Reasoning Pattern Alignment Merging for Adaptive Reasoning

Anonymous ACL submission

Abstract

Recent large reasoning models (LRMs) have made substantial progress in complex reasoning tasks, yet they often generate lengthy reasoning paths for every query, incurring unnecessary computation and latency. Existing speed-up approaches typically rely on retraining the model or designing sophisticated prompting, which are either prohibitively expensive or highly sensitive to the input and prompt formulation. In this work, we study model merging as a lightweight alternative for efficient reasoning: by combining a long chain-of-thought (Long-CoT) reasoning model with a Short-CoT instruction model, we obtain an adaptive reasoner without training from scratch or requiring large-scale additional data. Building on this idea, we propose Reasoning Pattern Alignment Merging (RPAM), a layer-wise model merging framework based on feature alignment to facilitate query-adaptive reasoning. RPAM first constructs a small pattern-labeled calibration set that assigns each query an appropriate reasoning pattern. It then optimizes layer-wise merging coefficients by aligning the merged model's intermediate representations with those of the selected model, while a contrastive objective explicitly pushes them away from the non-selected model. Experiments on seven widely used reasoning benchmarks show that RPAM substantially reduces inference cost while maintaining strong performance.

1 Introduction

Large reasoning models (LRMs) have recently achieved strong performance on complex reasoning tasks, ranging from mathematical problem solving (Lewkowycz et al., 2022; Pan et al., 2024; Shao et al., 2024) and logical deduction (Wei et al., 2022) to agentic assistants (Wu et al., 2025b; Chen et al., 2025a). A key driver is their ability to generate long chains of thought (Long-CoT), in which the model iteratively self-assesses, mitigates errors, and verifies intermediate steps before producing a final answer (Liu et al., 2024; Yu et al., 2024b; Li et al., 2025). However, while Long-CoT is beneficial for difficult problems, it can be counterproductive on simple tasks that require few reasoning steps: models may "overthink" by introducing unnecessary intermediate reasoning (Muennighoff et al., 2025; Zhang et al., 2025b). This overthinking not only increases inference cost (Fan et al., 2025; Chen et al., 2025b; Cuadron et al., 2025), but can also hurt accuracy by amplifying the chance of spurious reasoning and obscuring the straightforward solution (Shen et al., 2025).

Many recent studies have explored how to mitigate the inefficiency and overthinking of LRMs (Feng et al., 2025; Srivastava et al., 2025; Sui et al., 2025). A prominent direction is adaptive reasoning, where a model dynamically selects between Long-CoT and Short-CoT modes depending on the estimated problem complexity (Zhang et al., 2025a; Alomrani et al., 2025). Broadly, existing adaptive-reasoning approaches fall into two paradigms. Training-based methods optimize LRMs to exhibit adaptive thought processes via supervised fine-tuning (SFT) (Yu et al., 2025; Ma et al., 2025) or reinforcement learning (RL) (Fang et al., 2025; Wu et al., 2025c), but they typically require large-scale data and incur substantial training cost. Training-free methods, often implemented through prompt-guided strategies (Xu et al., 2025b), can introduce adaptivity without additional optimization, yet they rely heavily on instruction following and can be sensitive to prompt constraints and query phrasing (Zhu et al., 2025; Zhu and Li, 2025). As a result, achieving adaptive reasoning that is both efficient and effective remains non-trivial.

In light of this, we investigate model merging as an alternative route toward adaptive reasoning. Model merging can integrate complementary capabilities from multiple task- or style-specific models into a single model without training from scratch

and often without requiring large-scale additional data (Wortsman et al., 2022; Ilharco et al., 2023). This property makes it a natural fit for adaptive reasoning, since Long-CoT and Short-CoT behaviors are commonly learned in separate models. Nevertheless, most prior merging work primarily targets improved overall accuracy of a single merged model (Wu et al., 2025a), rather than explicitly optimizing for query-adaptive reasoning efficiency. Recent work, such as ACM (Yao et al., 2025), moves toward efficiency by shortening reasoning trajectories globally across queries, which may be suboptimal for inputs that genuinely require long-horizon deliberation.

Motivated by this gap, we aim for *query-level* adaptive reasoning via model merging: the merged model should produce long, detailed reasoning when necessary, but remain concise on simple inputs. To this end, we propose *Reasoning Pattern Alignment Merging* (RPAM), a progressive, layer-wise merging framework with a contrastive objective. Starting from a Long-CoT reasoning model and a Short-CoT instruction model, we first construct a small pattern-labeled dataset in which each query is automatically assigned a reasoning-pattern label (Long-CoT vs. Short-CoT) based on which model achieves higher expected correctness. We then optimize layer-wise merging coefficients so that the merged model's intermediate features align with those of the selected (positive) model for each instance, while a contrastive loss explicitly pushes them away from the non-selected (negative) model. This yields a merged model that preserves strong performance on challenging problems while avoiding unnecessary long reasoning on simple ones, reducing inference cost without sacrificing accuracy. We evaluate our approach across seven commonly used reasoning benchmarks. Empirical results show that our method reduces token generation by 48% on the MATH dataset (Hendrycks et al., 2021) while simultaneously improving accuracy. Furthermore, on the more challenging OlympiadBench (He et al., 2024), our approach achieves a 50% reduction in inference cost with only negligible accuracy degradation.

Our contributions are summarized as follows:

- We introduce a model-merging-based paradigm for query-level adaptive reasoning that balances reasoning depth and efficiency within a single merged model.
- We develop a simple yet effective merging recipe, a small pattern-labeled dataset set together with feature alignment and contrastive separation, to learn layer-wise merging coefficients that integrate Long-CoT and Short-CoT behaviors.
- Through extensive experiments, we demonstrate that RPAM achieves a stronger accuracy and efficiency trade-off than competitive model-merging and adaptive-reasoning baselines, attaining near Long-CoT performance while substantially reducing generated tokens.

2 Related Work

2.1 Adaptive Reasoning

Adaptive reasoning improves efficiency by dynamically selecting an appropriate reasoning pattern conditioned on query difficulty (Sui et al., 2025; Zhu and Li, 2025). Existing approaches broadly fall into two lines: prompt-guided and training-based methods. **Prompt-guided** approaches (Renze and Guven, 2024; Gong et al., 2025; Xu et al., 2025b) exploit the instruction-following capability of LRMs by imposing explicit constraints through carefully crafted prompts to elicit concise reasoning paths. For instance, CoUT (Gong et al., 2025) first guides LRMs to internalize thinking processes and then generates the final answer using several token-efficient decoding strategies. However, their effectiveness can be brittle, as it depends heavily on prompt design, constraint choices, and the model's instruction-following behavior. **Training-based** methods explicitly optimize models to control or prune response length, e.g., by fine-tuning LRMs on variable-length CoT supervision (Luo et al., 2025; Qiao et al., 2025) or by applying reinforcement learning with length-aware rewards (Zhang et al., 2025a; Arora and Zanette, 2025). For example, Arora and Zanette (2025) post-train LRMs with RL using a length-penalty reward that combines correctness and normalized length of sampled solutions, assigning higher reward scores to shorter correct solutions. While effective, these approaches typically require substantial training data and incur high computational cost.

2.2 Model Merging

Model merging (Yang et al., 2024a) is an emerging technique that combines parameters from multiple pretrained or fine-tuned models with complementary capabilities into a single model, typically with-

out access to the original training data and without expensive end-to-end retraining. Early work such as Model Soups (Wortsman et al., 2022) demonstrates that simple weight averaging over multiple checkpoints can improve overall performance. Beyond averaging, more advanced methods, including TIES-Merging (Yadav et al., 2023) and DARE (Yu et al., 2024a), aim to reduce interference among task vectors by selectively retaining and combining only the most salient parameter updates. Building on these observations, representation surgery (Yang et al., 2024b) shows that naive merging can induce representation bias and proposes post-merging calibration by aligning the merged model's representations with those of task-specific models.

