.2

3.3 Moderately Higher No-Think Data Proportions Improve Mode Control

Consistent with our earlier observations on data scaling in Section 3.1, we find that think-mode accuracy is relatively insensitive to scale, whereas no-think conciseness improves significantly. This suggests a practical lever for controllability: upweighting no-think data may strengthen the model’s ability to remain concise in the no-think mode, thereby improving overall hybrid-thinking control.

To validate this hypothesis, we fine-tune the Qwen2.5-7B-Instruct and Llama3.1-8B-Instruct model with 80k training examples while controlling the ratio between think and no-think data, and then evaluate its hybrid thinking ability on MATH500 and AIME24. As shown in Figure 5, with more comprehensive data in Tables 17 to 22 provided in Appendix A.3, we find that appropriately increasing the no-think data proportion improves no-think mode controllability, as evidenced by shorter outputs, while leaving the accuracy and output length of the think mode, as well as the

Table 3: Comparison of hybrid, pure-think, and pure-no-think models trained from Qwen2.5-7B-Instruct on MATH500 and AIME24 under think and no-think modes. We report accuracy (Acc.), output length (Len.), and #Wait count (i.e., occurrences of reflection words such as “wait”, “hmm”, “alternatively”). Results show that in the no-think mode, hybrid models still produce much longer outputs than pure no-think models and generate frequent reflection words.

Model Type	Think			No-think		
	Acc.	Len.	#Wait	Acc.	Len.	#Wait
MATH500						
Pure think	86.66	4340.62	99149	–	–	–
Pure no-think	–	–	–	53.10	430.44	0
Hybrid think	86.12	4539.38	105655	77.42	2971.03	64622
AIME24						
Pure think	33.33	12324.62	14460	–	–	–
Pure no-think	–	–	–	2.33	583.10	0
Hybrid think	29.33	12615.44	15217	5	853.82	1

accuracy of the no-think mode, largely unaffected.

3.4 Two-Phase Training: Training on Think Data Followed by Hybrid Thinking Training Enhances Mode Control

In the previous sections, we collected both think and no-think data and directly shuffled them for fine-tuning, enabling the model to acquire hybrid thinking ability, which we called as mix training. In contrast, following the training strategy of models such as Qwen3-32B (Yang et al., 2025a), hybrid thinking can also be achieved through a *two-phase training* process: first training the model on think-mode data to acquire reasoning ability, and then applying fine-tuning with both think and no-think data as a “Thinking Mode Fusion” phase to integrate the no-think capabilities.

To compare the effectiveness of two-phase training (think first, then think-mode fusion) against the random mixing strategy, we fine-tune the Qwen2.5-7B-Instruct and Llama3.1-8B-Instruct model with 80k examples, keeping the ratio of think and no-think pairs roughly 1:1 (an approximate proportion of 5:4). As shown in Figure 6, with the comprehensive experimental evidence detailed in Tables 23 to 26 provided in Appendix A.4, we observe that under different data scales, two-phase training consistently reduces output length in the no-think mode compared to mix training. For example, with 20k data, the no-think output length is reduced to 870 on MATH500 and 1847 on AIME24, compared to 2214 and 5654 under mix training.

The results demonstrate that two-phase training

effectively reduces the output length in the no-think mode while preserving the model’s think-mode accuracy. This suggests that two-phase training mitigates the influence of think-mode data on no-think outputs, thereby improving no think control.

We hypothesize this stems from *interference* between reasoning-acquisition and mode-formatting objectives under mix training: think-mode data provides dense, high-magnitude gradient signals that dominate training and impede learning the concise no-think behavior, consistent with the degraded no-think controllability of hybrid vs. pure no-think models in Section 4.1. We term this the *Interference Hypothesis*. Two-phase training mitigates it by disentangling the objectives temporally: Phase 1 (pure think) solidifies reasoning capability in isolation; Phase 2 (thinking-mode fusion) then only needs to learn *when* to reason rather than *how* and *when* simultaneously, freeing capacity for mode-routing and formatting. This explains why two-phase training yields shorter, more strictly formatted no-think outputs without sacrificing think-mode accuracy. This mirrors Qwen3’s reasoning-first-then-fusion order (Yang et al., 2025a); it remains a plausible interpretation, not one we have tested via dedicated ablation.

4 How Can We Achieve True Hybrid Thinking?

The previous sections mainly examined the training factors that influence hybrid thinking. In this section, we first compare hybrid thinking models with pure thinking and pure no-thinking models

to analyze the trade-offs inherent in hybrid thinking. We also evaluate open-source models such as Qwen3 for further insights. Building on these analyses, we then propose a training recipe inspired by the findings of the previous section, with the goal of enhancing control over the no-think mode.

Experimental Setups. We sample 80K examples from the *OpenR1-Math* default subset to construct three datasets: Pure think (only think responses), Pure no-Think (only no_think responses), and Hybrid think (a 5:4 mixture of think and no_think). *Qwen2.5-7B-Base* is fine-tuned on each dataset for 3 epochs with a learning rate of 1.0×10^{-5} and a warm-up ratio of 0.1, while all other optimization settings remain at their defaults.

4.1 Limitations of Current Hybrid Thinking Training on Mode Switching

Since current hybrid thinking models lack direct pure-think or pure no-think counterparts, we train three variants from the same base model: a hybrid model (80k hybrid data), a pure-think model (80k think data), and a pure no-think model (80k no-think data). As shown in Table 3, the hybrid model achieves nearly identical performance to the pure-think model in the think mode (e.g., MATH500 output length 4539 vs. 4340; AIME24 output length 12615 vs. 12324), suggesting that think-mode ability is unaffected. However, in the no-think mode, hybrid models achieve higher accuracy but generate substantially longer outputs than pure no-think models, indicating interference from think data that undermines controllability.

We further examine open-source hybrid models such as Qwen3 in Table 4. Even with large-scale training data, these models still exhibit reasoning-related behaviors in the no-think mode. To quantify this phenomenon, we count the occurrences of reasoning-supportive tokens (e.g., “wait”, “hmm”, “alternatively”) and observe frequent appearances across no-think outputs. For instance, all Qwen3 models (4B, 8B, and 14B) produce such tokens in the no-think setting.

These findings underscore a fundamental limitation of current hybrid thinking models: they only partially suppress reasoning and thus fail to guarantee clean separation between modes.

4.2 New Recipe for Improving No Think Control Based on Our Findings

Although hybrid thinking models cannot yet achieve 100% controllability over whether to en-

gage in reasoning, the analysis in previous sections suggests practical strategies to improve their performance. The results on MATH500 and MMLU-STEM are reported in Table 5.

Here we use the pure-think model and the Qwen2.5-7B-Instruct model as baselines, representing the best achievable performance in the think and no-think modes, respectively. Inspired by the findings in Section 3, we take the 80k mixed training as the original recipe and compare it with our new 2-phase recipe trained on 140k data with pairs. Here, our recipe adopts paired data as a trade-off: as discussed in Section 3.1, using unpaired think/no-think data can slightly improve controllability under the same data scale, but each instance is then used only once—either for think or for no-think—thus limiting the overall dataset size. We therefore prioritize enlarging the dataset at the cost of pairing, and the final results show that this scale-oriented recipe yields better overall performance. As shown, our recipe achieves nearly identical accuracy and output length to the original recipe in the think mode, but in the no-think mode it substantially reduces verbosity while maintaining accuracy—for example, on the MATH500 task, the average output length decreases from 1085 to 585 tokens, and the number of “wait” occurrences drops from 5917 to 522.

We additionally evaluate the recipe on the full MMLU benchmark (57 subjects) and TruthfulQA (Section D); reasoning leakage persists on general-knowledge and open-ended tasks beyond math, and our recipe’s no-think accuracy remains close to the pure-instruct baseline while reflective-token counts drop by more than an order of magnitude relative to pure reasoning-tuned models.

