AdaReason: Progressive Training of Multi-LoRA Adapters for Budget-Adaptive Language Reasoning Models

Jiacheng Wang[2*], Tianle Chen[2*], Pengyu Cheng[2*], Xiaofeng Hou[3], Jiacheng Liu [1†]

[1]Hong Kong University of Science and Technology, Hong Kong, China
[2]Xi'an Jiao Tong University, Xi'an, China
[3]Shanghai Jiao Tong University, Shanghai, China
{jiacheng,tianlechen1030,pengyucheng0423}@stu.xjtu.edu.cn,xfhelen@sjtu.edu.cn,jiachengliu@ust.hk

Abstract

Large reasoning models (LRMs) have demonstrated remarkable capabilities in solving complex problems through extended chain-of-thought reasoning. However, existing approaches face a fundamental trade-off between computational efficiency and reasoning accuracy. Current methods either lack support for user-specified computational budgets or require maintaining multiple independent models, leading to significant resource overhead. In this paper, we present AdaReason, a unified framework that trains a single base model to support arbitrary user-defined computational budgets through dynamic adapter composition. Our approach introduces three key innovations: (1) a length-adaptive step reward function that stabilizes training across diverse budget constraints, (2) a progressive training strategy that gradually tightens computational bounds while maintaining model performance, and (3) a runtime adapter merging mechanism that dynamically interpolates between different computational preferences. Unlike existing methods that suffer from training instability in large context windows, AdaReason achieves stable convergence through careful reward shaping and progressive constraint tightening. Additionally, we provide a rigorous theoretical analysis, establishing a performance bound for our merged model. Experiments on different reasoning benchmarks demonstrate that AdaReason establishes a new state-of-the-art in the performance-efficiency trade-off and enables flexible runtime budget adaptation.

1 Introduction

The landscape of artificial intelligence has been fundamentally transformed by the advent of large reasoning models (LRMs) that demonstrate unprecedented capabilities in complex problem-solving tasks (Plaat et al. 2024; Xu et al. 2025a). Recent breakthroughs, exemplified by OpenAI's o1 (OpenAI 2024) and DeepSeek-R1 (DeepSeek-AI 2025), have established a compelling paradigm: allowing models to engage in extended chain-of-thought (CoT) reasoning can yield dramatic improvements in performance across challenging domains ranging from mathematical problem-solving to scientific reasoning. These models generate comprehensive reasoning traces that can span tens of thousands of tokens, meticulously working through problems with a level of deliberation that mirrors human expert reasoning.

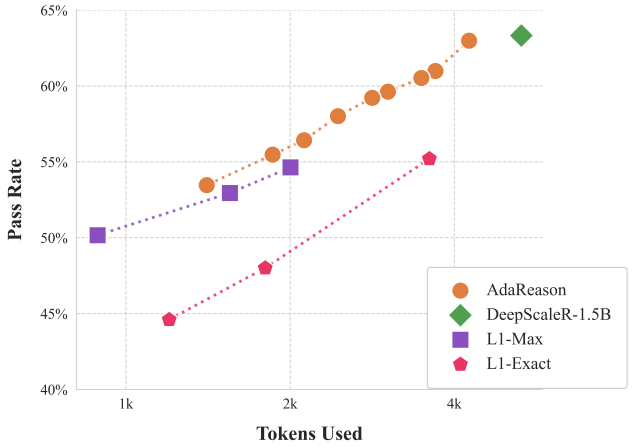

Figure 1: AdaReason achieves higher performance per token than state-of-the-art methods.

However, this remarkable capability introduces a critical challenge as these models transition to production systems. The computational overhead of extensive reasoning traces creates substantial deployment barriers, particularly in resource-constrained or real-time applications (Feng et al. 2025; Sui et al. 2025). This fundamental tension between reasoning quality and computational efficiency cannot be resolved through simple heuristics. The challenge is complicated by diverse real-world scenarios where users have varying computational budgets based on task complexity, available resources, latency requirements, and economic constraints (Li et al. 2025). While a financial trading system might require rapid responses with minimal reasoning overhead, a scientific research application could afford extensive computational resources for thorough analysis. Ideally, a single reasoning model should dynamically adapt its computational investment to match available budgets while optimizing performance within those constraints.

