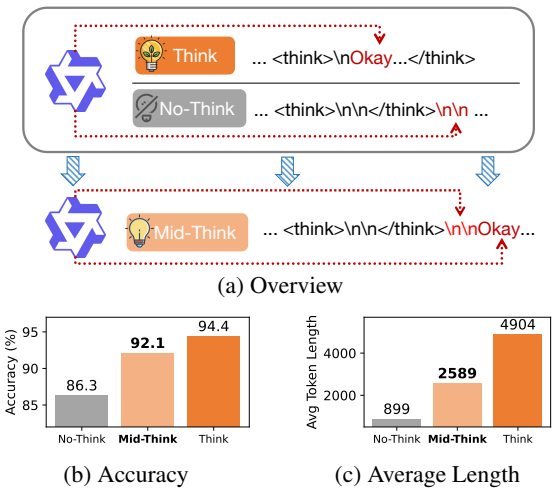
Mid-Think: Training-Free Intermediate-Budget Reasoning via Token-Level Triggers

Wang Yang[1*], **Shouren Wang**[1*], **Debargha Ganguly**[1], **Xinpeng Li**[1]
Chaoda Song[1], **Vikash Singh**[1], **Vipin Chaudhary**[1], **Xiaotian Han**[1]
[1]Case Western Reserve University
[*]Equal contribution
{wxy320,dxg512,sxw992,vxs465,vipin,xhan}@case.edu

Abstract

Hybrid reasoning language models are commonly controlled through high-level Think/No-think instructions to regulate reasoning behavior, yet we found that such mode switching is largely driven by a small set of trigger tokens rather than the instructions themselves. Through attention analysis and controlled prompting experiments, we show that a leading "Okay" token induces reasoning behavior, while the newline pattern following "</think>" suppresses it. Based on this observation, we propose *Mid-Think*, a simple training-free prompting format that combines these triggers to achieve intermediate-budget reasoning, consistently outperforming fixed-token and prompt-based baselines in terms of the accuracy-length trade-off. Furthermore, applying Mid-Think to RL training after SFT reduces training time by approximately 15% while improving final performance of Qwen3-8B on AIME from 69.8% to 72.4% and on GPQA from 58.5% to 61.1%, demonstrating its effectiveness for both inference-time control and RL-based reasoning training. Our code is available at https://github.com/uservan/Mid-Think.



(a) Overview

(b) Accuracy (c) Average Length

Figure 1: Illustration of Mid-Think and its performance on MATH500. (a) Overview of Mid-Think comparing with Think and No-Think. In the Think mode, subsequent tokens primarily attend to the cue token "Okay", while in the No-think mode, generated tokens focus on the newline following the </think> marker (i.e., the \n\n token). Mid-Think combines both cues into a unified prompting format to induce intermediate reasoning behavior. (b) Accuracy and (c) average output length under No-think, Think, and Mid-Think settings. Mid-Think achieves intermediate-budget reasoning without additional training, retaining most accuracy gains while significantly reducing generation length.

1 Introduction

Large language models exhibit a variety of emergent phenomena (Sun et al., 2024; Robinson et al., 2024), many of which have been actively exploited to improve model. The work of Attention Sink (Xiao et al., 2023; Gu et al., 2024) has been leveraged to accelerate long-context inference. Explicit </think> tags are used to expose intermediate reasoning and enable hybrid thinking behaviors (Fang et al., 2025; Yang et al., 2025a). Token-level cues such as *"wait"* and *"alternatively"* have been analyzed to regulate reasoning probability and entropy, forming the methods like No-Wait (Wang et al., 2025a) and SpecExit (Yang et al., 2025b). Speculative Thinking (Yang et al., 2025f) exploits structural patterns such as \n\n to coordinate cooperation between large and small reasoning models.

As shown in Figure 1, we find that **the Think and No-think behaviors of reasoning models are largely governed by a small number of trigger tokens**. Using Qwen3-8B as a representative example, we compute the average attention from generated tokens to each prompt token. The analysis reveals that only a few tokens consistently attracts substantially higher attention than other prompt components. Specifically, in the Think mode, reasoning behavior is dominated by the "Okay" token immediately following the <think> tag, whereas in the No-think mode, the model primarily attends to

Findings of the Association for Computational Linguistics: ACL 2026, pages 6024–6038

the newline pattern after the </think> tag.

To further verify this phenomenon, we design multiple prompting formats and evaluate them on MATH500, AIME, and GPQA (Rein et al., 2024). We observe that prompts containing a leading "Okay" token consistently induce reasoning behavior, achieving accuracy and wait count comparable to the standard Think mode. In contrast, </think>+\n\n pattern significantly reduces accuracy and wait count, yielding behavior closer to No-think regime. Together, it demonstrate that reasoning and non-reasoning behaviors are not determined by high-level instructions, but are instead dominated by a small set of token-level triggers.

Motivated by this observation, **we propose a new format, termed Mid-Think (<think>\n\n</think>\n\n<reason>\nOkay...)**, which enables intermediate-budget reasoning without any additional training. By explicitly integrating both reasoning-activating and reasoning-suppressing cues, Mid-Think induces a balanced reasoning behavior between the standard Think and No-Think modes. The comparison results of these modes are shown in Figure 1.

Empirically, we find that Mid-Think consistently achieves performance comparable to, and in some cases better than, fixed intermediate budgets, effectively lying on or beyond the Pareto frontier between accuracy and output length, Mid-Think outperforms existing training-free approaches proposed in Qwen3, including fixed-token budgets and prompt-based budget control, demonstrating a more reliable and fine-grained mechanism for reasoning budget regulation.

Finally, we apply Mid-Think to RL training after SFT and observe consistent improvements in both efficiency and performance. Compared to standard Think-mode training, Mid-Think significantly reduces RL training time while achieving better post-training results. For Qwen3-8B, a model trained with Mid-Think attains higher accuracy when evaluated in the Think mode on AIME (72.4% vs. 69.8%), while reducing training time from 54 hours to 46 hours, corresponding to an efficiency gain of approximately 15%.

2 Motivation: Reasoning Is Governed by a Few Tokens

This section presents an empirical observation of an *"overfitting"* phenomenon in the reasoning processes. We first analyze the homogenization of patterns in reasoning modes and datasets. Then from an attention-based perspective, we observe that models tend to focus on a small set of specific tokens: the reasoning mode is largely anchored to the lexical cue "Okay", while the non-reasoning mode is primarily associated with "\n\n" following the </think> tag. Finally, we design controlled experiments with different reasoning formats to systematically validate this phenomenon.

Token	DeepSeek	Qwen3	OpenR1-Math
<think>	✓	✓	✓
</think>	✓	✓	✓
Okay	✓	✓	✓

Table 1: Occurrence of explicit reasoning-related tokens in the outputs of reasoning models and training datasets. OpenR1-Math means the reasoning training dataset of open-r1/OpenR1-Math-220k

2.1 "Overfitting" in Reasoning Initiation

Mainstream reasoning models, such as DeepSeek (Liu et al., 2024; Guo et al., 2025) and Qwen3 (Yang et al., 2025a) thinking models, typically enclose their reasoning processes using explicit <think> tags. During the reasoning phase, these models often follow a fixed lexical pattern to initiate their thoughts, commonly starting with tokens such as "Okay" or "Alright".