We assess the recipe’s stability and efficiency by repeating evaluation over 10 random seeds, reporting mean and standard deviation for output length and reflective-token count on MATH500 and AIME24, as well as wall-clock inference time and throughput on MATH500 (Section E); the low variance confirms these reductions are stable, and no-think inference is consistently faster than think-mode inference across model families.

5 Related Works

LLM Reasoning. Current approaches to enhancing reasoning in language models (Chen et al., 2026; Plaat et al., 2025; Sun et al., 2023) mainly fall into reinforcement learning (RL)

Table 4: Performance of Qwen3 models on MATH500 and AIME24 under think and no-think modes. We report accuracy (Acc.), output length (Len.), and #Wait count. Results show that even open-source models still produce such reflection words in the no-think mode.

Model	Mode	MATH500			AIME24		
		Acc.	Len.	#Wait	Acc.	Len.	#Wait
Qwen3-8B	Think	92.82	4384.35	83703	63.33	11394.54	12184
	No-think	82.90	958.31	646	24.00	4061.67	184
Qwen3-4B	Think	92.02	4679.54	97998	61.67	11595.51	13754
	No-think	82.22	1004.13	611	20.67	4636.37	762
Qwen3-14B	Think	93.96	4450.98	65320	67.67	11401.06	11091
	No-think	85.88	886.17	498	29.33	3497.86	295

Table 5: Performance of different training settings on MATH500 and MMLU-STEM under think and no-think modes. We report accuracy (Acc.), output length (Len.), and #Wait count (i.e., occurrences of reflection words such as “wait”, “hmm”, “alternatively”). Our recipe maintains accuracy while substantially reducing no-think verbosity and reflective tokens.

MATH500						
Model	Think			No-think		
	Acc.	Len.	#Wait	Acc.	Len.	#Wait
Pure-think	86.66	4340.62	99149	–	–	–
Instruct	–	–	–	75.42	640.91	0
Original Recipe	85.88	4539.53	116663	63.16	1085.99	5917
Our Recipe	86.78	4481.74	103335	63.60	585.63	522
MMLU-STEM						
Model	Think			No-think		
	Acc.	Len.	#Wait	Acc.	Len.	#Wait
Pure-think	84.97	2924.20	47382	–	–	–
Instruct	–	–	–	66.25	427.01	18
Original Recipe	84.68	3129.53	52665	80.97	2014.49	24653
Our Recipe	84.74	3141.14	50574	60.61	956.57	4017

(Schulman et al., 2017; Jaech et al., 2024) and supervised fine-tuning (SFT) (Yang et al., 2024). In RL, DeepSeek (Guo et al., 2025; Liu et al., 2024) achieved state-of-the-art performance with GRPO (Shao et al., 2024; Yu et al., 2025) and distilled reasoning traces to smaller models, inspiring replications such as Logic RL (Xie et al., 2025) and SimpleRL-Zoo (Zeng et al., 2025). RL and distillation effects on reasoning are also analyzed (Yeo et al., 2025; Baek and Tegmark, 2025). In SFT, curated reasoning traces are widely used, e.g., SkyThought-T1 (Team, 2025a) and Bespoke-Stratos-32B (Labs, 2025). Analyses show that structure of reasoning steps may matter more than content (Li et al., 2025a), while initial tokens play

a key role (Ji et al., 2025). Data selection is also critical, with s1 (Muennighoff et al., 2025) showing that small high-quality sets yield large gains. Other works probe complementary aspects of reasoning: CoT boundaries (Chen et al., 2024), distillation factors (Chen et al., 2025a), RL dynamics (Liu et al., 2025), and training-free multi-round thinking (Tian et al., 2025). Overall, RL and SFT provide complementary pathways to improve reasoning, highlighting the importance of training objectives, data structure, and sample quality.

Efficient Reasoning. Current reasoning models still face efficiency challenges, with many producing excessively long outputs (Bandyopadhyay et al., 2025; Zhang et al., 2026). Early efforts such

Table 6: The performance of proposed recipe across different models. We reported the performance of our recipe and the baseline pure think and pure no_think models for Qwen2.5-7B, Llama3.1-8B, and Phi-4-14B. The results shows that our recipe consistently enhances performance across various model architectures (Qwen, Llama, and Phi) and scaling sizes (ranging from 7B to 14B), confirming its robustness and cross-architecture effectiveness. These results are from a different evaluation run than those in Table 5.

Model	Mode	MATH500			MMLU-STEM		
		Acc.	Len.	#Wait	Acc.	Len.	#Wait
Qwen2.5-7B	Pure-think	86.06	4361.15	94997	85.79	2991.11	54345
	Instruct	75.42	640.91	0	66.25	427.01	18
	Our Recipe-think	86.50	4488.40	113183	85.82	3099.89	49397
	Our Recipe-no-think	63.56	593.09	1535	59.44	774.42	2334
Llama3.1-8B	Pure-think	87.34	3814.88	96024	66.95	2089.36	43584
	Instruct	49.50	2364.44	7	53.00	862.21	0
	Our Recipe-think	80.46	5077.65	144079	78.94	3871.83	73016
	Our Recipe-no-think	49.92	732.56	54	50.30	449.57	381
Phi-4-14B	Pure-think	92.00	3410.81	9502	70.57	2582.76	3213
	Instruct	79.98	704.79	0	61.66	784.97	1
	Our Recipe-think	92.48	3447.88	61706	93.63	2136.26	25489
	Our Recipe-no-think	67.76	530.52	3	63.50	345.99	29

as Kimi 1.5 (Team et al., 2025) and Sky-Thought (Team, 2025b) reduce verbosity by aligning long and short responses via preference optimization, while methods like TokenSkip (Xia et al., 2025) and LightThinker (Zhang et al., 2025a) improve efficiency by removing redundant tokens or compressing intermediate thoughts. Beyond shortening reasoning, hybrid thinking (Sui et al., 2025) aims to control *when* to reason by introducing control tokens (e.g., \think, \no_think), a strategy now adopted by models such as Gemini, Qwen3, and DeepSeek V3.1 . However, current implementations still show imperfect mode separation, often leaking reasoning behavior into no-think outputs. Other works explore efficiency from different angles, such as early stopping with probing (Fu et al., 2025) or analyzing overthinking and underthinking behaviors (Wang et al., 2025; Sui et al., 2025; Chen et al., 2025b).

Hybrid Thinking. Hybrid thinking has recently emerged as a mechanism to balance efficiency and reasoning. It was first popularized by Gemini (Comanici et al., 2025) through an adjustable thinking budget that controls how much reasoning the model performs before answering, and later extended by Qwen3 (Yang et al., 2025a) through explicit control tokens that switch between direct answering and reasoning across multiple model scales. GPT-oss (Agarwal et al., 2025) further explored vary-

ing degrees of reasoning depth, while DeepSeek V3.1 (DeepSeek-AI, 2025) further extended the paradigm to improve agentic capabilities and efficiency. A concurrent line of work addresses this at the architecture level: Path-Lock Expert (Wang et al., 2026b) enforces mode separation via a fixed per-sequence routing decision through an existing interface token, rather than a learned router. Despite these advances, little work has systematically examined the training factors and trade-offs underlying hybrid thinking.

6 Conclusion

In this paper, we systematically analyzed hybrid thinking models, which balance efficiency and reasoning by switching between think and no-think modes. Our evaluation shows that while hybrid thinking enables partial controllability, current models still fail to fully separate the two modes. Through controlled experiments, we identified four key factors—data scale, paired vs. unpaired answers, data ratio, and two-phase training—with larger no-think proportions and two-phase training proving most effective. We conclude that hybrid thinking entails inherent trade-offs compared to pure-think or pure-no-think models, and recommend allocating moderately more no-think data and refining training strategies to further improve controllability of hybrid thinking models.