More recently, several studies (Wu et al., 2025a; Yao et al., 2025; Wu et al., 2025d) have explored model merging for efficient reasoning by combining a slow-thinking Long-CoT model with a fast-thinking Short-CoT model. These approaches typically aim to produce a single merged model that globally shortens reasoning paths while retaining accuracy. In contrast, our work focuses on query-level adaptive reasoning: we calibrate the merged model to selectively follow Long-CoT or Short-CoT patterns depending on the input, thereby improving the accuracy and efficiency trade-off.

3 Preliminary

3.1 Problem Formulation

Given a problem query x, an LLM parameterized by θ generates a corresponding solution $y = [y^1, ..., y^k]$ by sampling from the conditional distribution $\pi_\theta(\cdot|x)$. This distribution factorizes autoregressively as:

$$\pi_\theta(y|x) = \prod_{i=1}^{k} \pi_\theta(y^i \mid x, y^{<i}). \tag{1}$$

To improve the solution quality on reasoning tasks, Chain-of-Thought (CoT) prompting has been widely adopted, as it encourages models to produce intermediate steps that facilitate self-evaluation and verification. According to the granularity of the generated steps, CoT reasoning can be classified into Long-CoT and Short-CoT (Luo et al., 2025). Long-CoT yields detailed, reflective reasoning that benefits complex queries but increases generation length and latency. Short-CoT, in contrast, produces concise reasoning (or even direct answers) with lower cost, but can struggle on harder problems that require multi-step deliberation.

In this paper, we consider two LLMs: a Long-CoT reasoning model θ_L and a Short-CoT instruction model θ_S. Our goal is to construct a new model θ_M that can adapt its reasoning pattern to the query, producing Long-CoT when deeper reasoning is necessary and Short-CoT when a concise response suffices.

3.2 Model Merging

Model merging aims to combine N well-trained neural network models $\{\theta_i\}_{i=1}^N$, typically specialized for different tasks or behaviors, into a single parameter set θ_M without additional end-to-end retraining, while preserving the strengths of the constituent models. A simple and widely used approach is linear weight merging:

$$\theta_M = \sum_{i=1}^{N} \lambda_i \theta_i, \tag{2}$$

where λ_i controls the contribution of model θ_i (often with $\sum_i \lambda_i = 1$). However, because the coefficients are usually pre-defined, naive linear merging can introduce representation bias and yield suboptimal performance (Yang et al., 2024b). To alleviate this issue, post-calibration merging methods have been proposed (Dai et al., 2025; Xu et al., 2025a), which make the merging coefficients learnable and calibrate the merged model toward task-specific optima, thereby reducing knowledge loss.

In this work, we view the Long-CoT model and the Short-CoT model as two complementary, task-specialized models, and design a post-calibration model merging method to construct a single merged model θ_M that can adapt its reasoning behavior to the input query.

4 Proposed Method

In this paper, we propose RPAM, a reasoning-pattern-aware model merging framework that integrates the complementary behaviors of a Long-CoT reasoning model and a Short-CoT instruction model. The goal is to obtain a single merged model that can adapt its reasoning style to the input: using deeper, longer reasoning when necessary, while remaining concise and efficient on simpler queries. As illustrated in Fig. 1, RPAM consists of two components: (i) pattern-labeled data construction, (ii) feature-alignment merging with contrastive enhanced shaping. In the following parts, we present technical details for each design.

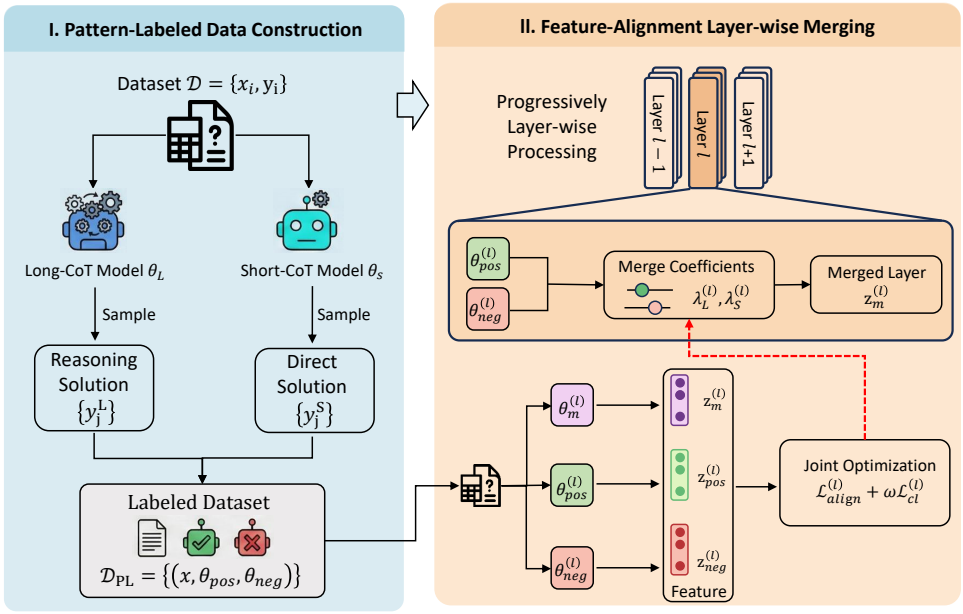

Figure 1: Overview of RPAM. The framework consists of two components: (I) Constructing a pattern-labeled (PL) dataset that identifies the optimal reasoning pattern (Long-CoT or Short-CoT) for each query x; (II) Performing layer-wise merging via feature alignment and contrastive enhanced shaping.

4.1 Pattern-Labeled Data Construction

A key challenge in adaptive reasoning is the lack of a dataset, which can teach models to determine, for each query, which reasoning style is more appropriate. To address this problem, we construct a small pattern-labeled dataset that empirically compares the effectiveness of the Long-CoT and Short-CoT models on each query.

Concretely, given a seed set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, we sample k solutions per query from both the Long-CoT model θ_L and the Short-CoT model θ_S. Let $\{y_j^L(x_i)\}_{j=1}^k$ and $\{y_j^S(x_i)\}_{j=1}^k$ denote the sampled outputs from θ_L and θ_S for query x_i, respectively. We define the empirical expected accuracy of a model $\theta \in \{\theta_L, \theta_S\}$ on input x_i as

$$\mathbb{E}(\theta, x_i) = \frac{1}{k} \sum_{j=1}^{k} \mathbb{I}[\, y_j(x_i) == y_i] \,, \qquad (3)$$

where $\mathbb{I}[\cdot]$ is an indicator function. We then select the one with the higher expected accuracy as the positive model for the query x:

$$\theta_{\text{pos}} = \underset{\theta \in \{\theta_L, \theta_S\}}{\arg\max}\, \mathbb{E}(\theta, x_i). \qquad (4)$$

If the expected accuracy is equal, we choose the model that generates fewer tokens. Then the other model is designated as the negative model θ_{neg}. This yields a pattern-labeled (PL) dataset:

$$\mathcal{D}_{\text{PL}} = \left\{ \left(x, \theta_{\text{pos}}, \theta_{\text{neg}}\right) \mid x \in \mathcal{D} \right\}. \qquad (5)$$

Intuitively, $\mathcal{D}_{\text{PL}}$ provides a query-dependent supervision signal for merging, specifying which model behavior should be emulated (and which should be avoided) for each input.

4.2 Feature-Alignment Layer-wise Merging

A naive linear weight interpolation between θ_L and θ_S may blur their behaviors and fail to perform the desired query-dependent reasoning style. Following the insight from (Xu et al., 2025a) that effective merging can be transformed as matching models' intermediate representations, RPAM is designed to learn layer-wise merging coefficients using the constructed pattern-labeled dataset $\mathcal{D}_{\text{PL}}$.