*These authors contributed equally.
†Corresponding author

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

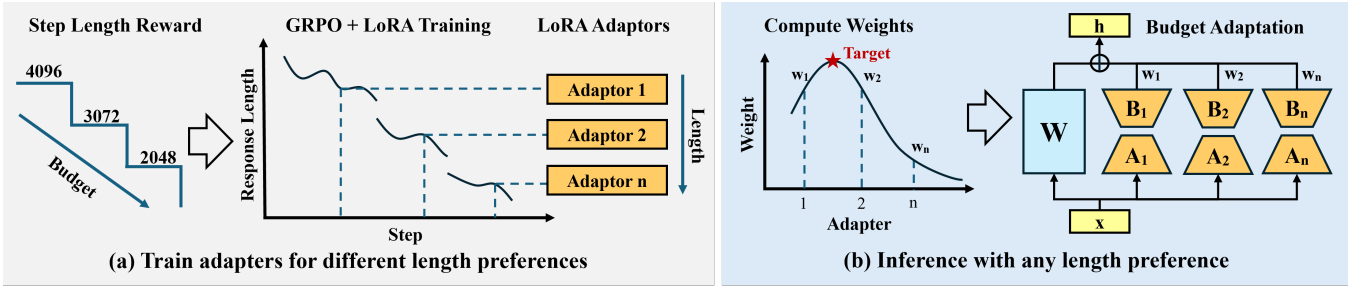

Figure 2: An overview of our proposed framework for controllable response length. (a) Training Phase: We use a progressive training strategy with a decreasing length budget to train a set of specialized LoRA adaptors, each an expert for a specific length preference. (b) Inference Phase: To generate a response of a desired target length, we dynamically compute weights for each adaptor and combine them to adapt the model's output on the fly.

Current approaches to address this efficiency-accuracy trade-off fall into two categories, each with inherent limitations. The first category employs direct truncation strategies (Muennighoff et al. 2025; Han et al. 2024a), which impose hard limits on reasoning length through external mechanisms. While computationally efficient, this approach frequently results in incomplete reasoning chains and significant performance degradation. The second category leverages preference optimization and reinforcement learning frameworks that incorporate length-based reward signals to modulate model behavior (Ma et al. 2025; Xia et al. 2025; Xu et al. 2025b). Traditional approaches within this paradigm require training multiple specialized models with distinct length preferences to accommodate diverse computational requirements, leading to prohibitive storage and deployment costs in practical applications. Recently, L1 (Aggarwal and Welleck 2025) proposed training unified models capable of internalizing user-specified computational constraints. However, these methods exhibit significant training instability when optimizing across diverse budget targets, as the sparse distribution of target lengths generates highly variable and often conflicting gradients, leading to convergence difficulties and inconsistent performance.

To address these limitations, we propose AdaReason, a novel framework that reconceptualizes the problem through a fundamentally different architectural approach. Rather than attempting to train a single monolithic model to handle all possible computational budgets or maintaining multiple completely independent models, our method leverages the power of parameter-efficient fine-tuning through parallel low-rank adapters (LoRA) (Hu et al. 2022). Each adapter specializes in a specific computational preference while sharing a common base model, creating a unified system that can efficiently serve diverse reasoning requirements.

Our approach is grounded in the key insight that *different reasoning lengths inherently require fundamentally different reasoning strategies and cognitive approaches*. Short reasoning demands highly focused, direct problem-solving techniques that quickly identify the most promising solution paths and execute them efficiently. In contrast, long reasoning benefits from extensive exploration of the problem space, comprehensive verification of intermediate steps, and thorough consideration of alternative approaches. By explicitly acknowledging and accommodating these different reasoning modalities through specialized adapters, our framework can optimize each approach independently while maintaining computational efficiency. The AdaReason framework introduces several technical innovations that collectively address the limitations of existing approaches. Our training methodology simultaneously optimizes multiple LoRA adapters with different computational preferences, enabling each adapter to develop specialized expertise while benefiting from shared representations in the base model. At inference time, we employ a novel adapter merging mechanism that dynamically interpolates between trained adapters based on user-specified computational budgets, enabling smooth adaptation to arbitrary constraints without requiring additional training. The main contributions of this work can be summarized as follows,