Limitations

Our experiments are conducted primarily on earlier-generation models, including Qwen2.5 and Llama3.1, and do not yet cover the latest generation of models such as Qwen3. This choice is mainly due to practical considerations: at the time of our study, there’s no officially released text-only instruction-tuned Qwen3-Instruct model available, and directly comparing against Qwen3-base models may not constitute a fair or controlled evaluation. We leave a systematic investigation of newer-generation models, including Qwen3 and beyond, to future work, which will help further validate and refine our conclusions.

Acknowledgments

This work was supported in part by NSF awards 2117439 and 2112606.

The authors used LLM-based assistants to help with LaTeX editing, drafting and polishing text, proofreading, and code implementation; all content was reviewed and verified by the authors.

References

- Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). *arXiv preprint arXiv:2508.10925*.
- David D. Baek and Max Tegmark. 2025. [Towards understanding distilled reasoning models: A representational approach](#). *Preprint*, arXiv:2503.03730.
- Dibyanayan Bandyopadhyay, Soham Bhattacharjee, and Asif Ekbal. 2025. [Thinking machines: A survey of LLM based reasoning strategies](#). *arXiv preprint arXiv:2503.10814*.
- Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2026. [Towards reasoning era: A survey of long chain-of-thought for reasoning large language models](#). *Science China Information Sciences*, 69:161101.
- Qiguang Chen, Libo Qin, Jiaqi Wang, Jinxuan Zhou, and Wanxiang Che. 2024. [Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought](#). *Advances in Neural Information Processing Systems*, 37:54872–54904.
- Xinghao Chen, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, et al. 2025a. [Unveiling the key factors for distilling chain-of-thought reasoning](#). *arXiv preprint arXiv:2502.18001*, pages 15094–15119.
- 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. 2025b. [Do NOT think that much for 2+3=? on the overthinking of long reasoning models](#). In *Proceedings of the 42nd International Conference on Machine Learning*, pages 9487–9499.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *arXiv preprint arXiv:2507.06261*.
- Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. 2025. [The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks](#). *arXiv preprint arXiv:2502.08235*.
- DeepSeek-AI. 2025. [DeepSeek-V3.1](#). <https://huggingface.co/deepseek-ai/DeepSeek-V3.1>. Released 2025-08-21; accessed 2026-08-29.
- Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, Yian Ma, Aurick Qiao, Tajana Rosing, Ion Stoica, and Hao Zhang. 2025. [Efficiently scaling LLM reasoning with certain index](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 157566–157604.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. [DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning](#). *arXiv preprint arXiv:2501.12948*, pages 633–638.
- Mohsen Hariri, Weicon Chen, Nahal Shahini, Vikash Singh, Kai Ye, Amirhossein Samandar, Debargha Ganguly, Sreehari Sankar, Yanyan Zhang, Shouren Wang, Jerry Peng, Biyao Zhang, Michael Hinczewski, and Vipin Chaudhary. 2026a. [Test-time scaling in reasoning LLMs: Inference regimes, evaluation, and reproducibility](#). *arXiv preprint arXiv:2608.04001*.
- Mohsen Hariri, Michael Hinczewski, Jing Ma, and Vipin Chaudhary. 2026b. [Ranking reasoning LLMs under test-time scaling](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pages 33437–33478.
- Mohsen Hariri, Alan Luo, Weicon Chen, Tianyi Zhang, Qifan Wang, Xiaotian Han, and Vipin Chaudhary. 2026c. [Quantize what counts: More for keys, less for values](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 26393–26420, San Diego, California, United States. Association for Computational Linguistics.

- Hugging Face. 2025. [Open R1: A fully open reproduction of DeepSeek-R1](#).
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. [OpenAI o1 system card](#). *arXiv preprint arXiv:2412.16720*.
- Ke Ji, Jiahao Xu, Tian Liang, Qiuzhi Liu, Zhiwei He, Xingyu Chen, Xiaoyuan Liu, Zhijie Wang, Junying Chen, Benyou Wang, et al. 2025. [The first few tokens are all you need: An efficient and effective unsupervised prefix fine-tuning method for reasoning models](#). *arXiv preprint arXiv:2503.02875*, pages 10270–10290.
- Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. [LLM maybe LongLM: SelfExtend LLM context window without tuning](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 22099–22114. PMLR.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, pages 611–626.
- Bespoke Labs. 2025. [Bespoke-stratos: The unreasonable effectiveness of reasoning distillation](#). www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-distillation. Accessed: 2025-01-22.
- Ayeong Lee, Ethan Che, and Tianyi Peng. 2025. [How well do LLMs compress their own chain-of-thought? a token complexity approach](#). *arXiv preprint arXiv:2503.01141*.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. [Fast inference from transformers via speculative decoding](#). In *International Conference on Machine Learning*, pages 19274–19286. PMLR.
- Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hahmameshi, Shishir G. Patil, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025a. [LLMs can easily learn to reason from demonstrations structure, not content, is what matters!](#) *Preprint*, arXiv:2502.07374.
- Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. [NuminaMath](#).
- Runchao Li, Yao Fu, Mu Sheng, Xianxuan Long, Haotian Yu, and Pan Li. 2025b. [FAEDKV: Infinite-window Fourier transform for unbiased KV cache compression](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16856–16866, Suzhou, China. Association for Computational Linguistics.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri 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*.
- Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, et al. 2024. [DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model](#). *arXiv preprint arXiv:2405.04434*.
- 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*.
- Xianxuan Long, Zhikai Chen, Shenglai Zeng, Shouren Wang, Kai Guo, and Jiliang Tang. 2026. [MemTrace: Probing what final accuracy misses in long-term memory](#). *arXiv preprint arXiv:2606.17328*.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. [s1: Simple test-time scaling](#). *Preprint*, arXiv:2501.19393.
- Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Bäck. 2025. [Multi-step reasoning with large language models, a survey](#). *ACM Computing Surveys*, 58:1–35.
- 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](#). *Advances in Neural Information Processing Systems*, 36:53728–53741.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. [Gpqa: A graduate-level google-proof q&a benchmark](#). In *First Conference on Language Modeling*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. [DeepSeekMath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.

- Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. [Stop overthinking: A survey on efficient reasoning for large language models](#). *arXiv preprint arXiv:2503.16419*.
- Yuan Sui, Yufei He, Tri Cao, Sophia Simeng Han, Yulin Chen, and Bryan Hooi. 2026. [Meta-Reasoner: Dynamic guidance for optimized inference-time reasoning in large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 13268–13286.
- Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, et al. 2023. [A survey of reasoning with foundation models](#). *arXiv preprint arXiv:2312.11562*.
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. [Kimi k1.5: Scaling reinforcement learning with LLMs](#). *arXiv preprint arXiv:2501.12599*.
- NovaSky Team. 2025a. [Sky-T1: Train your own O1 preview model within \\$450](#). <https://novasky-ai.github.io/posts/sky-t1>. Accessed: 2025-01-09.
- NovaSky Team. 2025b. [Think less, achieve more: Cut reasoning costs by 50% without sacrificing accuracy](#). <https://novasky-ai.github.io/posts/reduce-overthinking>. Accessed: 2025-01-23.
- Xiaoyu Tian, Sitong Zhao, Haotian Wang, Shuaiting Chen, Yunjie Ji, Yiping Peng, Han Zhao, and Xiangang Li. 2025. [Think twice: Enhancing LLM reasoning by scaling multi-round test-time thinking](#). *Preprint*, arXiv:2503.19855.
- Shouren Wang. 2026. [Agent team work zone: An automated, persistent workspace for long-lived claude code agent teams](#). *arXiv preprint arXiv:2607.22917*.
- Shouren Wang, Debargha Ganguly, Vinooth Rao Kulka-rni, Wang Yang, Zhuoran Qiao, Daniel Blankenberg, Vipin Chaudhary, and Xiaotian Han. 2026a. [A survey on efficient protein language models](#). *Transactions on Machine Learning Research*. ISSN 2835-8856.
- Shouren Wang, Wang Yang, Chuang Ma, Debargha Ganguly, Vikash Singh, Chaoda Song, Xinpeng Li, Xianxuan Long, Vipin Chaudhary, and Xiaotian Han. 2026b. [Path-lock expert: Separating reasoning mode in hybrid thinking via architecture-level separation](#). In *Proceedings of the Conference on Language Modeling (COLM)*.
- Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. [Thoughts are all over the place: On the underthinking of o1-like LLMs](#). *Preprint*, arXiv:2501.18585.
- Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. 2025. [TokenSkip: Controllable chain-of-thought compression in LLMs](#). *Preprint*, arXiv:2502.12067.
- Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. [Logic-RL: Unleashing LLM reasoning with rule-based reinforcement learning](#). *Preprint*, arXiv:2502.14768.
- Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. [Chain of draft: Thinking faster by writing less](#). *arXiv preprint arXiv:2502.18600*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Wang Yang, Chaoda Song, Xinpeng Li, Debargha Ganguly, Chuang Ma, Shouren Wang, Zhihao Dou, Yuli Zhou, Vipin Chaudhary, and Xiaotian Han. 2026a. [AgentCE-Bench: Agent configurable evaluation with scalable horizons and controllable difficulty under lightweight environments](#). *arXiv preprint arXiv:2604.06111*.
- Wang Yang, Chaoda Song, Xinpeng Li, Shouren Wang, Nengbo Wang, Yanyan Zhang, Chuang Ma, Debargha Ganguly, Vikash Singh, Shuai Xu, Jing Ma, Yin Yu, Vipin Chaudhary, and Xiaotian Han. 2026b. [A survey on agent skills for LLMs: A lifecycle perspective from construction to ecosystems](#). *SSRN Electronic Journal*. SSRN preprint id 6746498.
- Wang Yang, Shouren Wang, Debargha Ganguly, Xinpeng Li, Chaoda Song, Vikash Singh, Vipin Chaudhary, and Xiaotian Han. 2026c. [Mid-think: Training-free intermediate-budget reasoning via token-level triggers](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 6024–6038, San Diego, California, United States. Association for Computational Linguistics.
- Wang Yang, Shouren Wang, Chaoda Song, Chuang Ma, Xinpeng Li, Nengbo Wang, Kaixiong Zhou, Vipin Chaudhary, and Xiaotian Han. 2026d. [When domains interact: Asymmetric and order-sensitive cross-domain effects in reinforcement learning for reasoning](#). *arXiv preprint arXiv:2602.01365*.
- Wang Yang, Xiang Yue, Vipin Chaudhary, and Xiaotian Han. 2025b. [Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time](#). In *Proceedings of the Conference on Language Modeling (COLM)*.
- Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. [Demystifying long chain-of-thought reasoning in LLMs](#). *Preprint*, arXiv:2502.03373.

- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. 2025. [DAPO: An open-source LLM reinforcement learning system at scale](#). *arXiv preprint arXiv:2503.14476*, pages 125532–125554.
- Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. [SimpleRL-Zoo: Investigating and taming zero reinforcement learning for open base models in the wild](#). *Preprint*, arXiv:2503.18892.
- Duzhen Zhang, Zhong-Zhi Li, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhijiang Guo, Le Song, and Cheng-Lin Liu. 2026. [From system 1 to system 2: A survey of reasoning large language models](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48:3335–3354.
- Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. 2025a. [LightThinker: Thinking step-by-step compression](#). *arXiv preprint arXiv:2502.15589*, pages 13307–13328.
- Tianyi Zhang, Mohsen Hariri, Shaochen Zhong, Vipin Chaudhary, Yang Sui, Xia Hu, and Anshumali Shrivastava. 2025b. [70% size, 100% accuracy: Lossless LLM compression for efficient GPU inference via dynamic-length float](#). In *Advances in Neural Information Processing Systems*, pages 109696–109724.

Appendix

A Additional Results on Factors Affect Hybrid Thinking

A.1 Results for Data Scales

Comprehensive results for models trained at different scales (20k, 40k, 80k, and 140k samples) are provided in [Tables 7 to 9](#) for Qwen2.5-7B-Instruct and [Tables 10 to 12](#) for Llama3.1-8B-Instruct.

As shown in [Tables 7 to 9](#), large-scale data is essential for effective control. While think-mode accuracy improves with scale (e.g., up to 86.58% on MATH500), the most notable impact is the reduction in no-think length—decreasing from 2214.08 to 775.67 tokens on MATH500 as scale increases. This consistency across architectures confirms that 140k samples are vital for achieving mode separation and minimizing reasoning leakage.

Table 7: Performance of **Qwen2.5-7B-Instruct** with different data scales on **MATH500**. “Instruct” refers to the original Qwen2.5-7B-Instruct model, while 20k/40k/80k/140k denote the number of training samples used. All training sets maintain a nearly 1:1 ratio of think to no-think data. We report accuracy (%) and average output length under both modes for each model.

Data Scale	Accuracy (%)		Avg. Output Len.	
	Think	No-think	Think	No-think
Instruct	–	74.96	–	608.85
20k	83.18	60.12	4704.91	2214.08
40k	85.46	62.10	4589.04	1437.72
80k	85.88	63.16	4539.53	1086.00
140k	86.58	63.90	4442.49	775.67

Table 8: Performance of **Qwen2.5-7B-Instruct** with different data scales on **AIME24**. “Instruct” refers to the original Qwen2.5-7B-Instruct model, while 20k/40k/80k/140k denote the number of training samples used. All training sets maintain a nearly 1:1 ratio of think to no-think data. We report accuracy (%) and average output length under both modes for each model.

Data Scale	Accuracy (%)		Avg. Output Len.	
	Think	No-think	Think	No-think
Instruct	–	13.33	–	1272.64
20k	18.33	7.00	13397.91	5654.04
40k	19.33	7.00	13586.36	2584.13
80k	27.67	5.33	12799.13	2086.05
140k	36.00	5.00	12507.64	1293.08

Table 9: Performance of **Qwen2.5-7B-Instruct** with different data scales on **GPQA**. “Instruct” refers to the original Qwen2.5-7B-Instruct model, while 20k/40k/80k/140k denote the number of training samples used. All training sets maintain a nearly 1:1 ratio of think to no-think data. We report accuracy (%) and average output length under both modes for each model.

Data Scale	Accuracy (%)		Avg. Output Len.	
	Think	No-think	Think	No-think
Instruct	–	31.01	–	977
20k	40.00	39.50	9444.03	5503.32
40k	40.61	38.99	9571.43	5330.81
80k	40.25	38.89	9712.00	4811.29
140k	41.01	34.95	9826.74	4583.18

Table 10: Performance of **Llama3.1-8B-Instruct** with different data scales on **MATH500**. “Instruct” refers to the original Llama3.1-8B-Instruct model, while 20k/40k/80k/140k denote the number of training samples used. All training sets maintain a nearly 1:1 ratio (an approximate proportion of 5:4) of think to no-think data. We report accuracy (%) and average output length under both modes for each model.