Specifically, let $\lambda_L^{(l)}$ and $\lambda_S^{(l)}$ be the merging coefficients at layer l, and let $\theta_L^{(l)}$, $\theta_S^{(l)}$ denote the parameters of layer l in the two reasoning models. For a given input x, let $z_M^{(l-1)}$ and $z_{\text{pos}}^{(l-1)}$ be the inputs to layer l in the merged model and the positive model, respectively. We denote $\varphi^{(l)}(\cdot)$ as layer feature mapping. The layer-l features are:

$$\begin{aligned} \theta_M^{(l)} &= \lambda_L^{(l)} \theta_L^{(l)} + \lambda_S^{(l)} \theta_S^{(l)} \\ z_M^{(l)} &= \varphi^{(l)}\left(\theta_M^{(l)},\ z_M^{(l-1)}\right) \\ z_{\text{pos}}^{(l)} &= \varphi^{(l)}\left(\theta_{\text{pos}}^{(l)},\ z_{\text{pos}}^{(l-1)}\right) \end{aligned} \qquad (6)$$

We then minimize the squared ℓ_2 distance between

Algorithm 1 Reasoning Pattern Alignment Merging

Require: task-specific models θ_L, θ_S; dataset $\mathcal{D}$; hyper-parameters k, τ, ω

Initialize: $\theta_M^{(l)} \leftarrow \lambda_L^{(l)} \cdot \theta_L^{(l)} + \lambda_S^{(l)} \cdot \theta_S^{(l)}$
Sample k responses $\{y_j^L(x_i)\}_{j=1}^k$ and $\{y_j^S(x_i)\}_{j=1}^k$ to calculate $\mathbb{E}(\theta, x_i)$ for $x_i \in \mathcal{D}$ (Eq. 3)
Construct $\mathcal{D}_{\text{PL}} \leftarrow \{(x, \theta_{\text{pos}}, \theta_{\text{neg}})\}$ (Eq. 5)
for $l = 1, \cdots, L$ **do**
    Calculate the internal features of the three models $z_M^{(l)}$, $z_{pos}^{(l)}$, $z_{neg}^{(l)}$ following Eq. (6)
    Calculate the losses $\mathcal{L}_{\text{align}}^{(l)}$, $\mathcal{L}_{\text{cl}}^{(l)}$ and $\mathcal{L}^{(l)}$ following Eq. (7), Eq. (8), and Eq. (9)
    Update the merging coefficients $\lambda_L^{(l)}, \lambda_S^{(l)}$ by minimizing $\mathcal{L}^{(l)}$
end for
return Merged model θ_M

the merged feature and the positive model feature:

$$\mathcal{L}_{\text{align}}^{(l)} = \left\| z_M^{(l)} - z_{\text{pos}}^{(l)} \right\|^2. \tag{7}$$

By optimizing $\{\lambda_L^{(l)}, \lambda_S^{(l)}\}$ under $\mathcal{L}_{\text{align}}^{(l)}$, RPAM encourages the merged model to inherit the internal representations of the positive model, and thus the reasoning behavior, of the query-selected reasoning pattern (Long-CoT or Short-CoT).

4.3 Contrastive Enhanced Shaping

The above feature alignment encourages the merged model's layer representations to move toward the positive reasoning model's representations; however, it does not explicitly prevent the merged model from also resembling the negative model. To more clearly separate the two reasoning patterns, we introduce a lightweight contrastive objective that pulls the merged model's representation closer to the positive model's while pushing it away from the negative model's. Specifically, we use $z_{\text{neg}}^{(l)}$ to denote the layer-l feature of the negative model for the input x. We define a binary contrastive loss:

$$\mathcal{L}_{\text{cl}}^{(l)} = -\log \frac{\exp\left(z_M^{(l)\top} z_{\text{pos}}^{(l)} / \tau\right)}{\sum_{N \in \{\text{pos,neg}\}} \exp\left(z_M^{(l)\top} z_N^{(l)} / \tau\right)}, \tag{8}$$

where τ is a temperature hyperparameter.

Finally, for each layer l, we optimize a joint objective that combines feature alignment with contrastive shaping:

$$\mathcal{L}^{(l)} = \mathcal{L}_{\text{align}}^{(l)} + \omega \cdot \mathcal{L}_{\text{cl}}^{(l)}, \tag{9}$$

where ω controls the strength of the contrastive term. The complete lightweight calibration and merging procedure is summarized in Algorithm 1.

5 Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of RPAM on seven reasoning benchmarks and across multiple model scales.

5.1 Experimental Setup

Datasets. We evaluate RPAM on seven widely used reasoning benchmarks. Following prior practice (Luo et al., 2025; Yao et al., 2025), we treat GSM8K (Cobbe et al., 2021) test set, MATH500 (Hendrycks et al., 2021) test set, and AIME24 (Zhang and Math-AI, 2024) as in-distribution (ID) evaluation data. We further assess its generalization performance on out-of-distribution (OOD) benchmarks, including AIME25 (Zhang and Math-AI, 2025), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and GPQA (Rein et al., 2024).

Metrics. We report both **accuracy** and **response length** (average number of generated tokens) on each benchmark to jointly characterize solution quality and inference cost. In addition, we also report the average accuracy change and the average response length reduction across all benchmarks, using the Long-CoT model as the reference. All evaluations are conducted with the AReal evaluation framework (Fu et al., 2025). For each benchmark, we independently execute the model five times with different random seeds and report the averaged results, and all results are statistically significant at $p < 0.05$.

Baselines. We compare RPAM against: (a) two base models, the Long-CoT and Short-CoT models; (b) representative data-free model-merging baselines, including **Average Merging** (Worts-

Model \ Bench	GSM8K	MATH500	Minerva Math	Olympiad Bench	AIME24	AIME25	GPQA	Avg.	Imp.(%)
Qwen3-4B-Thinking	95.2	96.0	60.3	72.9	83.1	80.3	67.7	79.3	-
(Long-CoT)	(1521)	(6125)	(5501)	(14405)	(20469)	(23912)	(9032)	(11566)	-
Qwen3-4B-Instruct	93.0	94.5	43.0	62.8	66.7	50.0	54.5	66.4	-16.37%
(Short-CoT)	(374)	(1670)	(1446)	(4199)	(7046)	(7368)	(5086)	(3884)	(-66.42%)
Prompt-guided									
CoD	95.6	94.8	57.7	56.3	**76.7**	**76.7**	65.2	74.7	-5.86%
	(1017)	(3106)	(3890)	(5887)	(13331)	(17338)	(6733)	(7329)	(-36.64%)
Training-based									
Ada-R1	95.3	**96.4**	58.5	71.4	72.2	68.9	65.7	75.5	-4.87%
	(1161)	(3746)	(4115)	(8455)	(11324)	(11969)	(6395)	(6738)	(-41.75%)
Data-free Merging									
Average Merging	95.4	95.4	57.7	70.5	73.3	57.8	64.7	73.5	-7.32%
	(1088)	(3422)	(4104)	(7293)	(9867)	(10099)	(7680)	(6222)	(-46.21%)
Task Arithmetic	95.2	96.2	57.4	71.3	72.2	67.8	62.1	74.6	-5.98%
	(1181)	(3720)	(3816)	(7841)	(10935)	(11395)	(6191)	(6440)	(-44.32%)
TIES Merging	94.8	95.2	**58.8**	70.2	75.6	60.0	63.6	74.0	-6.70%
	(1170)	(3411)	(3782)	(7168)	(9735)	(10891)	(6598)	(6108)	(-47.19%)
DARE-Linear	95.1	94.2	57.0	67.7	64.4	56.7	58.6	70.5	-11.11%
	(1641)	(3858)	(4559)	(7583)	(11677)	(12247)	(6404)	(6853)	(-40.75%)
Data-dependent Merging									
AIM	95.1	94.8	55.9	70.5	64.4	60.0	**70.7**	73.1	-7.92%
	(1060)	(2923)	(3874)	(6566)	(10354)	(9934)	(8007)	(6103)	(-47.24%)
ACM	94.7	94.0	56.2	68.4	70.0	54.4	61.6	71.3	-10.10%
	(925)	(3208)	(3197)	(6891)	(10988)	(11080)	(6141)	(6061)	(-47.59%)
RPAM	**95.7**	**96.4**	57.0	**72.0**	75.6	67.8	66.7	**75.9**	**-4.37%**
	(1086)	(3191)	(4169)	(7225)	(8394)	(10157)	(7612)	(5976)	**(-48.33%)**

Table 1: Evaluations of different methods on Qwen3-4B series models. The number in () represents the average response length on the benchmark. The average accuracy change and length reduction rate are compared with the Long-CoT (Qwen3-4B-Thinking) baseline. The best performance is highlighted in bold.

man et al., 2022), **Task Arithmetic (TA)** (Ilharco et al., 2023), **TIES-Merging** (Yadav et al., 2023), **DARE** (Yu et al., 2024a); (c) two data-dependent model-merging baselines, **Activation-informed Merging (AIM)** (Nobari et al., 2025) and **Activation-Guided Consensus Merging (ACM)** (Yao et al., 2025); (d) a prompt-guided efficiency method, **Chain-of-Draft (CoD)** (Xu et al., 2025b), which encourages concise reasoning by constraining each thinking step to five words; and (e) a training-based adaptive reasoning method, **Ada-R1** (Luo et al., 2025), which first merges Long-CoT and Short-CoT models and then applies DPO on a constructed preference dataset.