Beyond model outputs, a substantial portion of existing fine-tuning data is derived from generations produced by these reasoning models themselves. As a result, such datasets are heavily populated with these unified reasoning patterns and are subsequently used to fine-tune new models. Representative examples include OpenR1-Math-220k (Hugging Face, 2025), OpenMathReasoning (Moshkov et al., 2025) and so on, which is summarized in Table 1.

2.2 Attention-Based Evidence of Token-Level Triggers

A natural question is whether such unified opening patterns may lead to "overfitting", where a small set of tokens dominates the model reasoning behavior and effectively serves as a switch for different reasoning modes.

To investigate this question, we take Qwen3-8B as a case study and examine five different generation settings: (1) No-Think mode, (2) standard Think mode, (3) a generation starting with the pattern </think> followed by "Okay", (4) a think-style generation without explicit <think> tags and (5) a

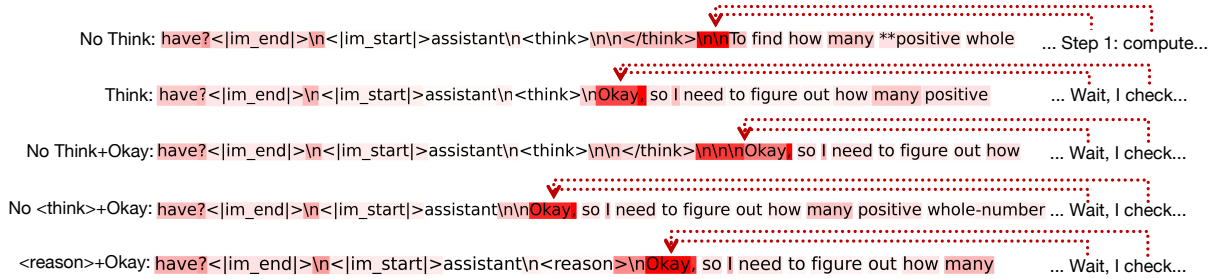


Figure 2: Average attention from generated tokens to different opening tokens under five generation modes. Darker red indicates higher attention received from subsequent tokens. When "Okay" appears at the beginning, the model produces an explicit reasoning process ("wait", "alternatively", etc), and attention is primarily concentrated on "Okay". In the No-Think mode, the \n\n following </think> absorbs most of the attention mass.

Mode	Format	MATH500		AIME		GPQA	
		Acc (%)	Wait	Acc (%)	Wait	Acc (%)	Wait
No-think	<think>\n\n</think>\n\n	83.2	2082	21.3	447	37.5	459
Think	<think>\nOkay	94.6	81367	71.6	40342	60.9	71817
No Think + Okay	<think>\n\n</think>\n\nOkay	92.3	33724	55.3	19360	47.3	56177
No <think> + Okay	\n\nOkay	94.1	80272	72.7	40407	59.3	71914
<reason> + Okay	<reason>\nOkay	94.0	81075	72.7	38067	56.6	66105

Table 2: Formats and performance statistics under five different generation modes of Qwen3-8B. The Format column specifies the exact prompting patterns used to trigger different behaviors. We report accuracy on MATH500, AIME, and GPQA, together with the corresponding wait counts. When the prompt begins with "Okay", both accuracy and wait statistics closely match those of the Think mode.

generation starting with the pattern <reason> followed by "Okay". For each setting, we let the model generate tokens and compute the average attention from subsequent tokens to the opening tokens. Specifically, we average attention weights across all layers and attention heads.

The results are shown in Figure 2. We find that in the 4 settings: Think mode, No <think>+"Okay", </think>+"Okay" and <reason>+"Okay", the token "Okay" consistently receives the highest attention from later tokens. This suggests that the model's reasoning behavior is driven primarily by the lexical cue "Okay". By contrast, in the No-Think mode, attention is predominantly concentrated on the \n\n followed by </think> token.

Together, these observations indicate that the *model uses "Okay" as a trigger to enter the reasoning mode, while relying on the \n\n followed by </think> token as the primary signal to activate the No-Think behavior*. To further quantify this phenomenon, we compute the trigger-token attention mass for each reasoning mode and report detailed results in Appendix A.1.

2.3 Experimental Verification of Token-Level Triggers

To validate the hypothesized "overfitting" of reasoning models to token-level cues, we construct compositional prompting modes combining the triggers discussed above: No-think, Think, No-think + Okay, No <think> + Okay, and <reason> + Okay (see Table 2 for exact formats).

We evaluate Qwen3-8B on MATH500, AIME22–24, and GPQA, reporting accuracy and wait count in Table 2. Modes containing "Okay" as the opening cue (No <think> + Okay, <reason> + Okay) consistently match Think-mode accuracy and wait count, confirming its role as a reasoning trigger. The No-think + Okay setting, however, shows degraded performance, as the co-occurrence of \n\n</think> and "Okay" induces conflicting signals, yielding an intermediate rather than a clean reasoning regime.

3 Mid-Think: Intermediate-Budget Reasoning without Training

3.1 Mid-Think: Implementation

Building on the observations from the previous section, we find that in the Think mode, generated tokens primarily attend to the "Okay" cue following the <think> tag, whereas in the No-think mode, attention concentrates on the newline pattern following the </think> tag. Leveraging this insight, we combine these two cues to induce an intermediate reasoning regime without training, enabling the

Mode	Format (literal \n denotes newlines)
Think	<|im_start|>user{query}<|im_end|><|im_start|>assistant<think>{thinking_content}</think>{response}<|im_end|>
No-Think	<|im_start|>user{query}<|im_end|><|im_start|>assistant<think>\n\n</think>{response}<|im_end|>
Mid-Think	<|im_start|>user{query}<|im_end|><|im_start|>assistant<think>\n\n</think>\n\n<reason>\nOkay...{thinking_content}</reason>{response}<|im_end|>

Table 3: Prompt formats under three generation modes. Mid-Think combines key tokens from Think and No-Think (e.g., "Okay" and </think>\n\n) and wraps the thinking content with a special token (e.g., <reason>). The specific token choice is not essential; we evaluate three variants: <reason>, <begin>, and <less think>.

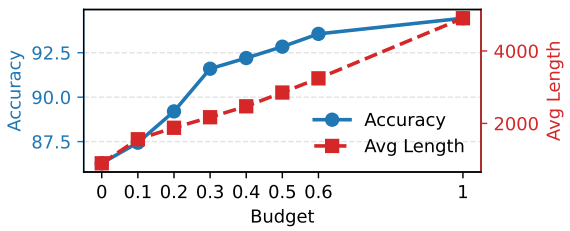


Figure 3: Budget-controlled reasoning on MATH500 using Qwen3-14B. The figure reports the average length and accuracy under different reasoning budgets. Both metrics increase steadily as the budget grows, validating the effectiveness of the budget-control mechanism.

Figure 4: Overview of the budget-controlled method. The model first generates a full response. The reasoning (think) content is then truncated to the specified budget (in tokens) and concatenated with the remaining prompt, after which the model generates the final response.

model to reason with a reduced token budget.

As shown in Table 3, we introduce a new prompting format, termed Mid-Think. Concretely, the format is: <think>\n\n</think>\n\n <reason>\nOkay.... This design exploits the joint presence of the </think> newline cue and the trigger "Okay", allowing the model to be simultaneously influenced by reasoning and non-reasoning signals.