Data Scale	Accuracy (%)		Avg. Output Len.	
	Think	No-think	Think	No-think
Instruct	–	49.32	–	2426.36
20k	69.10	49.12	6008.2	1111.1
40k	73.40	53.92	5685.3	1118.5
80k	78.12	55.22	5263.7	899.7
140k	82.00	57.72	4896.8	780.9

Table 11: Performance of **Llama3.1-8B-Instruct** with different data scales on **AIME24**. “Instruct” refers to the original Llama3.1-8B-Instruct model, while 20k/40k/80k/140k denote the number of training samples used. All training sets maintain a nearly 1:1 ratio (an approximate proportion of 5:4) of think to no-think data. We report accuracy (%) and average output length under both modes for each model.

Data Scale	Accuracy (%)		Avg. Output Len.	
	Think	No-think	Think	No-think
Instruct	–	6.0	–	7153.73
20k	8.33	5.67	14182.8	2949.1
40k	10.00	4.00	14088.2	2995.5
80k	20.33	4.00	13253.2	2108.0
140k	23.67	2.67	13141.8	2229.5

Table 12: Performance of **Llama3.1-8B-Instruct** with different data scales on **GPQA**. “Instruct” refers to the original Llama3.1-8B-Instruct model, while 20k/40k/80k/140k denote the number of training samples used. All training sets maintain a nearly 1:1 ratio (an approximate proportion of 5:4) of think to no-think data. We report accuracy (%) and average output length under both modes for each model.

Data Scale	Accuracy (%)		Avg. Output Len.	
	Think	No-think	Think	No-think
Instruct	–	29.29	–	3190.01
20k	36.41	32.53	9119.5	2814.4
40k	36.52	30.35	10144.5	1957.3
80k	37.58	31.16	10004.7	987.2
140k	34.95	31.67	10190.1	3352.6

A.2 Results for Pairs and No-Pairs Data

Comprehensive results for models trained using data with pairs and without pairs are provided in [Tables 13 and 14](#) for Qwen2.5-7B-Instruct and [Tables 15 and 16](#) for Llama3.1-8B-Instruct.

Results across scales and architectures reinforce that non-paired responses improve no-think controllability. As shown in [Tables 13 and 14](#), the no-pairs setting consistently yields shorter no-think outputs (e.g., 942.87 vs. 1437.72 tokens at 40k scale on MATH500) with comparable accuracy. This consistent pattern across Qwen2.5 and Llama3.1 suggests that decoupling responses helps models better reduces reasoning leakage.

Table 13: Comparison between pairs and no-pairs on **MATH500** under think and no-think modes for **Qwen2.5-7B-Instruct**. *Pairs* means each question has both think and no-think responses; *no-pairs* means responses come from different questions. We report accuracy (%) and average output length. The no-pairs setting yields shorter no-think outputs across scales while maintaining comparable accuracy.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	pairs	83.18	4704.91	60.12	2214.08
	no-pairs	83.44	4733.91	54.26	1597.64
40k	pairs	85.46	4589.04	62.10	1437.72
	no-pairs	84.40	4560.66	61.76	942.87
80k	pairs	85.88	4539.53	63.16	1086.00
	no-pairs	86.6	4584.13	56.12	884.79

Table 14: Comparison between pairs and no-pairs on **AIME24** under think and no-think modes for **Qwen2.5-7B-Instruct**. *Pairs* means each question has both think and no-think responses; *no-pairs* means responses come from different questions. We report accuracy (%) and average output length. No-pairs setting yields shorter no-think outputs across scales while maintaining accuracy.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	pairs	18.33	13397.91	7.00	5654.04
	no-pairs	23.00	12913.75	5.33	2560.70
40k	pairs	19.33	13586.36	7.00	2584.13
	no-pairs	24.00	13308.04	7.67	1912.82
80k	pairs	27.67	12799.13	5.33	2086.05
	no-pairs	32.67	12765.81	4.0	1146.767

Table 15: Comparison between pairs and no-pairs on **MATH500** under think and no-think modes for **Llama3.1-8B-Instruct**. *Pairs* means each question has both think and no-think responses; *no-pairs* means responses come from different questions. We report accuracy (%) and average output length. The no-pairs setting yields shorter no-think outputs across scales while maintaining comparable accuracy.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	pairs	69.10	6008.17	49.12	1111.11
	no-pairs	69.50	6187.14	51.70	1159.26
40k	pairs	73.40	5685.25	53.92	1118.49
	no-pairs	73.52	5679.92	54.34	1096.61
80k	pairs	78.12	5263.71	55.22	899.72
	no-pairs	77.88	5353.67	56.32	864.91

A.3 Results for Think-to-No-Think Data Ratios

Comprehensive results for models trained using data at different think-to-no-think ratios (1:4, 1:2, 1:1, 2:1) are provided in [Tables 17 and 18](#) for Qwen2.5-7B-Instruct and [Tables 19 to 22](#) for Llama3.1-8B-Instruct respectively.

Experiments show that increasing no-think data proportions enhances control. Per [Tables 17 and 18](#), a 1:2 ratio reduces no-think length from 1086.00 to 761.30 tokens on MATH500 compared to 1:1, with stable think-mode accuracy. This confirms that upweighting no-think samples in the dataset is a practical lever for mode separation across various benchmarks and architectures.

Table 16: Comparison between pairs and no-pairs on **AIME24** under think and no-think modes for **Llama3.1-8B-Instruct**. *Pairs* means each question has both think and no-think responses; *no-pairs* means responses come from different questions. We report accuracy (%) and average output length. The no-pairs setting yields shorter no-think outputs across scales while maintaining comparable accuracy.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	pairs	8.33	14182.81	5.67	2949.06
	no-pairs	11.33	13880.53	3.33	2850.43
40k	pairs	10.00	14088.24	4.00	2995.47
	no-pairs	13.33	13926.53	3.00	3203.56
80k	pairs	20.33	13253.21	4.00	2108.03
	no-pairs	18.00	13437.94	2.67	1877.41

Table 17: Effect of different think-to-no-think data ratios on **MATH500** under think and no-think modes for **Qwen2.5-7B-Instruct**. Ratios denote the proportion of think to no-think samples, with the total training size fixed at 80k. Increasing the proportion of no-think data effectively reduces no-think output length while largely preserving accuracy in both modes. *Note: The “1:1” ratio is an approximate proportion ($\approx 5:4$) due to the design for dataset construction.*

Think:No-think	Think		No-think	
	Acc.	Len.	Acc.	Len.
1:4	83.14	4843.01	61.16	788.70
1:2	84.70	4577.90	66.80	761.30
1:1	85.88	4539.53	63.16	1086.00
2:1	86.6	4452.42	62.48	1121.09

Table 18: Effect of different think-to-no-think data ratios on **AIME24** under think and no-think modes for **Qwen2.5-7B-Instruct**. Ratios denote the proportion of think to no-think samples, with the total training size fixed at 80k. Increasing the proportion of no-think data effectively reduces no-think output length while largely preserving accuracy in both modes. *Note: The “1:1” ratio is an approximate proportion ($\approx 5:4$) due to the design for dataset construction.*

Think:No-think	Think		No-think	
	Acc.	Len.	Acc.	Len.
1:4	20.56	13229.06	4.33	1453.21
1:2	25.00	13233.21	3.00	1592.96
1:1	27.67	12799.13	5.33	2086.05
2:1	31.33	12524.66	3.33	2233.13

Table 19: Effect of different think-to-no-think data ratios on **MATH500** under think and no-think modes for **Llama3.1-8B-Instruct**. Ratios denote the proportion of think to no-think samples, with the total training size fixed at 80k. Increasing the proportion of no-think data effectively reduces no-think output length while largely preserving accuracy in both modes. *Note: The “1:1” ratio is an approximate proportion ($\approx 5:4$) due to the design for dataset construction.*

Think:No-think	Think		No-think	
	Acc.	Len.	Acc.	Len.
1:4	69.56	6184.32	50.40	802.80
1:2	74.32	5743.64	53.04	811.43
1:1	78.12	5263.71	55.22	899.72
2:1	77.96	5147.29	55.16	1025.80