- We propose a multi-adapter framework, AdaReason, that decouples budget-specific reasoning strategies using parallel LoRA modules, eliminating the gradient conflicts inherent in monolithic models.
- We develop a stable progressive training algorithm incorporating length-adaptive rewards that mitigates training instability via curriculum learning by gradually tightening budget constraints.
- We introduce a zero-shot runtime budget adaptation mechanism enabling seamless adaptation to varying computational constraints.
- We provide rigorous theoretical analysis, providing a formal performance bound for the interpolated model, guaranteeing graceful performance degradation for intermediate budgets.
- We provide comprehensive empirical validation demonstrating superior performance compared to existing methods while maintaining similar token budgets.

2 Methodology

In this section, we propose AdaReason, a framework that provides fine-grained control over the computational budget of large reasoning models without compromising per-

formance or training stability. The overall architecture of our approach is illustrated in Figure 2.

2.1 Problem Formulation

Let $\mathcal{X}$ denote the space of input queries and $\mathcal{Y}$ the space of reasoning traces with final answers. Given a query $x \in \mathcal{X}$ and a computational budget $b \in \mathbb{N}$ (maximum tokens), our goal is to train a model π_θ that generates a response $y \sim \pi_\theta(\cdot|x, b)$ satisfying:

$$\max_\theta \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x,b)} \left[R(x, y) \cdot \mathbb{I}[L(y) \leq b]\right] \tag{1}$$

where $R(x, y)$ is the correctness reward, $L(y)$ denotes the length of response y, and $\mathbb{I}[\cdot]$ is the indicator function.

The key challenge is that optimal reasoning strategies vary significantly across budgets. Formally, let π_b^* denote the optimal policy for budget b. The reasoning patterns in π_{500}^* (concise, direct solutions) differ fundamentally from those in π_{5000}^* (exploratory, multi-path reasoning with verification). Existing approaches attempt to learn a single policy π_θ that approximates all π_b^* simultaneously, leading to:

$$\mathcal{L}_{\text{conflict}} = \sum_{b\in\mathcal{B}} \|\pi_\theta(\cdot|x, b) - \pi_b^*(\cdot|x)\|^2 \tag{2}$$

This multi-objective optimization suffers from conflicting gradients when π_b^* and $\pi_{b'}^*$ prescribe different actions for the same state, resulting in suboptimal compromises.

2.2 AdaReason Architecture Design

Our framework decomposes this multi-objective optimization into tractable subproblems using K specialized Low-Rank Adapters. Given a base large reasoning model with frozen parameters θ_{base}, we introduce adapter set $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_K\}$ where each $\mathcal{A}_k$ targets a specific budget range B_k. For each transformer layer's weight matrix $W \in \mathbb{R}^{d\times d}$, adapter $\mathcal{A}_k$ introduces low-rank updates:

$$W_k = W + \Delta W_k = W + W_{\text{down}}^{(k)} W_{\text{up}}^{(k)} \tag{3}$$

where $W_{\text{down}}^{(k)} \in \mathbb{R}^{d\times r}$, $W_{\text{up}}^{(k)} \in \mathbb{R}^{r\times d}$ with rank $r \ll d$. This design enables each adapter to develop specialized reasoning patterns while maintaining computational efficiency through parameter sharing in the base model.

Adapter Configuration Strategy: We adopt uniform sampling to ensure equitable distribution across the computational spectrum:

$$B_k = B_{\min} + (k - 1) \cdot \frac{B_{\max} - B_{\min}}{K - 1} \tag{4}$$

Our multi-LoRA design provides two fundamental advantages: (1) *Eliminates Gradient Conflicts*: Each adapter optimizes independently for its specific budget, avoiding the conflicting gradients that plague single-model approaches. (2) *Dramatic Storage Reduction*: Storage scales as $\mathcal{O}(K \cdot r \cdot d)$ with $r \ll d$, achieving $(1 - r/d) \approx 98\%$ reduction compared to separate models.