The <reason> tag is introduced to explicitly delimit the reasoning content, serving a structural role analogous to that of the <think> tag. Notably, the choice of <reason> is not essential; alternative tokens can be used to achieve the same effect. In our experiments, we consider three variants: <reason>, <begin>, and <less think>.

3.2 Mid-Think Achieves Pareto-Optimal Performance

To validate the effectiveness of Mid-Think, we first introduce a budget-controlled evaluation protocol that measures a reasoning model's performance under different budgets in the standard Think mode, enabling us to construct a budget–performance curve. We then place Mid-Think on this curve and observe that it achieves intermediate-budget reasoning behavior and, in some cases, attains Pareto-optimal performance relative to similar budgets.

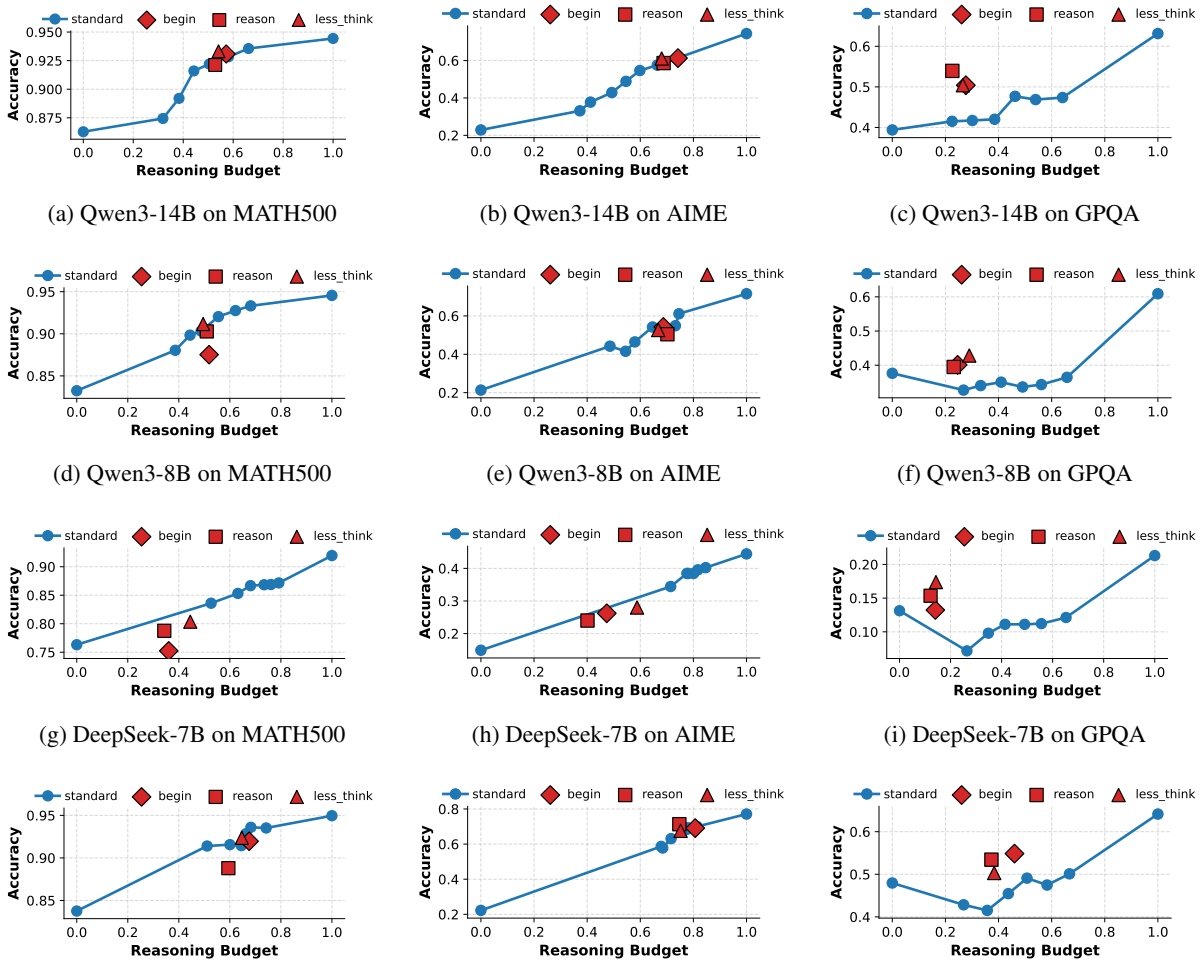
3.2.1 Budget-Controlled Reasoning Baseline

Implementation. To obtain a model's reasoning capability under different budgets, we first let the reasoning model generate a complete reasoning trajectory in the standard Think mode. Suppose the full reasoning process contains n tokens. We then construct budget-controlled variants by retaining only a fraction of the reasoning tokens according to a predefined budget ratio.

Concretely, under a budget of 0.9, we keep the first $0.9 \times n$ reasoning tokens, whereas under a budget of 0.1, only the first $0.1 \times n$ tokens are preserved, as illustrated in Figure 4. This procedure allows us to systematically control the effective reasoning budget and obtain reasoning performance across different budget levels.

Verification. To verify the effectiveness, we conduct experiments using Qwen3-14B on MATH500. We evaluate the model under multiple budget settings, including $0.1, 0.2, 0.3, 0.4, 0.5, 0.6$, as well as the No-Think setting (corresponding to a budget of 0.0) and the standard Think setting (corresponding to a budget of 1.0).

As shown in Figure 3, both the average generation length and accuracy increase monotonically with the reasoning budget. This behavior is consistent with the intended effect of the budget-control mechanism, confirming that the proposed method provides a reliable and interpretable way to modulate the model's effective reasoning capacity.



(a) Qwen3-14B on MATH500 (b) Qwen3-14B on AIME (c) Qwen3-14B on GPQA

(d) Qwen3-8B on MATH500 (e) Qwen3-8B on AIME (f) Qwen3-8B on GPQA

(g) DeepSeek-7B on MATH500 (h) DeepSeek-7B on AIME (i) DeepSeek-7B on GPQA

(j) Qwen3-32B on MATH500 (k) Qwen3-32B on AIME (l) Qwen3-32B on GPQA

Figure 5: Comparison between models under different reasoning budgets and Mid-Think across multiple datasets and model types, including hybrid-thinking (Qwen3-8B, Qwen3-14B, Qwen3-32B) and pure-think (DeepSeek-R1-Distill-Qwen-7B) models. Mid-Think is evaluated with different tags (`<reason>`, `<begin>`, `<less think>`). Across all models and datasets (MATH500, AIME, and GPQA), Mid-Think consistently achieves performance corresponding to intermediate reasoning budgets, and on GPQA it even surpasses fixed-budget baselines, yielding Pareto-optimal accuracy–efficiency trade-offs between Think and No-Think mode.

3.2.2 Mid-Think v.s. Different Budget

Experimental Setup. We evaluate a diverse set of models, including hybrid reasoning models (Qwen3-8B and Qwen3-14B), a pure supervised model (DeepSeek-Qwen-7B), and a purely RL-trained model (Qwen3-32B). For each model, we first measure performance under different fixed budgets ranging from 0.1 to 0.6, as well as the standard Think and No-think settings, on MATH500, AIME22–24, and GPQA. We then evaluate the same models under the proposed Mid-Think mode and visualize all results jointly for comparison.