Table 20: Effect of different think-to-no-think data ratios on **AIME24** under think and no-think modes for **Llama3.1-8B-Instruct**. Ratios denote the proportion of think to no-think samples, with the total training size fixed at 80k. Increasing the proportion of no-think data effectively reduces no-think output length while largely preserving accuracy in both modes. *Note: The “1:1” ratio is an approximate proportion ($\approx 5:4$) due to the design for dataset construction.*

Think:No-think	Think		No-think	
	Acc.	Len.	Acc.	Len.
1:4	5.33	14416.15	2.00	1234.35
1:2	13.33	13767.91	3.33	1774.01
1:1	20.33	13253.21	4.00	2108.03
2:1	19.33	13328.44	4.67	2229.22

Table 21: Effect of different think-to-no-think data ratios on **GPQA** under think and no-think modes for **Llama3.1-8B-Instruct**. Ratios denote the proportion of think to no-think samples, with the total training size fixed at 80k. Increasing the proportion of no-think data effectively reduces no-think output length while largely preserving accuracy in both modes. *Note: The “1:1” ratio is an approximate proportion ($\approx 5:4$) due to the design for dataset construction.*

Think:No-think	Think		No-think	
	Acc.	Len.	Acc.	Len.
1:4	32.73	10886.36	30.81	1470.06
1:2	33.23	10493.00	31.11	1039.15
1:1	37.58	10004.75	31.16	987.24
2:1	38.48	9885.08	31.92	1359.05

Table 22: Effect of different think-to-no-think data ratios on **MMLU-STEM** under think and no-think modes for **Llama3.1-8B-Instruct**. Ratios denote the proportion of think to no-think samples, with the total training size fixed at 80k. Increasing the proportion of no-think data effectively reduces no-think output length while largely preserving accuracy in both modes. *Note: The “1:1” ratio is an approximate proportion ($\approx 5:4$) due to the design for dataset construction.*

Think:No-think	Think		No-think	
	Acc.	Len.	Acc.	Len.
1:4	75.45	4474.87	50.02	568.52
1:2	77.74	4181.22	49.60	540.94
1:1	79.16	3774.97	48.84	546.10
2:1	79.80	3542.41	49.67	584.47

A.4 Results for 2-Phase and Mix Training

Comprehensive results for models trained via 2-phase and mix training strategies are provided in [Tables 23 and 24](#) for **Qwen2.5-7B-Instruct** and [Tables 25 to 28](#) for **Llama3.1-8B-Instruct**, which confirm the advantage of two-phase training in improving no-think controllability. Compared to mix training, the two-phase approach consistently produces shorter no-think outputs across benchmarks—such as **MATH500** (585.63 vs. 775.67 tokens at 140k scale)—without sacrificing think-mode accuracy. This confirms that sequential training on think and hybrid data effectively reduces reasoning leakage.

Table 23: Comparison of 2-phase and mix training strategies on **MATH500** under think and no-think modes for **Qwen2.5-7B-Instruct**. *Scale* denotes the number of samples (e.g., 20k = 20,000). We report accuracy (%) and average output length. The *2-phase* strategy first trains on pure-think data before fine-tuning with mixed data, while the *mix* strategy directly trains on mixed data. Across all scales, *2-phase* strategy yields shorter no-think outputs compared to *mix* training.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	2-phase	83.38	4797.40	47.26	870.50
	mix	83.18	4704.91	60.12	2214.08
40k	2-phase	85.18	4670.88	43.52	798.87
	mix	85.46	4589.04	62.10	1437.72
80k	2-phase	85.86	4626.42	51.86	639.60
	mix	85.88	4539.53	63.16	1086.00
140k	2-phase	86.78	4481.74	63.60	585.63
	mix	86.58	4442.49	63.90	775.67

Table 24: Comparison of 2-phase and mix training strategies on **AIME24** under think and no-think modes for **Qwen2.5-7B-Instruct**. *Scale* denotes the number of training samples (e.g., 20k = 20,000). We report accuracy (%) and average output length. The *2-phase* strategy first trains on pure-think data before fine-tuning with mixed data, while the *mix* strategy directly trains on mixed data. Across all scales, *2-phase* strategy yields shorter no-think outputs compared to *mix* training.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	2-phase	20.00	13620.71	4.33	1847.35
	mix	18.33	13397.91	7.00	5654.04
40k	2-phase	22.67	13562.19	1.67	1635.34
	mix	19.33	13586.36	7.00	2584.13
80k	2-phase	30.00	12962.41	3.00	1398.48
	mix	27.67	12799.13	5.33	2086.05
140k	2-phase	32.67	12271.01	5.00	853.82
	mix	36.00	12507.64	5.00	1293.08

Table 25: Comparison of 2-phase and mix training strategies on **MATH500** under think and no-think modes for **Llama3.1-8B-Instruct**. Here, *Scale* denotes the number of training samples (e.g., 20k means 20,000 samples). We report both accuracy (%) and average output length. The *2-phase* strategy first trains on pure-think data before fine-tuning with mixed think and no-think data, while the *mix* strategy directly trains on think and no-think mixed data. Notably, across all scales, *2-phase* strategy consistently yields shorter output lengths under the no-think mode compared to *mix* training.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	2-phase	68.56	6300.03	47.18	1180.40
	mix	69.10	6008.17	49.12	1111.11
40k	2-phase	72.52	6033.99	48.50	1170.26
	mix	73.40	5685.25	53.92	1118.49
80k	2-phase	76.90	5475.04	50.00	870.21
	mix	78.12	5263.71	55.22	899.72
140k	2-phase	80.66	5059.14	50.76	731.98
	mix	82.00	4896.76	57.72	780.87

Table 26: Comparison of 2-phase and mix training strategies on **AIME24** under think and no-think modes for **Llama3.1-8B-Instruct**. Here, *Scale* denotes the number of training samples (e.g., 20k means 20,000 samples). We report both accuracy (%) and average output length. The *2-phase* strategy first trains on pure-think data before fine-tuning with mixed think and no-think data, while the *mix* strategy directly trains on think and no-think mixed data. Notably, across all scales, *2-phase* strategy consistently yields shorter output lengths under the no-think mode compared to *mix* training.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	2-phase	7.00	13737.77	4.33	2735.59
	mix	8.33	14182.81	5.67	2949.06
40k	2-phase	9.67	14034.31	4.33	2692.68
	mix	10.00	14088.24	4.00	2995.47
80k	2-phase	16.00	13690.04	5.00	1776.30
	mix	20.33	13253.21	4.00	2108.03
140k	2-phase	20.33	12837.85	1.67	1429.10
	mix	23.67	13141.84	2.67	2229.50

Table 27: Comparison of 2-phase and mix training strategies on **GPQA** under think and no-think modes for **Llama3.1-8B-Instruct**. Here, *Scale* denotes the number of training samples (e.g., 20k means 20,000 samples). We report both accuracy (%) and average output length. The *2-phase* strategy first trains on pure-think data before fine-tuning with mixed think and no-think data, while the *mix* strategy directly trains on think and no-think mixed data. Notably, across all scales, *2-phase* strategy consistently yields shorter output lengths under the no-think mode compared to *mix* training.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	2-phase	36.31	10182.25	26.67	966.86
	mix	36.41	9119.54	32.53	2814.42
40k	2-phase	35.15	10626.91	29.60	975.13
	mix	36.52	10144.51	30.35	1957.28
80k	2-phase	37.22	10100.31	27.98	705.01
	mix	37.58	10004.75	31.16	987.24
140k	2-phase	35.10	10077.06	28.48	745.44
	mix	34.95	10190.07	31.67	3352.59

Table 28: Comparison of 2-phase and mix training strategies on **MMLU-STEM** under think and no-think modes for **Llama3.1-8B-Instruct**. Here, *Scale* denotes the number of training samples (e.g., 20k means 20,000 samples). We report both accuracy (%) and average output length. The *2-phase* strategy first trains on pure-think data before fine-tuning with mixed think and no-think data, while the *mix* strategy directly trains on think and no-think mixed data. Notably, across all scales, *2-phase* strategy consistently yields shorter output lengths under the no-think mode compared to *mix* training.