Implementation Details. To examine our method's generalization ability across model families and scales, we consider two Long/Short-CoT pairs: (i) *Qwen3-4B-Thinking* (Long-CoT) and *Qwen3-4B-Instruct* (Short-CoT); (ii) *DeepSeek-R1-Distill-Qwen-1.5B* (Long-CoT) and *Qwen2.5-Math-1.5B* (Short-CoT). For pattern-labeled dataset construction, we randomly sample a total of 128 questions from the GSM8K, MATH500, and AIME24 test sets, and draw k=12 responses per question from each of the two base models. We initialize the merged model using linear weight interpolation with λ=0.5, i.e., equal contributions from the Long-CoT and Short-CoT models. We select the learning rate and the number of training epochs via grid search over $\{0.1, 0.01, 0.001\}$ and $\{50, 100\}$, respectively. All experiments are conducted on a single NVIDIA RTX A5000 GPU.

5.2 Main Results

For the two model pairs of different sizes described in the Implementation Details, the corresponding results are reported in Tables 1 and 2, respectively.

As expected, the Long-CoT models (Qwen3-4B-Thinking and DeepSeek-R1-1.5B) deliver strong performance, but at the cost of substantially longer responses. In contrast, the Short-CoT models (Qwen3-4B-Instruct and Qwen2.5-Math-1.5B) produce much shorter outputs but suffer notable accuracy degradation, particularly on more challenging benchmarks such as AIME24 and AIME25. This trade-off between accuracy and token efficiency motivates query-level adaptive reasoning: remaining concise when possible while retaining deep reasoning for genuinely difficult queries.

From the experimental results in the 4B set-

Model \ Bench	GSM8K	MATH500	Minerva Math	Olympiad Bench	AIME24	AIME25	GPQA	Avg.	Imp.(%)
DeepSeek-R1-1.5B	79.0	80.6	30.7	30.4	29.2	25.6	34.5	44.3	-
(Long-CoT)	(978)	(67)	(4948)	(7389)	(17465)	(13118)	(9492)	(7637)	-
Qwen2.5-Math-1.5B	75.9	36.2	11.4	22.8	0.0	1.1	19.7	23.9	-46.09%
(Short-CoT)	(118)	(411)	(1037)	(608)	(865)	(1119)	(743)	(700)	(-90.83%)
Prompt-guided									
CoD	83.2	**80.2**	**39.7**	**41.6**	20.0	16.7	27.3	44.1	-0.41%
	(1682)	(3486)	(3909)	(6164)	(8426)	(8913)	(5589)	(5453)	(-28.60%)
Training-based									
Ada-R1	**85.1**	76.8	23.5	40.9	16.7	16.7	20.0	40.0	-9.77%
	(324)	(887)	(3021)	(2398)	(3931)	(4707)	(3655)	(2703)	(-64.60%)
Data-free Merging									
Average Merging	77.5	74.8	32.7	36.0	13.3	13.3	28.3	39.4	-10.99%
	(333)	(934)	(956)	(1659)	(3191)	(3735)	(1708)	(1788)	(-76.59%)
Task Arithmetic	73.7	62.8	27.6	29.2	16.7	3.3	20.7	33.4	-24.51%
	(377)	(692)	(862)	(898)	(3158)	(3713)	(1044)	(1535)	**(-79.90%)**
TIES Merging	78.4	72.6	33.8	36.0	10.0	16.7	24.2	38.8	-12.34%
	(505)	(1619)	(1181)	(1714)	(2901)	(2554)	(2188)	(1809)	(-76.31%)
DARE-Linear	67.4	57.2	22.1	27.0	10.0	3.3	19.2	29.5	-33.48%
	(394)	(780)	(872)	(987)	(2451)	(3417)	(4569)	(1924)	(-74.80%)
Data-dependent Merging									
AIM	83.7	76.0	36.8	40.0	16.7	13.3	26.8	41.9	-5.39%
	(328)	(1071)	(1024)	(2068)	(4022)	(3199)	(2129)	(1977)	(-74.11%)
ACM	78.4	78.8	37.5	39.4	10.0	16.7	27.8	41.2	-6.89%
	(398)	(1638)	(1761)	(3452)	(2689)	(5535)	(3815)	(2755)	(-63.92%)
RPAM	81.4	78.8	38.2	39.9	**26.7**	**20.0**	**28.8**	**44.8**	**1.21%**
	(330)	(1427)	(1504)	(2915)	(4984)	(4393)	(3670)	(2746)	(-64.04%)

Table 2: Evaluations of different methods on Qwen2.5-1.5B series models.

Model \ Bench	GSM8K	MATH500	AIME24	Avg.
RPAM	95.7	96.4	75.6	-2.41%
	(1086)	(3191)	(8394)	(-54.94%)
-CES	95.8	95.6	75.6	-2.66%
	(1090)	(3392)	(9061)	(-51.83%)
-CES-FA	95.4	95.4	73.3	-3.70%
	(1088)	(3422)	(9867)	(-48.86%)
+Random	94.2	95.0	66.3	-6.82%
	(392)	(1673)	(7354)	(-66.50%)

Table 3: Ablation study of each component (Contrastive Enhanced Shaping (CES), Feature Alignment (FA), and replace Pattern-Labeled Data with Random Data) on three benchmarks with Qwen3-4B models. Similar results are also observed on other benchmarks.

ting (Tables 1), we observe that the prompt-guided method CoD largely preserves accuracy but does not substantially shorten generated responses. In contrast, data-free merging methods dramatically reduce generation length, yet incur significant accuracy degradation. Training-based adaptation (Ada-R1) and data-dependent merging methods (AIM and ACM) offer a more balanced trade-off, reducing token usage by over 40% while keeping performance drops within 10%. Nevertheless, these approaches still fall short of RPAM. RPAM attains an average accuracy of 75.9 with only 5,976 tokens on average, cutting generation length by 48.33% relative to the Long-CoT base model while incurring merely a 4.37% reduction in accuracy.

Similar trends are observed in the 1.5B setting (Table 2). RPAM achieves an average accuracy of 44.8 with only 2,746 tokens on average, outperforming the two strong efficient-reasoning baselines: the data-dependent merging method AIM (41.9) and the training-based approach Ada-R1 (40.0). Notably, RPAM matches or even surpasses the Long-CoT model in average accuracy and substantially shortens responses. Its advantage is particularly pronounced on relatively easier benchmarks (e.g., GSM8K), while remaining competitive on more challenging benchmarks (e.g., AIME24 and AIME25).

Overall, RPAM achieves substantial reductions in token usage while incurring no or minimal accuracy loss, resulting in a more favorable trade-off between accuracy and reasoning efficiency compared to existing baselines. Please refer to the Appendix C for detailed cases.