Hybrid-Think Model Results. Figure 5 presents the results of Qwen3-8B and Qwen3-14B on MATH500, AIME22–24, and GPQA. Across all three benchmarks, the performance of the proposed Mid-Think mode consistently aligns with that of an intermediate budget, approximately corresponding to a budget of 0.5 in terms of accuracy.

Notably, on GPQA, Mid-Think achieves the strongest performance, surpassing neighboring budget settings and even exceeding the Pareto frontier defined by the budget–performance trade-off. These results indicate that reasoning models with Mid-Think mode can obtain intermediate-budget reasoning capability in a training-free manner.

Similar trends are observed for Qwen3-14B, demonstrating that the effectiveness of Mid-Think generalizes across different model scales.

Pure-Think Model Results. To examine the effectiveness of Mid-Think on pure reasoning models, we additionally evaluate DeepSeek-Qwen-7B,

Model	Acc on MATH-500			Avg Length		
	No-T	Mid-T	Think	No-T	Mid-T	Think
Phi-4-mini-reasoning	75.6	**89.4**	90.6	1,047	3,333	4,031
DeepSeek-R1-Distill-Llama-8B	60.8	**72.6**	85.6	1,045	2,062	4,345
DeepSeek-R1-Distill-Qwen-2.5-7B	68.2	**74.6**	86.8	718	1,655	3,957
Qwen3-30B-A3B (MoE)	87.2	**92.0**	94.0	898	3,007	5,099

Table 4: Accuracy and average generation length on MATH-500 across different model families and architectures. Mid-Think (Mid-T) consistently improves over No-Think (No-T) while using fewer tokens than Think mode.

Model	Acc on LiveCodeBench			Avg Length		
	No-T	Mid-T	Think	No-T	Mid-T	Think
Qwen3-8B	40.8	**49.7**	65.9	760	2,212	9,931
Qwen3-14B	46.8	**62.7**	72.3	509	2,147	8,959

Table 5: Accuracy and average generation length on LiveCodeBench. Mid-Think (Mid-T) generalizes beyond math reasoning tasks, consistently outperforming No-Think (No-T) with significantly shorter outputs than Think mode.

with results shown in Figure 5. Under Mid-Think setting, the model's accuracy tends to align with smaller-budget regimes. We attribute this behavior to the model placing stronger emphasis on the </think>\n\n pattern, which biases the effective reasoning budget toward shorter reasoning spans.

Nevertheless, on GPQA, Mid-Think still achieves improved performance, surpassing the Pareto frontier defined by near-budget. This result suggests that even for pure thinking models, Mid-Think can yield favorable trade-offs between reasoning budget and performance.

RL-Model Results. To assess the effectiveness on RL-trained models, we evaluate Qwen3-32B on MATH500, AIME22–24, and GPQA, with results shown in Figure 5. We observe that RL-trained model remains compatible with the Mid-Think mode, exhibiting performance that closely matches an intermediate budget of approximately 0.5.

Moreover, on GPQA, Mid-Think again surpasses the Pareto frontier defined by near-budget. These findings indicate that the proposed Mid-Think strategy generalizes to RL-trained models and enables favorable budget–performance trade-offs even at larger model scales.

3.3 Generalization Across Model Families and Architectures

Across Model Families and Architectures. To assess whether Mid-Think generalizes beyond the Qwen family, we evaluate on four additional reasoning models spanning different families and architectures: Phi-4-mini-reasoning (Phi, dense), DeepSeek-R1-Distill-Llama-8B (Llama, dense), DeepSeek-R1-Distill-Qwen-2.5-7B (Qwen, dense), and Qwen3-30B-A3B (Qwen, MoE). All models were trained on reasoning traces beginning with "`Okay`". As shown in Table 4, Mid-Think consistently outperforms No-Think across all models and architectures while remaining substantially shorter than Think mode, demonstrating that the method is model-family and architecture agnostic.

Beyond Math: Coding Tasks. To assess whether Mid-Think generalizes beyond mathematical reasoning, we evaluate on LiveCodeBench(Jain et al., 2024) using Qwen3-8B and Qwen3-14B. As shown in Table 5, Mid-Think maintains its effectiveness on this coding benchmark: it substantially outperforms No-Think in accuracy while generating far fewer tokens than full Think mode. This demonstrates that the token-level trigger mechanism is not specific to mathematical reasoning but generalizes across task domains.

3.4 Trigger Origin and Robustness

A natural follow-up question is whether the observed trigger dependence is an emergent property of model *architecture*, or a consequence of *training data templates*. Our experiments support the latter. Across Qwen, Llama, and Phi model families—all of which were trained on reasoning traces beginning with "`Okay`". To further isolate the effect of training templates, we test `Alibaba-Apsara/DASD-4B-Thinking`, a model whose reasoning traces begin with "`We need`" rather than "`Okay`". Attention analysis on DASD-4B confirms that the dominant trigger token shifts from "`Okay`" to "`We`", suggesting the trigger adapts

Mode	Trigger	Acc (%)	Avg Len
No-Think	—	82.4	2,219
Mid-Think	`We`	88.8	2,796
Think	—	90.8	3,516

Table 6: Mid-Think with a template-adapted trigger ("`We`") on DASD-4B-Thinking (MATH-500). The method remains effective when the trigger is matched to the model's training template.

Mode	Acc (%)	Avg Len
No-Think	83.2	1,013
Mid-Think (`We`)	82.2	1,104
Mid-Think (`Okay`)	92.3	2,753
Think	94.6	5,557

Table 7: Sensitivity to trigger token choice on Qwen3-8B (MATH-500). A mismatched trigger causes Mid-Think to degenerate to No-Think behavior, confirming trigger-template alignment is essential.

to the training template. Replacing the Mid-Think trigger accordingly ("`Okay`" $\rightarrow$ "`We`") preserves Mid-Think's effectiveness, as shown in Table 6.

In contrast, applying a mismatched trigger (*e.g.*, using "`We`" on Qwen3-8B, which was trained with "`Okay`") causes Mid-Think to collapse to near No-Think behavior, as shown in Table 7. This confirms that the trigger must match the model's training template to be effective, and that the phenomenon originates from training data rather than model architecture.

3.5 Comparison with Fixed-Token and Prompt-Based Methods

This section compares our approach with existing training-free methods for achieving intermediate reasoning capability. Most prior approaches rely on additional training to control the reasoning budget. In contrast, the primary training-free alternatives are the fixed-token budget mechanism in the Qwen3 series and prompt-based budget constraints. We therefore compare Mid-Think against these two training-free baselines.

Experimental Setup. To compare against existing training-free budget control methods, we use Qwen3-14B as the backbone model and evaluate three baselines on MATH500. First, we adopt the fixed-token budget strategy provided by the Qwen3 series, setting the maximum generation length to 2k, 3k, and 4k tokens, respectively. Second, we evaluate the proposed Mid-Think mode under the same evaluation setting. Finally, we test prompt-based budget control by applying prompting templates designed to constrain or guide the reasoning process. The results are summarized in Table 8.

Method	Setting	Acc(%)	Avg Len
Original	No-Think	86.3	899
	Think	94.4	4904
Fixed Tokens	2k tokens	89.6	2673
	3k tokens	91.2	3315
	4k tokens	90.8	3793
	5k tokens	92.2	4136
Mid-Think	training-free	**92.1**	**2589**
Prompt-based	Prompt	91.2	3131

Table 8: Comparison of training-free budget control methods on MATH500 using Qwen3-14B. Fixed Tokens limits the number of tokens allocated to the reasoning process, while Prompt-based methods prepend explicit instructions encouraging shorter outputs while preserving accuracy.