Scale	Setting	Think		No-think	
		Acc.	Len.	Acc.	Len.
20k	2-phase	77.26	4056.91	53.38	572.71
	mix	76.82	3632.27	59.88	1127.23
40k	2-phase	79.10	3973.26	53.98	592.56
	mix	78.50	3866.03	55.22	765.63
80k	2-phase	79.10	3980.86	46.02	448.93
	mix	79.16	3774.97	48.84	546.10
140k	2-phase	78.50	3822.84	48.40	461.48
	mix	78.43	3782.77	52.52	849.36

B Additional Results on Training Strategies

Tables 29 to 31 report results of two-phase and mix training across scales on Math500, AIME24, and GPQA-Diamond. Two-phase consistently reduces no-think output length while preserving think-mode accuracy, showing its advantage in suppressing unnecessary reasoning traces without harming task performance.

Table 29: Performance of **Qwen2.5-7B-Instruct** under different training settings on **MATH500**. We report accuracy (Acc.) and output length (Len.) for both think and no-think modes.

Setting	Think		No-think	
	Acc.	Len.	Acc.	Len.
Baseline	–	–	74.96	608.85
Demo (w/o pairs)	37.08	650.35	31.68	390.63
20k-2phase	83.38	4797.40	47.26	870.50
20k-mix	83.18	4704.91	60.12	2214.08
40k-2phase	85.18	4670.88	43.52	798.87
40k-mix	85.46	4589.04	62.10	1437.72

Table 30: Performance of **Qwen2.5-7B-Instruct** under different training settings on **AIME24**. We report accuracy (Acc.) and output length (Len.) for both think mode and no-think mode.

Setting	Think		No-think	
	Acc.	Len.	Acc.	Len.
20k-2phase	20.00	13620.71	4.33	1847.35
20k-mix	18.33	13397.91	7.00	5654.04
40k-2phase	22.67	13562.19	1.67	1635.34
40k-mix	19.33	13586.36	7.00	2584.13

Table 31: Performance of **Qwen2.5-7B-Instruct** under different training settings on **GPQA-Diamond**. We report accuracy (Acc.) and output length (Len.) for both think mode and no-think mode.

Setting	Think		No-think	
	Acc.	Len.	Acc.	Len.
20k-2phase	41.16	9872.89	26.72	1404.46
20k-mix	40.00	9444.03	39.50	5503.32
40k-2phase	41.62	9903.88	32.12	2892.41
40k-mix	40.61	9571.43	38.99	5330.81

C System Prompt Ablation

Table 32 compares Qwen 2-phase training with and without system prompts on Math500, AIME24, and GPQA-Diamond. Results show that while system prompts slightly reduce output length, they also lead to degraded accuracy in no-think mode.

Table 32: Comparison of **Qwen** 2-phase training with and without system prompt on **MATH500**, **AIME24**, and **GPQA-Diamond**.

Model	Think		No-think	
	Acc.	Len.	Acc.	Len.
Math500				
w/o Sys. Prompt	86.1	4539	77.4	2971
w/ Sys. Prompt	84.6	4383	76.6	2840
AIME24				
w/o Sys. Prompt	29.3	12615	23.3	5744
w/ Sys. Prompt	28.7	11799	16.0	6269
GPQA-Diamond				
w/o Sys. Prompt	40.6	10283	35.9	8378
w/ Sys. Prompt	39.0	9931	37.6	7784

D Generalization Beyond Mathematical Reasoning

This section reports the full results supporting the generalization evaluation in Section 4.2: accuracy, output length, and reflective-token count on the full MMLU benchmark (57 subjects) and TruthfulQA (Tables 33 to 38). Throughout this section, “Instruct” and “Reasoning” denote the pure-instruct and pure-reasoning-tuned baselines for each model family, and “Our Recipe” denotes our proposed hybrid-thinking training recipe in both modes.

Table 33: Generalization beyond math for **Qwen** on **full MMLU** (57 subjects; not the MMLU-STEM subset used elsewhere in this paper): accuracy, output length, and reflective-token count.

Setting	Acc.	Len.	#Refl.
Instruct	46.17%	267.60	13
Reasoning	75.24%	3723.06	516808
Our Recipe (think)	76.93%	2822.63	274499
Our Recipe (no-think)	57.06%	1547.63	9073

Table 34: Generalization beyond math for **Qwen** on **TruthfulQA**: accuracy, output length, and reflective-token count.

Setting	Acc.	Len.	#Refl.
Instruct	69.0%	17.2	0
Reasoning	42.8%	22283.3	32220
Our Recipe (think)	61.5%	6474.3	437
Our Recipe (no-think)	56.5%	735.6	80

Table 35: Generalization beyond math for **Llama** on **full MMLU** (57 subjects; not the MMLU-STEM subset used elsewhere in this paper): accuracy, output length, and reflective-token count.

Setting	Acc.	Len.	#Refl.
Instruct	64.32%	578.48	13
Reasoning	61.32%	1869.46	242607
Our Recipe (think)	70.89%	3748.36	491453
Our Recipe (no-think)	52.12%	531.77	14858

Table 36: Generalization beyond math for **Llama** on **TruthfulQA**: accuracy, output length, and reflective-token count.

Setting	Acc.	Len.	#Refl.
Instruct	49.5%	9.2	0
Reasoning	54.2%	6210.8	8276
Our Recipe (think)	58.8%	3564.1	3916
Our Recipe (no-think)	41.0%	1256.9	0

Table 37: Generalization beyond math for **Phi** on **full MMLU** (57 subjects; not the MMLU-STEM subset used elsewhere in this paper): accuracy, output length, and reflective-token count.

Setting	Acc.	Len.	#Refl.
Instruct	70.47%	933.33	21
Reasoning	76.05%	7485.83	32478
Our Recipe (think)	87.44%	1881.09	158644
Our Recipe (no-think)	65.36%	365.22	2906

Table 38: Generalization beyond math for **Phi** on **TruthfulQA**: accuracy, output length, and reflective-token count.

Setting	Acc.	Len.	#Refl.
Instruct	75.0%	25.5	0
Reasoning	74.8%	257.6	0
Our Recipe (think)	70.5%	4335.6	6447
Our Recipe (no-think)	67.0%	159.0	0

E Stability and Inference Efficiency

This section reports the full results supporting the stability and efficiency evaluation in [Section 4.2](#): mean and standard deviation over 10 random seeds for output length, reflective-token count, wall-clock inference time, and throughput. Throughout this section, “Instruct” and “Reasoning” denote the pure-instruct and pure-reasoning-tuned baselines for each model family, and “Our Recipe” denotes our proposed hybrid-thinking training recipe in both modes.

Table 39: Statistical rigor for **Qwen** on **MATH500**: mean $\pm$ std over 10 independent random seeds for reflective-token count (#Refl.) and total output length (Len.).

Setting	#Refl.	Len.
Instruct	0.00 $\pm$ 0.00	329.00 $\pm$ 105.86
Reasoning	1346.00 $\pm$ 282.37	4251.20 $\pm$ 420.17
Our Recipe (think)	1335.50 $\pm$ 130.81	4396.16 $\pm$ 330.28
Our Recipe (no-think)	0.80 $\pm$ 1.75	529.41 $\pm$ 119.65

Table 40: Statistical rigor for **Qwen** on **AIME24**: mean $\pm$ std over 10 independent random seeds for reflective-token count (#Refl.) and total output length (Len.).