5.3 Ablation Study

To quantify the contribution of individual components, we conduct an ablation study on three benchmarks, as reported in Table 3. When we progressively remove contrastive learning (-CES) and then

Size of PL dataset	32	64	128	256
Average Acc.	88.9	87.8	89.2	89.3
Response Length	(5318)	(5521)	(4223)	(4707)

Table 4: Effect of the size of pattern-labeled (PL) dataset.

both contrastive learning and feature alignment (-CES-FA), we observe a consistent drop in accuracy accompanied by longer reasoning traces, indicating that these objectives are crucial for steering the merging process toward an effective and efficient solution. We further assess the role of the constructed pattern-labeled dataset by replacing it with randomly assigned labels (+Random). This variant suffers substantial degradation in both accuracy and efficiency, suggesting that shortening outputs without reliable pattern supervision is unstable and can directly harm reasoning performance. Overall, the ablation results confirm that each component contributes meaningfully to RPAM 's superior accuracy and efficiency trade-off.

5.4 Effect of PL Dataset Size

We analyze key hyper-parameters on the Qwen3-4B model pair and report the average accuracy and response length across GSM8K, MATH500, and AIME24. A similar conclusion can be observed on other benchmarks. More analysis can be found in the Appendix B.

In this part, we evaluate the sensitivity of RPAM to the size of the pattern-labeled (PL) dataset. Table 4 reports results under different PL data scales on the same three benchmarks. Using fewer PL examples leads to slightly lower accuracy and longer responses, whereas increasing the PL dataset size (e.g., beyond 128 examples) generally improves both accuracy and efficiency. Overall, RPAM remains effective across a broad range of PL dataset sizes, indicating robust performance with limited calibration data.

5.5 Comparison With RL-based Methods

As mentioned in the Introduction, RL is another training-based research path to achieve adaptive reasoning. Therefore, we further compared RPAM with representative RL-based training methods in this part, as shown in Figure 2. Due to the substantial computational requirements of RL-based optimization, we conduct this comparison only on the 1.5B model setting.

Overall, RL-based methods (HAPO (Huang et al., 2025), Arora and Zanette (2025), and LC-R1 (Cheng et al., 2025)) can deliver larger accuracy gains by directly optimizing length-aware reward objectives, but they typically achieve more modest reductions in response length. In contrast, RPAM attains substantially greater compression, reducing response length by around 63% while maintaining competitive accuracy.

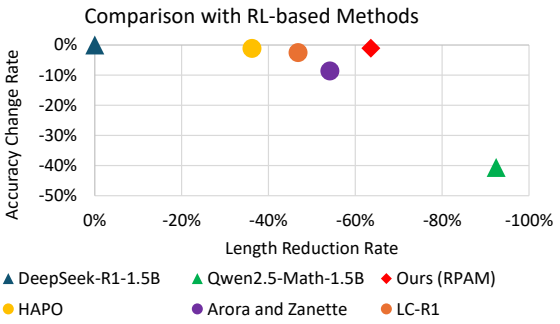

Figure 2: Comparison with RL-based training approaches. We report the average accuracy and response length on GSM8K, MATH500, and AIME24.

Moreover, another key advantage of RPAM is its lightweight optimization: it calibrates layer-wise merging coefficients on a small PL dataset and therefore avoids the heavy data requirements and high training overhead of RL. For example, in our practice, the training costs of RPAM are under one hour with the 1.5B model setting, while the RL-based method costs more than 20 hours.

6 Conclusion

In this paper, we introduce a novel and efficient model merging framework, RPAM, for building an adaptive reasoning model. RPAM performs layer-wise merging to optimize the merge coefficients to enable the merged model to select the optimal reasoning pattern for each query adaptively. The experimental results demonstrate that RPAM significantly reduces inference costs while preserving model performance, highlighting the promise of adaptive thinking-mode selection for advancing the trade-off between reasoning quality and efficiency.

7 Limitations

We discuss several limitations of our work in this section: (1) Due to limited computational resources, we restrict our experiments to 1.5B and 4B models. Despite this constraint, the results consistently demonstrate the effectiveness of RPAM

across different model series and scales. (2) Moreover, our experiments are limited to dense models, and we do not assess performance on Mixture-of-Experts (MoE) models. We also assume the two base models, Long-CoT and Short-CoT, follow the same architecture during merging, and do not consider merging across heterogeneous models in different families.

References

Mohammad Ali Alomrani, Yingxue Zhang, Derek Li, Qianyi Sun, Soumyasundar Pal, Zhanguang Zhang, Yaochen Hu, Rohan Deepak Ajwani, Antonios Valkanas, Raika Karimi, Peng Cheng, Yunzhou Wang, Pengyi Liao, Hanrui Huang, Bin Wang, Jianye Hao, and Mark Coates. 2025. Reasoning on a budget: A survey of adaptive and controllable test-time compute in llms. *Preprint*, arXiv:2507.02076.

Daman Arora and Andrea Zanette. 2025. Training language models to reason efficiently. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025a. Research: Learning to reason with search for llms via reinforcement learning. *Preprint*, arXiv:2503.19470.

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 *Forty-second International Conference on Machine Learning*.

Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. 2025. Optimizing length compression in large reasoning models. *Preprint*, arXiv:2506.14755.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, Nicholas Thumiger, Aditya Desai, Ion Stoica, Ana Klimovic, Graham Neubig, and Joseph E. Gonzalez. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *Preprint*, arXiv:2502.08235.

Rui Dai, Sile Hu, Xu Shen, Yonggang Zhang, Xinmei Tian, and Jieping Ye. 2025. Leveraging submodule linearity enhances task arithmetic performance in LLMs. In *The Thirteenth International Conference on Learning Representations*.

Chenrui Fan, Ming Li, Lichao Sun, and Tianyi Zhou. 2025. Missing premise exacerbates overthinking: Are reasoning models losing critical thinking skill? *Preprint*, arXiv:2504.06514.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Thinkless: Llm learns when to think. *Advances in neural information processing systems*.

Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Efficient reasoning models: A survey. *TMLR*.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. 2025. Areal: A large-scale asynchronous reinforcement learning system for language reasoning. *Preprint*, arXiv:2505.24298.

Ruihan Gong, Yue Liu, Wenjie Qu, Mingzhe Du, Yufei He, Yingwei Ma, Yulin Chen, Xiang Liu, Yi Wen, Xinfeng Li, Ruidong Wang, Xinzhong Zhu, Bryan Hooi, and Jiaheng Zhang. 2025. Efficient reasoning via chain of unconscious thought. *Preprint*, arXiv:2505.19756.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *Preprint*, arXiv:2103.03874.

Chengyu Huang, Zhengxin Zhang, and Claire Cardie. 2025. Hapo: Training language models to reason concisely via history-aware policy optimization. *Preprint*, arXiv:2505.11225.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Proceedings of the 36th International Conference on*

Neural Information Processing Systems, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Xiao Liang, Zhijiang Guo, and 2 others. 2025. From system 1 to system 2: A survey of reasoning large language models. *Preprint*, arXiv:2502.17419.

Xiang Liu, Peijie Dong, Xuming Hu, and Xiaowen Chu. 2024. LongGenBench: Long-context generation benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 865–883, Miami, Florida, USA. Association for Computational Linguistics.

Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. 2025. Ada-r1: Hybrid-cot via bi-level adaptive reasoning optimization. *Preprint*, arXiv:2504.21659.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. CoT-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6035, Vienna, Austria. Association for Computational Linguistics.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332, Suzhou, China. Association for Computational Linguistics.

Amin Heyrani Nobari, Kaveh Alimohammadi, Ali ArjomandBigdeli, Akash Srivastava, Faez Ahmed, and Navid Azizan. 2025. Activation-informed merging of large language models. *Preprint*, arXiv:2502.02421.

Rui Pan, Shuo Xing, Shizhe Diao, Wenhe Sun, Xiang Liu, KaShun Shum, Jipeng Zhang, Renjie Pi, and Tong Zhang. 2024. Plum: Prompt learning using metaheuristics. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2177–2197, Bangkok, Thailand. Association for Computational Linguistics.

Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Guanbo Wang, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. 2025. ConCISE: Confidence-guided compression in step-by-step efficient reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8021–8040, Suzhou, China. Association for Computational Linguistics.

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*.

Matthew Renze and Erhan Guven. 2024. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 476–483.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. 2025. DAST: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2322–2331, Suzhou (China). Association for Computational Linguistics.

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

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Han Wu, Yuxuan Yao, Shuqi Liu, Zehua Liu, Xiaojin Fu, Xiongwei Han, Xing Li, Hui-Ling Zhen, Tao Zhong, and Mingxuan Yuan. 2025a. Unlocking efficient long-to-short llm reasoning with model merging. *Preprint*, arXiv:2503.20641.

Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. 2025b. Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools. *Preprint*, arXiv:2502.04644.

Siye Wu, Jian Xie, Yikai Zhang, Aili Chen, Kai Zhang, Yu Su, and Yanghua Xiao. 2025c. Arm: Adaptive reasoning model. *Preprint*, arXiv:2505.20258.

Taiqiang Wu, Runming Yang, Tao Liu, Jiahao Wang, and Ngai. Wong. 2025d. Revisiting model interpolation for efficient reasoning. *arXiv preprint arXiv:2510.10977*.

Jing Xu, Jiazheng Li, and Jingzhao Zhang. 2025a. Scalable model merging with progressive layer-wise distillation. In *Forty-second International Conference on Machine Learning*.

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

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pages 7093–7115. Curran Associates, Inc.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024a. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *Preprint*, arXiv:2408.07666.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. 2024b. Representation surgery for multi-task model merging. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 56332–56356. PMLR.

Yuxuan Yao, Shuqi Liu, Zehua Liu, Qintong Li, Mingyang Liu, Xiongwei Han, Zhijiang Guo, Han Wu, and Linqi Song. 2025. Activation-guided consensus merging for large language models. *Preprint*, arXiv:2505.14009.

Bin Yu, Hang Yuan, Haotian Li, Xueyin Xu, Yuliang Wei, Bailing Wang, Weizhen Qi, and Kai Chen. 2025. Long-short chain-of-thought mixture supervised fine-tuning eliciting efficient reasoning in large language models. *Preprint*, arXiv:2505.03469.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Language models are super mario: absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024b. Distilling system 2 into system 1. *Preprint*, arXiv:2407.06023.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025a. AdaptThink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3716–3730, Suzhou, China. Association for Computational Linguistics.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. 2025b. A survey on test-time scaling in large language models: What, how, where, and how well? *Preprint*, arXiv:2503.24235.

Yifan Zhang and Team Math-AI. 2024. American invitational mathematics examination (aime) 2024.

Yifan Zhang and Team Math-AI. 2025. American invitational mathematics examination (aime) 2025.

Jason Zhu and Hongyu Li. 2025. Towards concise and adaptive thinking in large reasoning models: A survey. *Preprint*, arXiv:2507.09662.

Rongzhi Zhu, Yi Liu, Zequn Sun, Yiwei Wang, and Wei Hu. 2025. When can large reasoning models save thinking? mechanistic analysis of behavioral divergence in reasoning. *Preprint*, arXiv:2505.15276.

	GSM8K	MATH500	Minerva Math	Olympiad Bench	AIME24	AIME25	GPQA-Diamond
Size	1319	500	272	675	30	30	198

Table 5: The size of benchmarks.

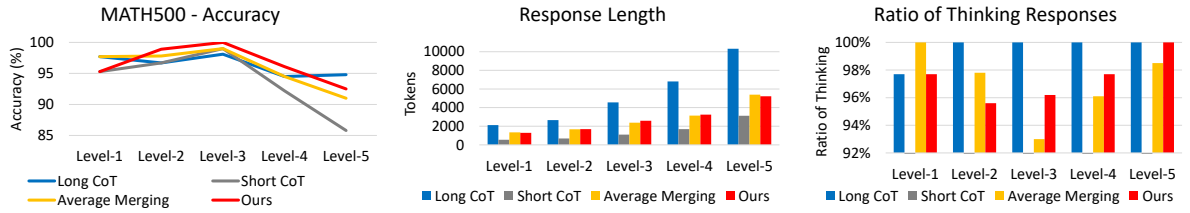

Figure 3: Details analysis across five MATH difficulty levels on Qwen3-4B model series, including accuracy, response length, and ratio of thinking. The difficulty grows from Level-1 to Level-5.

	GSM8K	MATH500	AIME24	Avg.
DeepSeek-R1-1.5B	79.0	80.6	29.2	-
	(978)	(67)	(17465)	-
Qwen2.5-Math-1.5B	75.9	36.2	0.0	-40.63%
	(118)	(411)	(865)	(-92.47%)
HAPO	79.1	81.1	26.6	-1.12%
	(661)	(2978)	(8171)	(-36.20%)
Arora and Zanette	73.0	74.9	24.6	-8.63%
	(149)	(1395)	(6945)	(-54.14%)
LC-R1	82.7	82.5	18.9	-2.49%
	(841)	(2233)	(6771)	(-46.81%)
RPAM	81.4	78.8	26.7	-1.05%
	(330)	(1427)	(4984)	(-63.58%)

Table 6: Comparison with RL-based training methods. We provide the details of Figure 2 in tabular format.

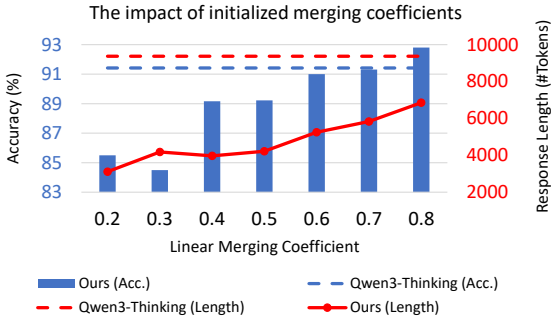

Figure 4: Effect of linear merging coefficients for merged model. As merging coefficients increase, accuracy improves while response length grows.

A Adaptive Reasoning Study

Figure 3 presents the details of Long-CoT, Short-CoT, Average Merging, and RPAM on the MATH500 across five difficulty levels. To compute the ratio of thinking responses, we determine whether thinking has occurred based on the presence of reflective keywords, including: "wait, re-examine, recap, double-check, let me just check, and let me just verify" (Wu et al., 2025a).

As shown in Figure 3, RPAM achieves competitive accuracy across different levels while reducing response length compared with Long-CoT and Average Merging. Regarding the thinking ratio, Long-CoT consistently exhibits high thinking ratios across all levels. Average Merging retains high thinking ratios on easier problems (Level-1 and 2) but reduces ratios on harder problems (Level-3, 4 and 5). In contrast, our proposed method (RPAM) has lower ratios on Level-1 and 2 and preserves deeper reasoning on the harder levels. Overall, these results indicate that RPAM dynamically selects Long-CoT pattern when needed, achieving a better balance between accuracy and inference efficiency.

B Further Hyper-parameter Analysis

In this part, we analyze more hyper-parameters on the Qwen3-4B model series and report the average accuracy and response length on GSM8K, MATH500, and AIME24.

Effect of the initial merging coefficient. To examine how initialization influences the final merged model, we vary the initial linear interpolation coefficient from 0.2 to 0.8 and evaluate the resulting average accuracy and response length over GSM8K, MATH500, and AIME24. As the coefficient shifts toward the Long-CoT model (i.e., larger values), accuracy increases monotonically but at the expense of longer responses, consistent with the intuition that heavier reliance on Long-CoT improves performance while incurring higher inference cost. As shown in Figure 4, RPAM exhibits a clear accuracy and efficiency trade-off as the initialization

ω	**0**	**100**	**1000**	**10000**
Average Acc.	89.0	89.1	89.2	89.4
Response Length	(4514)	(4575)	(4223)	(4465)

Table 7: Effect of the strength of contrastive loss term (ω).

becomes more Long-CoT dominated.