Comparison Results. As shown in Table 8, while the Fixed Tokens strategy enables budget control by explicitly limiting the number of reasoning tokens, its control is relatively coarse. Under comparable accuracy levels, Mid-Think consistently achieves substantially shorter average generation lengths. Moreover, fixed tokens require pre-specifying the token limit without knowing the difficulty of each instance in advance, which often leads to insufficient reasoning for harder problems and consequently degrades accuracy, despite successfully constraining output length.

Prompt-based methods also underperform compared to Mid-Think. Although such prompts can partially reduce the reasoning budget, the resulting average generation length remains significantly higher than that achieved by Mid-Think, indicating limited controllability. In contrast, Mid-Think leverages overfitted token-level triggers to induce intermediate-budget reasoning behavior, providing a more reliable and fine-grained mechanism for balancing reasoning quality and output length.

4 Applying Mid-Think to RL Training after SFT

This section applies the proposed Mid-Think mode to RL training on top of supervised fine-tuning (SFT). Prior work has explored RL training using No-think settings (Xu et al., 2025b) or fixed-token budgets (Xu et al., 2025c) to accelerate and improve reasoning training. We show that the Mid-Think mode can also be directly trained via RL,

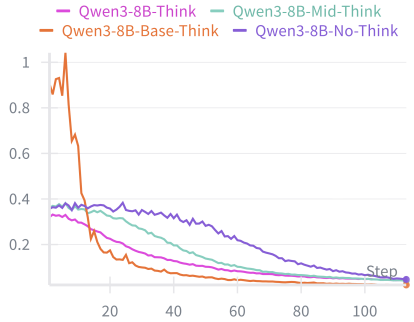


Figure 6: Entropy during GRPO. We compare Qwen3-8B-Base trained in the Think mode with Qwen3-8B trained under Think, No-think, and Mid-Think modes. The panel plots entropy versus training steps. Notably, Mid-Think both increases training entropy.

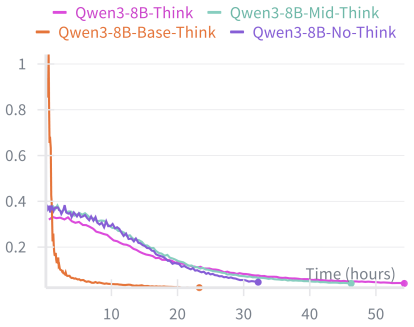


Figure 7: Entropy during GRPO. We compare Qwen3-8B-Base trained in the Think mode with Qwen3-8B trained under Think, No-think, and Mid-Think modes. The panel shows entropy versus relative training time. Notably, Mid-Think reduces overall training time.

enabling more efficient training while improving overall model performance.

4.1 Experimental Setup

In this section, we conduct RL training using Qwen3-4B and Qwen3-8B. All models are trained with the `verl` framework using the GRPO algorithm, with a learning rate of 1×10^{-6} for six training epochs. We set the maximum generation length to 16K tokens during training. All experiments are conducted using 8 NVIDIA H200 GPUs.

We use 2 training datasets, each consisting of 5,000 samples, sampled from Skywork-OR1-RL-Data (He et al., 2025a,b) and OpenScienceReasoning-2 (NVIDIA et al., 2025). Models trained on Skywork-OR1-RL-Data are evaluated on AIME, while models trained on OpenScienceReasoning-2 are evaluated on GPQA. This evaluation enables validation across both mathematical and scientific reasoning domains.

4.2 Experimental Results

4.2.1 Results of Training Entropy

We analyze the evolution of training entropy for Qwen3-8B under different training modes, as illustrated in Figure 6. Specifically, we report entropy curves on math training data for Qwen3-8B-Base trained in the Think mode, Qwen3-8B trained in the Think mode, and Qwen3-8B trained under the No-think and Mid-Think modes.

Qwen3-8B-Base trained in the Think mode exhibits the highest initial entropy, which rapidly decreases and eventually collapses. In contrast, Qwen3-8B trained in the Think mode starts with substantially lower entropy and remains in a low-entropy regime throughout training. We attribute this behavior to the fact that Qwen3-8B, after SFT, already exhibits a highly fixed reasoning pattern in the Think mode, thereby limiting exploration during RL training.

Notably, both the No-think and Mid-Think training modes lead to higher and more sustained entropy. This indicates that relaxing or partially disrupting the fixed reasoning pattern can effectively increase policy entropy, facilitating exploration and improving the stability of RL training.

4.2.2 Results of Training Time

Figure 7 reports the training time under different training modes. We observe that Qwen3-8B trained in the Think mode incurs long training time, as the model generates excessively long reasoning sequences during optimization. In contrast, Qwen3-8B-Base exhibits much shorter training time since it has not undergone prior fine-tuning and does not produce explicit reasoning traces.

Notably, both the No-think and Mid-Think modes significantly reduce training time by shortening the model's reasoning outputs. This demonstrates that controlling the reasoning budget not only affects learning dynamics but also leads to substantial gains in training efficiency.

4.2.3 Test Performance.

We report the test performance of Qwen3-8B and Qwen3-4B after RL training under different modes on MATH500, AIME, and GPQA, including accuracy, average generation length, and wait count, as summarized in Table 9.

Across all benchmarks, models trained in the standard Think mode consistently underperform those trained with the No-think and Mid-Think modes. Moreover, after No-think training, the

Model	Training	Mode	AIME						GPQA					
			No-think Test			Think Test			No-think Test			Think Test		
			Acc	Len	Wait	Acc	Len	Wait	Acc	Len	Wait	Acc	Len	Wait
Qwen3-8B	No	–	21.3	4520	447	71.6	16847	40342	37.5	1335	459	60.9	9008	71817
	RL	Think	19.3	7624	9423	69.8	13330	34701	34.7	1567	2179	58.5	9413	52228
	RL	No-Think	62.9	13591	11936	72.0	18980	34585	47.8	4435	9863	59.6	9047	59419
	RL	Mid-Think	27.6	7114	6055	**72.4**	15318	44142	36.2	1293	1968	**61.1**	8257	55272
Qwen3-4B	No	–	20.0	4605	542	68.7	16699	40089	36.9	1560	590	53.2	8729	72509
	RL	Think	17.6	7973	917	61.1	13347	32118	35.5	1789	252	47.3	10481	58476
	RL	No-Think	36.7	17421	20818	69.3	20551	61341	40.4	6871	17196	48.7	10774	82568
	RL	Mid-Think	24.9	12492	11271	**69.6**	15799	40216	38.2	2086	4790	**55.3**	8696	60790

Table 9: Performance of Qwen3-4B and Qwen3-8B after GRPO training under different modes (Think, No-think, and Mid-Think) on AIME and GPQA. We report accuracy, average generation length, and wait count. The Direct setting corresponds to the untrained model. No-think Test evaluates the model under the no-thinking mode, while Think Test evaluates the model under the standard thinking mode. Models trained with Mid-Think consistently outperform those trained with Think or No-think when evaluated in the Think mode, while largely preserving performance under the No-think test setting.

model fails to reliably preserve non-reasoning behavior and tends to produce excessively long outputs when evaluated in the Think mode.