Setting	#Refl.	Len.
Instruct	0.00 $\pm$ 0.00	927.37 $\pm$ 1792.48
Reasoning	255.30 $\pm$ 56.50	13013.27 $\pm$ 2491.25
Our Recipe (think)	223.90 $\pm$ 26.91	11962.17 $\pm$ 2716.82
Our Recipe (no-think)	0.00 $\pm$ 0.00	1222.47 $\pm$ 1630.07

F Broader Impact and Future Applications

Table 41: Statistical rigor for **Llama** on **MATH500**: mean $\pm$ std over 10 independent random seeds for reflective-token count (#Refl.) and total output length (Len.).

Setting	#Refl.	Len.
Instruct	0.00 $\pm$ 0.00	1554.07 $\pm$ 880.38
Reasoning	1971.80 $\pm$ 246.43	4231.61 $\pm$ 424.62
Our Recipe (think)	2578.50 $\pm$ 853.49	5580.90 $\pm$ 452.84
Our Recipe (no-think)	0.90 $\pm$ 1.91	728.17 $\pm$ 233.68

Table 42: Statistical rigor for **Llama** on **AIME24**: mean $\pm$ std over 10 independent random seeds for reflective-token count (#Refl.) and total output length (Len.).

Setting	#Refl.	Len.
Instruct	0.00 $\pm$ 0.00	4635.40 $\pm$ 4255.10
Reasoning	352.80 $\pm$ 125.31	11714.47 $\pm$ 3547.74
Our Recipe (think)	438.30 $\pm$ 436.76	14688.33 $\pm$ 1594.26
Our Recipe (no-think)	0.40 $\pm$ 1.26	1105.03 $\pm$ 600.98

Table 43: Statistical rigor for **Phi** on **MATH500**: mean $\pm$ std over 10 independent random seeds for reflective-token count (#Refl.) and total output length (Len.).

Setting	#Refl.	Len.
Instruct	0.00 $\pm$ 0.00	656.06 $\pm$ 58.05
Reasoning	127.90 $\pm$ 36.85	3010.61 $\pm$ 1336.74
Our Recipe (think)	1072.70 $\pm$ 121.07	3493.02 $\pm$ 334.76
Our Recipe (no-think)	0.00 $\pm$ 0.00	654.51 $\pm$ 243.35

Table 44: Statistical rigor for **Phi** on **AIME24**: mean $\pm$ std over 10 independent random seeds for reflective-token count (#Refl.) and total output length (Len.).

Setting	#Refl.	Len.
Instruct	0.00 $\pm$ 0.00	1229.37 $\pm$ 429.75
Reasoning	28.30 $\pm$ 14.07	6825.90 $\pm$ 2727.26
Our Recipe (think)	186.50 $\pm$ 53.14	10215.10 $\pm$ 1813.10
Our Recipe (no-think)	0.00 $\pm$ 0.00	645.17 $\pm$ 190.61

Table 45: Statistical rigor for **Qwen** on **MATH500**: mean $\pm$ std over 10 independent random seeds for wall-clock inference time (s/instance) and throughput (tokens/s). Latency profiling was conducted on MATH500 only.

Setting	Time (s)	Tok/s
Instruct	14.30 $\pm$ 34.67	314.66 $\pm$ 73.98
Reasoning	116.60 $\pm$ 3.68	333.54 $\pm$ 55.39
Our Recipe (think)	111.25 $\pm$ 14.24	321.88 $\pm$ 51.86
Our Recipe (no-think)	16.86 $\pm$ 33.80	330.18 $\pm$ 67.98

Table 46: Statistical rigor for **Llama** on **MATH500**: mean $\pm$ std over 10 independent random seeds for wall-clock inference time (s/instance) and throughput (tokens/s). Latency profiling was conducted on MATH500 only.

Setting	Time (s)	Tok/s
Instruct	70.27 $\pm$ 61.74	238.27 $\pm$ 88.20
Reasoning	114.16 $\pm$ 24.28	306.27 $\pm$ 55.07
Our Recipe (think)	130.46 $\pm$ 4.39	339.03 $\pm$ 46.82
Our Recipe (no-think)	11.86 $\pm$ 7.88	297.78 $\pm$ 49.75

Agents. In persistent agent workflows, most steps are routine (parsing a tool output, filling a tem-

Table 47: Statistical rigor for **Phi** on **MATH500**: mean $\pm$ std over 10 independent random seeds for wall-clock inference time (s/instance) and throughput (tokens/s). Latency profiling was conducted on MATH500 only.

Setting	Time (s)	Tok/s
Instruct	20.54 $\pm$ 9.87	187.35 $\pm$ 21.99
Reasoning	143.61 $\pm$ 66.79	153.76 $\pm$ 36.66
Our Recipe (think)	190.49 $\pm$ 16.70	160.36 $\pm$ 21.07
Our Recipe (no-think)	9.89 $\pm$ 2.39	195.37 $\pm$ 23.20

plate) rather than genuinely deliberative, and unnecessary reasoning compounds cost over long horizons rather than staying local (Cuadron et al., 2025); efficient serving of such long reasoning-augmented programs is itself an active target (Fu et al., 2025). A reliable no-think mode would let a system spend tokens only where deliberation is needed, which matters for persistent multi-agent workspaces where many calls are routine bookkeeping (Wang, 2026), for skill libraries that could declare a reasoning mode at call time (Yang et al., 2026b), for evaluation over long, controllable-difficulty horizons (Yang et al., 2026a), and for keeping long-term memory records compact rather than padded with abandoned reasoning (Long et al., 2026).

Inference Serving and Efficiency. On the serving side, most inference-time efficiency work targets the cost of producing each token – through shorter reasoning traces (Xu et al., 2025; Lee et al., 2025), step-compressed reasoning (Zhang et al., 2025a), or broader surveys of efficient reasoning (Sui et al., 2025) – or through serving infrastructure such as paged attention (Kwon et al., 2023). A reliable no-think mode instead reduces how many tokens are generated in the first place, and composes with these orthogonal axes: unbiased KV-cache compression via Fourier-domain windowing (Li et al., 2025b), quantization strategies applied selectively to different parts of the KV cache (Hariri et al., 2026c), lossless weight compression via entropy coding (Zhang et al., 2025b), and efficient architecture and inference techniques explored for specialized language models in other domains such as protein sequence models (Wang et al., 2026a).

Model Architecture. A separate line of work intervenes at the level of reasoning strategy or context handling rather than the token stream: surveys of reasoning strategies in LLMs (Zhang et al.,

2026; Bandyopadhyay et al., 2025) and inference-time reasoning guidance (Sui et al., 2026) motivate architecture-level control as a complementary axis. Training-free triggers can elicit an intermediate reasoning budget between full chain-of-thought and a direct answer (Yang et al., 2026c), context windows can be extended without additional tuning (Jin et al., 2024), and mode separation itself can be enforced architecturally, with a fixed per-sequence routing decision rather than a learned router (Wang et al., 2026b).

Post-Training and Evaluation. The no-think behavior studied here is a property of post-training rather than architecture alone (Rafailov et al., 2023). Post-training effects can interact across domains in order-sensitive, asymmetric ways, directly relevant to reasoning-focused reinforcement learning (Yang et al., 2026d; Yu et al., 2025; Shao et al., 2024): mode separation should be re-measured whenever the training mixture changes rather than assumed to transfer. Inference-time scaling raises its own evaluation questions, from regime and reproducibility standards (Hariri et al., 2026a; Muennighoff et al., 2025) to ranking reasoning models under test-time compute (Hariri et al., 2026b), and to guiding a small model with a larger one at inference time rather than retraining (Yang et al., 2025b; Leviathan et al., 2023).