2.3 Progressive Training Strategy

To address training instability inherent in multi-budget optimization, we develop a progressive curriculum that gradually transitions from permissive to restrictive length constraints. This approach fundamentally differs from existing methods by providing stable convergence guarantees while maintaining training efficiency.

Step Reward for Length Control We design a simplified yet effective step reward mechanism that directly incentivizes length compliance. For adapter k at training step t, we define the reward as:

$$R_k(x, y, t) = \mathcal{R}_{\text{correct}}(x, y) \cdot S_k(y, t) \tag{5}$$

where the step reward $S_k(y, t)$ is defined as:

$$S_k(y, t) = \text{clip}(\lambda_k(t) \cdot (B_k(t) - L(y)) + 0.5, 0, 1) \tag{6}$$

This formulation provides positive reinforcement for correct, budget-compliant responses and implements a smooth reward transition within a band around the budget threshold. It creates a clear signal for length-controlled reasoning without complex penalty structures.

Progressive Parameter Schedule The key innovation lies in time-dependent parameters that implement curriculum learning. During training adapter k, the budget constraint and penalty weight evolve according to:

$$B_k(t) = B_k^{\text{tar}} + (B_k^{\text{init}} - B_k^{\text{tar}}) \cdot \max(0, f(1 - t/T)) \tag{7}$$

$$\lambda_k(t) = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) \cdot \min(1, f(t/T)) \tag{8}$$

where T represents total training steps, f is the function that transforms the linear training progress into a staged, piecewise-constant profile to stabilize training. The progressive budget schedule ensures that each adapter k starts training with a relaxed budget constraint B_k^{init} and gradually transitions to its target constraint B_k^{tar}. This curriculum learning approach allows adapters to first learn fundamental reasoning patterns before specializing in their specific budget requirements.

2.4 Runtime Budget Adaptation Mechanism

At inference time, given user budget B^*, we dynamically merge trained adapters to match the specified constraint without requiring additional training. This capability represents a significant advancement over existing methods that require separate models or retraining for new budgets.

Adaptive Merging Strategy We compute interpolation weights based on budget proximity using a temperature-controlled softmax:

$$w_k(B^*) = \frac{\exp\left(-\frac{|B^* - B_k^{\text{target}}|^2}{2\sigma^2}\right)}{\sum_{j=1}^K \exp\left(-\frac{|B^* - B_j^{\text{target}}|^2}{2\sigma^2}\right)} \tag{9}$$

where σ controls interpolation smoothness. The merged adapter parameters become:

$$\Delta\theta^*(B^*) = \sum_{k=1}^K w_k(B^*) \cdot \Delta\theta_k \tag{10}$$

Our merging strategy provides several advantages: (1) *Smooth Adaptation*: Continuous interpolation enables fine-grained budget control without discrete jumps in performance. (2) *Zero-Shot Generalization*: No additional training required for arbitrary budgets within the covered range. (3) *Computational Efficiency*: Runtime LoRA merging adds negligible overhead compared to maintaining separate models, and is supported by existing systems (Chen et al. 2024).

2.5 Theoretical Analysis

In this section, we provide a formal analysis of our adaptive multi-adapter merging strategy. Our analysis rests on two mild assumptions about the smoothness of the performance landscape with respect to the model's parameters and the relationship between adapter parameters and their target budgets. In addition, we introduce one assumption about the curvature of the performance landscape, which is standard in approximation theory.

Assumption 1 (Lipschitz Continuity of Performance)**.** *Let $\mathcal{J}(\Delta\theta)$ be the expected reward of the policy $\pi_{\theta_{base}+\Delta\theta}$. We assume the function $\mathcal{J}$ is $L_{\mathcal{J}}$-Lipschitz continuous with respect to the adapter parameters $\Delta\theta$ in a neighborhood around the trained adapters, i.e., for any two adapters $\Delta\theta_a, \Delta\theta_b$:*

$$|\mathcal{J}(\Delta\theta_a) - \mathcal{J}(\Delta\theta_b)| \leq L_{\mathcal{J}} \|\Delta\theta_a - \Delta\theta_b\|_F \tag{11}$$

where $\|\cdot\|_F$ is the Frobenius norm. This is a standard assumption for neural networks with smooth activation functions.