In contrast, RL training with the proposed Mid-Think mode achieves a more favorable balance. It largely maintains the efficiency benefits of the No-think regime while simultaneously attaining the highest accuracy under the Think mode.

5 Related Works

Efficient LLM Reasoning. Recent reasoning models still face significant efficiency challenges, often producing excessively long outputs (Bandyopadhyay et al., 2025; Li et al., 2025; Wang et al., 2025b). Early approaches such as Kimi 1.5 (Team et al., 2025b) and Sky-Thought (Team, 2025) reduce verbosity by aligning long and short responses via preference optimization, while TokenSkip (Xia et al., 2025) and LightThinker (Zhang et al., 2025b) improve efficiency by pruning redundant tokens or compressing intermediate thoughts. Beyond shortening reasoning traces, hybrid thinking methods (Jiang et al., 2025; Liu et al., 2025a) aim to control *when* models reason, typically through explicit control tokens (e.g., \think, \nothink) (Sui et al., 2025; Chen et al., 2024; Yang et al., 2026). This paradigm has been adopted by models such as Gemini (Team et al., 2025a), Qwen3 (Yang et al., 2025a), GPT-oss (Agarwal et al., 2025; Zhang et al., 2025d; Yue et al., 2025), and DeepSeek V3.1 (Liu et al., 2024), with the latter further scaling hybrid thinking through large-scale RL.

Reinforcement Learning with Verifiable Rewards With the emergence of DeepSeek-R1 (Guo et al., 2025; Liu et al., 2024), GRPO has become a widely adopted approach for endowing language models with reasoning capabilities (Zhang et al., 2025c; Plaat et al., 2024; Xu et al., 2025a; Yang et al., 2025e). A growing body of work focuses on improving GRPO, including variants (Xi et al., 2025; Nan et al., 2025; Yang et al., 2025d) such as DAPO (Yu et al., 2025), Dr. GRPO (Liu et al., 2025b), and GSPO (Zheng et al., 2025). Other studies specifically address the issue of entropy collapse in GRPO-based training like *Rethinking Entropy Interventions* (Hao et al., 2025; Ganguly et al., 2026). Several works aim to reduce the high training cost of GRPO, like *It Takes Two* (Wu et al., 2025), *Thinking-Free Policy*. In parallel, recent research has examined the interaction between supervised fine-tuning (SFT) and GRPO-based RL like *On the Interplay* (Zhang et al., 2025a) and *Quagmires in SFT–RL Post-Training* (Kang et al., 2025; Yang et al., 2025c).

6 Conclusion

We show that hybrid thinking is largely driven by a small set of token-level triggers. Building on this, Mid-Think enables training-free intermediate-budget reasoning by strategically manipulating these triggers at inference time, achieving a better accuracy–efficiency trade-off than fixed-budget baselines. Beyond inference-time control, Mid-Think also proves effective as an RL training objective, yielding models that better balance Think and No-Think behavior. We hope these findings offer a lightweight and practical lens for understanding reasoning in hybrid thinking models.

Limitations

While Mid-Think enables intermediate-budget reasoning, it does not provide fully dynamic or fine-grained control over arbitrary budget levels (e.g., 0.1 or 0.7). Moreover, Mid-Think relies on the presence of existing overfitted token-level behaviors, and thus requires identifying such patterns in advance to realize Mid-Think.

Acknowledgements

This work was supported in part by NSF award 2117439. This research made use of the High Performance Computing Resource in the Core Facility for Advanced Research Computing at Case Western Reserve University (CWRU).

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Dibyanayan Bandyopadhyay, Soham Bhattacharjee, and Asif Ekbal. 2025. Thinking machines: A survey of llm based reasoning strategies. *arXiv preprint arXiv:2503.10814*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Thinkless: Llm learns when to think. *arXiv preprint arXiv:2505.13379*.

Debargha Ganguly, Sreehari Sankar, Biyao Zhang, Vikash Singh, Kanan Gupta, Harshini Kavuru, Alan Luo, Weicong Chen, Warren Morningstar, Raghu Machiraju, and 1 others. 2026. Trust the typical. *arXiv preprint arXiv:2602.04581*.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2024. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. 2025. Rethinking entropy interventions in rlvr: An entropy change perspective. *arXiv preprint arXiv:2510.10150*.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. 2025a. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Yang Liu, and Yahui Zhou. 2025b. Skywork open reasoner series. Notion Blog.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. 2025. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631*.

Feiyang Kang, Michael Kuchnik, Karthik Padthe, Marin Vlastelica, Ruoxi Jia, Carole-Jean Wu, and Newsha Ardalani. 2025. Quagmires in sft-rl post-training: When high sft scores mislead and what to use instead. *arXiv preprint arXiv:2510.01624*.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, and 1 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Keliang Liu, Dingkang Yang, Ziyun Qian, Weijie Yin, Yuchi Wang, Hongsheng Li, Jun Liu, Peng Zhai, Yang Liu, and Lihua Zhang. 2025a. Reinforcement learning meets large language models: A survey of advancements and applications across the llm lifecycle. *arXiv preprint arXiv:2509.16679*.

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

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. 2025. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*.

Gongrui Nan, Siye Chen, Jing Huang, Mengyu Lu, Dexun Wang, Chunmei Xie, Weiqi Xiong, Xianzhou Zeng, Qixuan Zhou, Yadong Li, and 1 others. 2025. Ngrpo: Negative-enhanced group relative policy optimization. *arXiv preprint arXiv:2509.18851*.

NVIDIA, :, Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, and 1 others. 2025. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model. *Preprint*, arXiv:2508.14444.

Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. 2024. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*.

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*.

Brian S Robinson, Nathan Drenkow, Colin Conwell, and Michael Bonner. 2024. A sparse null code emerges in deep neural networks. In *Proceedings of UniReps: The First Workshop on Unifying Representations in Neural Models*, pages 302–314. PMlR.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025a. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025b. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

NovaSky Team. 2025. Think less, achieve more: Cut reasoning costs by 50 https://novasky-ai.github.io/posts/reduce-overthinking. Accessed: 2025-01-23.

Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. 2025a. Wait, we don't need to" wait"! removing thinking tokens improves reasoning efficiency. *arXiv preprint arXiv:2506.08343*.

Shouren Wang, Wang Yang, Xianxuan Long, Qifan Wang, Vipin Chaudhary, and Xiaotian Han. 2025b. Demystifying hybrid thinking: Can llms truly switch between think and no-think? *arXiv preprint arXiv:2510.12680*.

Yihong Wu, Liheng Ma, Lei Ding, Muzhi Li, Xinyu Wang, Kejia Chen, Zhan Su, Zhanguang Zhang, Chenyang Huang, Yingxue Zhang, and 1 others. 2025. It takes two: Your grpo is secretly dpo. *arXiv preprint arXiv:2510.00977*.

Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, and 1 others. 2025. Bapo: Stabilizing off-policy reinforcement learning for llms via balanced policy optimization with adaptive clipping. *arXiv preprint arXiv:2510.18927*.

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.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, and 1 others. 2025a. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

Xin Xu, Cliveb AI, Kai Yang, Tianhao Chen, Yang Wang, Saiyong Yang, and Can Yang. 2025b. Thinking-free policy initialization makes distilled reasoning models more effective and efficient reasoners. *arXiv preprint arXiv:2509.26226*.