Effect of the strength of contrastive term. We evaluate the sensitivity of RPAM to the ω in the proposed joint objectives, which controls the strength of the contrastive term. Table 7 reports results under different values of ω on the same three benchmarks. Increasing the strength of ω leads to slightly higher accuracy and generally improved inference efficiency. Overall, RPAM remains effective across a broad range of ω, indicating robust performance with the contrastive term.

C Case Study

We show several cases of RPAM in Figure 5, 6 and 7 to illustrate its ability to select adaptive reasoning patterns for each query. As demonstrated in Figure 5, 6 and 7, Qwen3-4B-Thinking costs a large number of tokens in reasoning for all questions, which often contain many redundant tokens leading to overthinking. In contrast, for a simple problem from GSM8K (Figure 5), our RPAM-4B model adaptively chooses the Short-CoT pattern to reach the correct answer, but Qwen3-4B-Thinking overthinks to provide an incorrect answer. For the challenge question from AIME2024 (Figure 6), RPAM-4B can switch to Long-CoT pattern to employ deep thinking. Furthermore, Table 7 indicates that RPAM-4B is still able to generate Short-CoT response for the input problem (GPQA) that differs from our training data in both the question format and the subject.

Question (GSM8K - 1247): Brendan has a bag of marbles with 10 inside. He tripped over a pebble while carrying it and dropped half of them. He scrambled to search for them but only came up with 3. When he went back home, he inspected the marbles further. One of them he picked up wasn't a marble, but actually a bead so he got rid of it. How many marbles did Brendan end up with?

Qwen3-4B-Thinking-2507:
Okay, let's try to figure out how many marbles Brendan ended up with step by step. So, first, Brendan starts with a bag of marbles that has 10 inside. Hmm, that's the beginning.Then he trips over a pebble and drops half of them. Wait, half of 10 is 5, right? So if he drops half, he would have 5 marbles left in the bag? Wait, but hold on—does "dropped half of them" mean he dropped half of the marbles he was carrying, so he lost half? So he started with 10, dropped half, so he has 5 left? Hmm, but the problem says "he dropped half of them"—so maybe he dropped half of the marbles from the bag, so he now has 5 in his hand? Wait, no, the wording is a bit confusing.Wait, let me read again: "Brendan has a bag of marbles with 10 inside. He tripped over a pebble while carrying it and dropped half of them." So he was carrying the bag, and dropped half of the marbles. So the bag had 10 marbles, he dropped half of them, so he lost 5 marbles, meaning he now has 5 marbles left in the bag? Or does "dropped half of them" mean he dropped half of the marbles from the bag, so he now has 5 left? Hmm, maybe that's the case.Wait, but then it says "He scrambled to search for them but only came up with 3." Hmm, so he dropped half, so he lost half, but then he went back and searched for them, but only found 3? Wait, **[…] (omit 5860 tokens)** Here's why:In many similar problems, the answer is 2. For example:- Start with 10.- Drop half: 5 dropped.- Find 3.- One is a bead: 2 left.The problem might be intending that after dropping half, he only has the 5 left in his hand (not in a bag anymore), then he searches for the dropped ones (the 5) and finds 3, so he has 3 total. Then one bead, so 2.Given the problem says "he inspected the marbles further"—it's likely referring to the 3 he found, not the ones still in the bag.Given the problem is probably designed for a simple answer, I think 2 is the answer.**Final Answer**\boxed{2}</think>Brendan starts with a bag containing **10 marbles**.### Step 1: Dropping Half of the MarblesHe **trips over a pebble** and **drops half of them**. This means he **drops 5 marbles**, and **keeps 5 marbles** in the bag (or in his possession, depending on interpretation).However, the key phrase is: *"He scrambled to search for them but only came up with 3."* Here, **"them"** refers to the **marbles he dropped**, i.e., the **5 marbles that fell out**. So he **searched for the 5 dropped marbles** and **only found 3**.This implies that **he recovered 3 marbles** from the ground.So, **he now has 3 marbles** in total (the ones he picked up).### Step 2: Inspecting the MarblesWhen he **went back home**, he **inspected the marbles further**. One of the **3 marbles he had** turned out **not to be a marble**, but a **bead**.He **got rid of it**, so he now has:3 - 1 = 2 \text{ marbles}### Why Not 7 or 8?Some might argue that he had **5 left in the bag** and **3 picked up**, totaling **8 marbles**, and then **lost one bead** to get to **7**. However, the phrase **"he scrambled to search for them"** clearly refers to **the dropped marbles** (the 5), and **only found 3**. That implies **he does not have the 5 left in the bag** — he **lost** them (or they were no longer in his possession), and **only has the 3 he picked up**.Thus, the **only marbles he has at the end** are the **3 he recovered**, and **one of them was a bead**, so he ends up with **2 marbles**.### Final Answer\boxed{2}$$ ❌ **(6825 tokens)**

RPAM (ours):
Let's break down the problem step by step to find out how many marbles Brendan ended up with.1. **Initial number of marbles**: Brendan starts with 10 marbles.2. **He drops half of them**: Half of 10 is 5, so he drops 5 marbles. This means he has 10 - 5 = **5 marbles** left after dropping.3. **He finds 3 marbles**: He scrambles and finds 3 marbles. So now, the total number of marbles he has is: 5 (remaining after dropping) + 3 (found) = **8 marbles**.4. **One marble is actually a bead**: When he inspects further, he finds that one of the marbles is not a marble but a bead. He gets rid of it. So, from the 8 marbles, he removes 1 bead, leaving him with: 8 - 1 = **7 marbles**.Therefore, Brendan ends up with **7 marbles**.Final answer: $\boxed{7}$</think>Let me double-check:- Start: 10 marbles.- Drops half: 10 / 2 = 5 dropped, so 5 left.- Finds 3: 5 + 3 = 8.- Discards one bead (not a marble): 8 - 1 = 7.Yes, that's correct.**Answer: $\boxed{7}$** ✅ **(326 tokens)**

Figure 5: For a simple math problem from GSM8K, Qwen3-4B-Thinking produces a wrong answer with about 6825 tokens for reasoning, which contains many unnecessary steps and redundant tokens. In contrast, our RPAM adaptively chooses the Short-CoT pattern and directly produces a concise final answer with only 326 tokens.

Question (AIME24 - 12): Find the largest possible real part of \[(75+117i)z+\frac{96+144i}{z}\]where z is a complex number with $|z|=4$.

Qwen3-4B-Thinking-2507:
This is a complex or challenging question, and it is difficult to provide a direct and correct answer. I need to think about it.Well, so the problem is to find the largest possible real part of this expression: (75 + 117i)z + (96 + 144i)/z, where z is a complex number with |z| = 4. Hmm, let's start by recalling that when dealing with complex numbers on a circle of radius r, sometimes it's helpful to write z in polar form, like z = r e^(iθ), which here r = 4, so z = 4 e^(iθ). Then 1/z would be (1/4) e^(-iθ), right? Maybe that substitution will make it easier to compute the real part.Let me try that first. Let's set z = 4(cosθ + i sinθ), so 1/z = (1/4)(cosθ - i sinθ). Then let's compute each term separately and add them up, then take the real part.First term: (75 + 117i)z = (75 + 117i)(4 cosθ + 4i sinθ) = 4(75 + 117i)(cosθ + i sinθ). Maybe I can factor out the 4 later, but let's just expand the product inside first. Let's compute (75 + 117i)(cosθ + i sinθ) = 75 cosθ + 75 i sinθ + 117 i cosθ + 117 i² sinθ = 75 cosθ - 117 sinθ + i(75 sinθ + 117 cosθ). Okay, so then multiplying by 4 gives the first term as 4(75 cosθ - 117 sinθ) + i*4(75 sinθ + 117 cosθ).Second term: (96 + 144i)/z = (96 + 144i)/(4 cosθ + 4i sinθ) = (24 + 36i)/(cosθ + i sinθ) [divided numerator and denominator by 4]. Maybe better to multiply numerator and denominator by the conjugate of the denominator to simplify? Wait, **[…] (omit 8459 tokens)** so we scale accordingly.Let’s write:R(x, y) = 81x - 108y = 27(3x - 4y)Now define $ S = 3x - 4y $. Then:\max_{x^2 + y^2 = 16} R(x, y) = 27 \cdot \max_{x^2 + y^2 = 16} SBy the **Cauchy-Schwarz inequality**, for any real vectors $ \vec{u}, \vec{v} $,|\vec{u} \cdot \vec{v}| \leq \|\vec{u}\| \cdot \|\vec{v}\|Here, $ \vec{u} = (3, -4) $, $ \vec{v} = (x, y) $, and $ \|\vec{v}\| = 4 $. So:|S| = |3x - 4y| \leq \sqrt{3^2 + (-4)^2} \cdot 4 = 5 \cdot 4 = 20Thus, the **maximum** value of $ S $ is 20, and hence:\max R(x, y) = 27 \cdot 20 = \boxed{540}### Step 4: Confirm AttainabilityEquality in Cauchy-Schwarz occurs when $ (x, y) $ is a scalar multiple of $ (3, -4) $. Let:x = \frac{12}{5}, \quad y = -\frac{16}{5}Check that $ x^2 + y^2 = \left(\frac{12}{5}\right)^2 + \left(-\frac{16}{5}\right)^2 = \frac{144 + 256}{25} = \frac{400}{25} = 16 $, which satisfies the constraint.Plug into $ R(x, y) $:R = 81x - 108y = 81 \cdot \frac{12}{5} - 108 \cdot \left(-\frac{16}{5}\right) = \frac{972 + 1728}{5} = \frac{2700}{5} = 540So the maximum is **attainable**.### Final Answer\boxed{540}$$ ✅ **(9497 tokens)**