Yuhui Xu, Hanze Dong, Lei Wang, Doyen Sahoo, Junnan Li, and Caiming Xiong. 2025c. Scalable chain of thoughts via elastic reasoning. *arXiv preprint arXiv:2505.05315*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Rubing Yang, Huajun Bai, Song Liu, Guanghua Yu, Runzhi Fan, Yanbin Dang, Jiejing Zhang, Kai Liu, Jianchen Zhu, and Peng Chen. 2025b. Specexit: Accelerating large reasoning model via speculative exit. *arXiv preprint arXiv:2509.24248*.

Van Yang, Hongye Jin, Shaochen Zhong, Song Jiang, Qifan Wang, Vipin Chaudhary, and Xiaotian Han. 2025c. 100-longbench: Are de facto long-context benchmarks literally evaluating long-context ability? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17560–17576.

Van Yang, Xiang Yue, Vipin Chaudhary, and Xiaotian Han. 2025d. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time. In *Second Conference on Language Modeling*.

Wang Yang, Zirui Liu, Hongye Jin, Qingyu Yin, Vipin Chaudhary, and Xiaotian Han. 2025e. Longer context, deeper thinking: Uncovering the role of long-context ability in reasoning. *arXiv preprint arXiv:2505.17315*.

Wang Yang, Chaoda Song, Xinpeng Li, Debargha Ganguly, Chuang Ma, Shouren Wang, Zhihao Dou, Yuli Zhou, Vipin Chaudhary, and Xiaotian Han. 2026. Ace-bench: Agent configurable evaluation with scalable horizons and controllable difficulty under lightweight environments. *arXiv preprint arXiv:2604.06111*.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Linan Yue, Yichao Du, Yizhi Wang, Weibo Gao, Fangzhou Yao, Li Wang, Ye Liu, Ziyu Xu, Qi Liu, Shimin Di, and 1 others. 2025. Don't overthink it: A survey of efficient r1-style large reasoning models. *arXiv preprint arXiv:2508.02120*.

Charlie Zhang, Graham Neubig, and Xiang Yue. 2025a. On the interplay of pre-training, mid-training, and rl on reasoning language models. *arXiv preprint arXiv:2512.07783*.

Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. 2025b. Lightthinker: Thinking step-by-step compression. *arXiv preprint arXiv:2502.15589*.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, and 1 others. 2025c. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*.

Yiqun Zhang, Hao Li, Jianhao Chen, Hangfan Zhang, Peng Ye, Lei Bai, and Shuyue Hu. 2025d. Beyond gpt-5: Making llms cheaper and better via performance-efficiency optimized routing. In *Proceedings of the 2025 7th International Conference on Distributed Artificial Intelligence*, pages 122–129.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

A Appendix

A.1 Quantitative Attention Analysis

To further quantify this observation, we compute the *trigger-token attention mass* for each reasoning mode on Qwen3-8B. Specifically, we measure the average attention assigned to trigger tokens relative to all other tokens in the prompt, and define the ratio as:

$$\text{Ratio} = \frac{\text{Attention on Trigger Token}}{\text{Avg Attention on Other Tokens}} \quad (1)$$

Table 10 reports these ratios alongside MATH-500 accuracy and average generation length. In No-Think mode, the post-`</think>` newline token dominates with a ratio of 5.27×, suppressing reasoning. In Think mode, the "`Okay`" token dominates with a 3.18× ratio, activating reasoning. Crucially, Mid-Think simultaneously activates *both* trigger regions: "`Okay`" at 2.10× and the newline at 3.08×, resulting in intermediate accuracy (92.3%) and generation length between the two extremes. This quantitative evidence confirms that Mid-Think induces a genuinely hybrid attention pattern, rather than simply averaging the two modes.

Mode	Trigger	Ratio	Acc (%)	Avg Len
No-Think	`\n\n`	5.27×	83.2	1,012
Mid-Think	`\n\n`	3.08×	92.3	2,753
Mid-Think	`Okay`	2.10×	92.3	2,753
Think	`Okay`	3.18×	94.6	5,557

Table 10: Trigger-token attention mass ratio, MATH-500 accuracy, and average generation length under different reasoning modes on Qwen3-8B. Mid-Think jointly activates both trigger regions, yielding hybrid behavior.

A.2 Another Verification of Budget-Controlled Reasoning Baseline

Previously, we showed that the proposed budget-controlled reasoning method enables proportional control over the reasoning budget, resulting in systematic changes in average output length, accuracy, and wait count. In this section, we present the corresponding quantitative results. As shown in Tables 11 and 12, all three metrics increase monotonically with the reasoning budget, consistently exhibiting the expected budget–performance trade-off.

Budget	Avg Length	Wait	Acc (%)
0.0 (No-think)	899.1	329	86.3
0.1	1563.3	10252	87.4
0.2	1879.9	16225	89.2
0.3	2170.8	22293	91.6
0.4	2472.7	27339	92.2
0.5	2852.9	33080	92.8
0.6	3244.6	38788	93.6
1.0 (Think)	4904.3	59288	94.4

Table 11: Verification of budget-controlled method on MATH500 with Qwen3-14B. We report average output length, wait count, and accuracy under different budgets. As the budget increases, all three metrics improve steadily, demonstrating the effectiveness of the method.

Budget	Avg Length	Wait	Acc (%)
0.0 (No-think)	1012.6	2082	83.2
0.1	2145.3	19679	88.0
0.2	2466.6	26778	89.8
0.3	2709.0	33543	90.3
0.4	3085.2	40188	92.0
0.5	3456.7	52871	92.8
0.6	3788.3	53260	93.3
1.0 (Think)	5557.4	81367	94.6

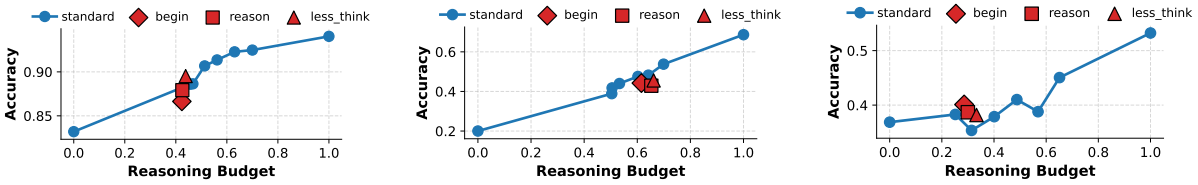
Table 12: Verification of budget-controlled reasoning on `MATH500` using `Qwen3-8B`. We report average output length, wait count, and accuracy under different budgets. As the reasoning budget increases, all three metrics improve steadily, demonstrating effective budget control.

A.3 Results of Mid-Think v.s. Different Budget

Previously, we presented the performance of Mid-Think across different model scales primarily through visualizations. To provide more concrete evidence, we now report the corresponding quantitative results. Tables 15, 14, and 13 summarize the detailed numerical results. Across all datasets, Mid-Think—regardless of the specific tag used—consistently achieves intermediate-budget reasoning behavior. Notably, on GPQA, certain Mid-Think variants even outperform standard budget scaling under comparable or lower reasoning cost.

A.4 Comparison Results on Qwen3-4B

Previously, we focused our analysis on hybrid models at the 8B and 14B scales. Here, we additionally report results on `Qwen3-4B`. As shown in Figure 5, Mid-Think continues to induce intermediate-budget reasoning behavior on `Qwen3-4B`, demonstrating that the effect generalizes to smaller model scales.



(a) Qwen3-4B on MATH500 (b) Qwen3-4B on AIME (c) Qwen3-4B on GPQA

Figure 8: Comparison between different budget reasoning and Mid-Think on pure-think and RL-based models across multiple datasets. We report results for Qwen3-4B under varying reasoning budgets, together with Mid-Think using different tags (<reason>, <begin>, <less think>). Across MATH500, AIME, and GPQA, Mid-Think consistently achieves performance corresponding to intermediate reasoning budgets, and on GPQA it surpasses fixed-budget baselines, yielding Pareto-optimal accuracy–efficiency trade-offs between Think and No-Think mode.

Format	Budget	Avg Length	Wait	Acc
standard	0.0	899.1	329	86.3
standard	0.1	1563.3	10252	87.4
standard	0.2	1879.9	16225	89.2
standard	0.3	2170.8	22293	91.6
standard	0.4	2472.7	27339	92.2
standard	0.5	2852.9	33080	92.8
standard	0.6	3244.6	38788	93.6
standard	1.0	4904.3	59288	94.4
begin	1.0	2805.5	23024	93.1
reason	1.0	2589.8	20933	92.1
less_think	1.0	2655.0	21249	93.3

Table 13: Quantitative results of budget-controlled reasoning on `MATH500` using `Qwen3-14B`. We report average output length, wait count, and accuracy under different reasoning budgets and prompt formats. Standard budget control exhibits a clear monotonic trade-off between budget and performance, while alternative formats (`begin`, `reason`, `less_think`) achieve comparable accuracy with substantially reduced reasoning length.

Format	Budget	Avg Length	Wait	Acc
standard	0.0	4084.9	501	22.9
standard	0.1	5884.0	7070	33.1
standard	0.2	6515.3	10911	37.8
standard	0.3	7784.5	14455	42.9
standard	0.4	8626.4	17597	48.9
standard	0.5	9449.8	20143	54.7
standard	0.6	10486.8	21793	57.6
standard	1.0	15792.0	31686	74.4
begin	1.0	11717.4	17083	61.3
reason	1.0	10862.2	14660	58.7
less_think	1.0	10747.9	14893	61.1

Table 14: Results of budget-controlled reasoning on `AIME` using `Qwen3-14B`. We report average output length, wait count, and accuracy under different reasoning budgets and prompt formats. Increasing the budget leads to substantial performance gains, while alternative formats (`begin`, `reason`, `less_think`) achieve competitive accuracy with reduced reasoning cost compared to standard full-thinking.

A.5 Mid-Think to RL Training after SFT on MATH500

Previously, we only reported results on AIME. We now extend our evaluation to **Math500** and examine the effect of applying **Mid-Think** as the reinforcement learning objective. Across both `Qwen3-8B` and `Qwen3-4B`, Mid-Think training consistently improves performance under the *Think* test setting, while largely preserving behavior under the *No-think* test setting.

As shown in Tables 16 and 17, models trained with standard Think supervision remain inferior to those trained with Mid-Think when evaluated in Think mode. Meanwhile, Mid-Think training maintains competitive accuracy under No-think evaluation, indicating that it serves as an effective intermediate reasoning objective that balances reasoning strength and inference efficiency.

A.6 Training Hyperparameters

To facilitate reproducibility, we summarize the full set of training hyperparameters in Table 18. Our setup follows the official VERL GRPO recipe, and all three training modes (Think, No-Think, and Mid-Think) use identical hyperparameters to ensure a fair comparison.

Response sampling. We sample 8 responses per prompt during rollout, which provides sufficient diversity for advantage estimation in GRPO while remaining computationally tractable. The maximum response length is capped at 16,384 tokens to accommodate long chain-of-thought outputs under the Think setting, while also allowing Mid-Think responses to naturally vary in length without artificial truncation.

Training data and epochs. We train on a curated set of 5,000 problems for 6 epochs. This

Format	Budget	Avg Length	Wait	Acc
standard	0.0	1211.4	398	39.4
standard	0.1	1756.2	7836	41.5
standard	0.2	2351.2	13849	41.7
standard	0.3	3005.8	20242	42.0
standard	0.4	3606.8	25896	47.7
standard	0.5	4208.1	30162	46.9
standard	0.6	5000.8	35780	47.4
standard	1.0	7799.4	50661	63.1
begin	1.0	2160.2	6138	50.4
reason	1.0	1763.0	4159	53.9
less_think	1.0	2073.1	4999	50.4

Table 15: Results of budget-controlled reasoning on `GPQA` using `Qwen3-14B`. We report average output length, wait count, and accuracy under different reasoning budgets and prompt formats. Standard budget control shows increasing performance with higher budgets, while alternative formats achieve competitive accuracy with substantially reduced reasoning cost.

Training	Mode	Test	Acc / Len
NO	–	No-think	83.2 / 1013
		Think	94.6 / 5557
RL	Think	No-think	81.9 / 1392
		Think	93.6 / 4568
RL	No-Think	No-think	91.6 / 2268
		Think	93.6 / 6335
RL	Mid-Think	No-think	85.8 / 1374
		Think	**94.1** / 5302

Table 16: Performance of `Qwen3-8B` under different training regimes. We report accuracy (%) and average generation length. Each trained model is evaluated under both No-think and Think test settings. Reason (Mid-Think) training achieves the best Think-Test accuracy with balanced generation length.

relatively compact dataset size is intentional: it reduces the risk of reward hacking on easy samples while maintaining a sufficient diversity of problem types. Training for 6 epochs strikes a balance between convergence and overfitting, as we observed diminishing returns beyond this point in preliminary experiments.

Optimization settings. We use a learning rate of 1×10^{-6} with a warmup ratio of 0.05 and weight decay of 0.1. The small learning rate is chosen conservatively to preserve the pretrained model's language generation capabilities while allowing the policy to shift meaningfully under RL. The weight decay serves as a regularizer to prevent large deviations from the reference policy, complementing the KL penalty already present in GRPO.

Training	Mode	Test	Acc / Len
No	–	No-think	83.2 / 991
		Think	94.0 / 5334
RL	Think	No-think	80.8 / 1362
		Think	92.4 / 4061
RL	No-Think	No-think	88.4 / 3234
		Think	93.9 / 7458
RL	Mid-Think	No-think	83.9 / 1852
		Think	93.9 / 5456

Table 17: Performance of `Qwen3-4B` on **Math500** under different training regimes. We report accuracy (%) and average generation length. Each model is evaluated under both No-think and Think test settings. Mid-Think achieves strong Think-Test accuracy while better preserving No-think behavior compared to standard Think training.

Hyperparameter	Value
Responses per prompt	8
Max response length (tokens)	16,384
Training epochs	6
Training data size	5,000
Learning rate	1e-6
LR warmup ratio	0.05
Weight decay	0.1
Train batch size	256
PPO mini-batch size	64
Micro-batch size per GPU	16

Table 18: Training hyperparameters for GRPO-based RL experiments. All three training modes use identical settings.

Batch configuration. The train batch size is 256, decomposed into PPO mini-batches of 64, with a micro-batch size of 16 per GPU. This hierarchical batching strategy is standard in large-scale RLHF pipelines and allows gradient accumulation across multiple forward passes while keeping per-device memory usage manageable.