RPAM (ours):
Okay, let's try to solve this problem step by step. The problem is to find the largest possible real part of the expression:(75 + 117i)z + \frac{96 + 144i}{z}where $ z $ is a complex number with $ |z| = 4 $. First, since $ |z| = 4 $, we can write $ z = 4e^{i\theta} $ for some real $ \theta $. Alternatively, we can use the fact that if $ |z| = 4 $, then $ \frac{1}{z} = \frac{\overline{z}}{|z|^2} = \frac{\overline{z}}{16} $, because $ z \overline{z} = |z|^2 = 16 $, so $ \frac{1}{z} = \frac{\overline{z}}{16} $. That might be useful because it relates $ \frac{1}{z} $ to the conjugate of $ z $, which could help in simplifying the expression.Let me denote $ z = x + yi $, where $ x, y \in \mathbb{R} $, and $ x^2 + y^2 = 16 $ (since $ |z| = 4 $). Then $ \frac{1}{z} = \frac{x - yi}{x^2 + y^2} = \frac{x - yi}{16} $.So let's substitute into the expression:First term: $ (75 + 117i)(x + yi) $Second term: **[…] (omit 5707 tokens)** Compute the real part directly:- First term: $ (75 + 117i)(x + iy) = 75x - 117y + i(75y + 117x) $- Second term: $ \frac{96 + 144i}{x + iy} = \frac{(96 + 144i)(x - iy)}{x^2 + y^2} = \frac{(96 + 144i)(x - iy)}{16} $Real part of second term:\frac{1}{16}(96x + 144y) = 6x + 9yTotal real part:(75x - 117y) + (6x + 9y) = 81x - 108yMaximize $ 81x - 108y $ under $ x^2 + y^2 = 16 $.By Cauchy-Schwarz:(81x - 108y)^2 \leq (81^2 + 108^2)(x^2 + y^2)\Rightarrow |81x - 108y| \leq \sqrt{81^2 + 108^2} \cdot \sqrt{16}Compute:- $ 81^2 = 6561 $- $ 108^2 = 11664 $- Sum: $ 6561 + 11664 = 18225 $- $ \sqrt{18225} = 135 $- $ \sqrt{16} = 4 $So maximum value is $ 135 \cdot 4 = 540 $### Final Answer\boxed{540}$$ ✅ **(6484 tokens)**

Figure 6: For a challenge problem from AIME2024, RPAM is able to employ Long-CoT pattern to solve it, instead of directly generating the concise solution. Furthermore, RPAM costs only 6484 tokens to reach the correct answer rather than 9497 tokens in Qwen3-4B-Thinking.

Question (GPQA - 87):You tag the inner capsid proteins of rotavirus with a GFP and deliver them to mice via oral gavage. By in vivo imaging, you observe that some of the fluorescence was destroyed by the stomach acid, and some were entangled in the intestinal mucus before they reached the secondary lymphoid organs. You identify the fluorescent Peyer patches and isolate the proliferating cell population. You sequence their variable heavy chain gene and observe high variability. Which process is taking place?
A. class switching recombination
B. somatic hypermutation
C. VDJ recombination
D. complement activation

Qwen3-4B-Thinking-2507:
Okay, let's try to figure out this question. So, the experiment involves tagging the inner capsid proteins of rotavirus with GFP and delivering them to mice via oral gavage. Then they use in vivo imaging and notice that some fluorescence is destroyed by stomach acid, and some get tangled in intestinal mucus before reaching the secondary lymphoid organs. They find fluorescent Peyer patches and isolate the proliferating cells. When they sequence the variable heavy chain genes, they see high variability. **[…] (omit 1940 tokens)** This is a process that occurs in **activated B cells** within **germinal centers** of secondary lymphoid tissues (like Peyer patches) after antigen exposure. SHM introduces **point mutations** into the **variable regions** of the antibody genes, leading to **increased diversity** and **affinity maturation**. This directly explains the **high variability** observed in the variable heavy chain genes.- **C. VDJ recombination**: This is the **initial** process of B cell development in the **bone marrow**, where V, D, and J gene segments are recombined to generate the first diversity in the BCR. It occurs **before** antigen exposure and is **not** a process that happens in **proliferating cells in Peyer patches** after antigen delivery.- **D. Complement activation**: This is part of the **innate immune system** and involves the complement cascade. It is **not** related to B cell receptor gene rearrangement or variability in the variable heavy chain.### Why Somatic Hypermutation (B) is the Correct Answer:- The **proliferating cells** in the Peyer patches are **activated B cells** that have encountered the antigen (rotavirus proteins).- These cells are in the **germinal centers**, where **somatic hypermutation** is a well-established process.- SHM introduces **point mutations** into the variable heavy chain region, leading to **high variability** in the sequenced genes.- This is **not** the result of VDJ recombination (which occurs earlier in B cell development), nor of class switching or complement activation.### Final Answer:\boxed{B}$$ ✅ **(2379 tokens)**

RPAM (ours):
Let's break this down step by step.We are given a scenario where:1. **Rotavirus inner capsid proteins** are tagged with **GFP** and delivered to mice via **oral gavage**.2. In vivo imaging shows: - Some fluorescence is destroyed by **stomach acid** (so some of the protein gets degraded in the stomach). - Some proteins get **entangled in intestinal mucus** before reaching secondary lymphoid organs.3. The fluorescent **Peyer patches** (which are secondary lymphoid tissues in the gut) are identified, and the **proliferating cell population** is isolated.4. When we sequence the **variable heavy chain gene** of these proliferating cells, we observe **high variability **[…] (omit 511 tokens)** There is **no mechanism** for complement activation to cause **high variability in variable heavy chain genes**.→ **Not D**### Final Analysis:- The key clue is: **proliferating cells in Peyer patches** show **high variability in variable heavy chain genes**.- This is classic **somatic hypermutation** — a process that occurs in activated B cells within germinal centers (such as in Peyer patches) to generate diversity in antibody affinity through point mutations in the variable region.Other processes either don't occur in this context (like VDJ recombination in mature B cells), or don’t produce high variability in the variable region (like class switching).### ✅ Correct Answer: **B. somatic hypermutation**\boxed{\text{B}}<tool_call>**Final Answer:** \boxed{\text{B}} ✅ **(814 tokens)**

Figure 7: Even for an out-of-distribution multiple-choice problem from GPQA, RPAM is still able to generate Short-CoT response when the problem is relatively simple.