Assumption 2 (Budget-Parameter Proximity)**.** *We assume that the distance between the parameters of two optimally trained adapters is bounded by a function of the distance between their target budgets. Specifically, for any two adapters k and j:*

$$\|\Delta\theta_k - \Delta\theta_j\|_F \leq C_B |B_k^{target} - B_j^{target}| \tag{12}$$

for some constant C_B. This assumption posits that adapters trained for similar budgets will converge to similar parameterizations, which is empirically observed in practice.

Assumption 3 (Bounded Performance Curvature)**.** *Let $\mathcal{J}_{opt}(B)$ be the performance of the true optimal policy for a given budget B. We assume that the performance landscape is smooth and does not have infinitely sharp turns. Formally, we assume its second derivative with respect to the budget is bounded, i.e., $|\mathcal{J}''_{opt}(B)| \leq M$ for some constant M. This implies the performance cost of deviating from an optimal budget is locally quadratic.*

Our main theorem bounds the performance of the merged model.

Theorem 1 (Performance Bound of the Multi-Adapter Merged Model)**.** *Let $\{\Delta\theta_k\}_{k=1}^K$ be the set of adapters trained for budgets $\{B_k^{target}\}_{k=1}^K$. For an arbitrary target budget B^*, let $\Delta\theta^*(B^*) = \sum_{k=1}^K w_k(B^*)\Delta\theta_k$ be the interpolated adapter created by our multi-adapter merging strategy, where the weights $w_k(B^*)$ are given by the softmax function in Eq. (9). Let $k^* = \arg\min_k |B^* - B_k^{target}|$ be the index of the adapter with the closest budget anchor to B^*. The performance of the interpolated model, $\mathcal{J}(\Delta\theta^*(B^*))$, is bounded as follows:*

$$\mathcal{J}(\Delta\theta^*(B^*)) \geq \mathcal{J}(\Delta\theta_{k^*}) - L_{\mathcal{J}} C_B \sum_{k=1}^K w_k(B^*) |B_k^{target} - B_{k^*}^{target}| \tag{13}$$

This theorem can be proved by using the Lipschitz continuity of the loss function to relate performance to parameter distance, and then bounding this distance using the triangle inequality. Based on Theorem 1, we now present a new Theorem that analyzes the performance gap relative to the true optimum as a function of adapter density.

Theorem 2 (Performance Improvement with Adapter Density)**.** *Consider a set of K adapters trained for budgets $\{B_k^{target}\}$ uniformly distributed over a range $[B_{\min}, B_{\max}]$. The spacing between adapters is $\Delta B = \frac{B_{\max} - B_{\min}}{K-1}$. Under Assumptions 1-3, the gap between the performance of the true optimal policy for a budget B^* and the performance of our merged model is bounded by the square of the adapter spacing:*

$$\mathcal{J}_{opt}(B^*) - \mathcal{J}(\Delta\theta^*(B^*)) \leq C \cdot (\Delta B)^2 = \mathcal{O}\left(\frac{1}{K^2}\right) \tag{14}$$

where C is a constant independent of K.

This theorem can be proved by treating the merging as a function approximation problem. Theorem 2 provides a powerful theoretical justification for our multi-adapter strategy. It reveals a clear and direct relationship between the number of adapters and the quality of the final merged model. The $\mathcal{O}(1/K^2)$ result shows that the performance gap between our model and the theoretical optimum shrinks quadratically with the number of adapters. This provides a strong incentive for using a larger set of specialized adapters to cover the budget space more densely.

3 Experiment Results and Analysis

We conduct a comprehensive set of experiments to validate the effectiveness of AdaReason. Our evaluation is designed to answer the following key research questions:

(1) How does the performance-efficiency trade-off of AdaReason compare to strong baseline methods across various reasoning tasks?

(2) What are the contributions of the adaptive merging mechanism?

(3) Does AdaReason learn to adapt its reasoning behavior and strategy in response to different computational budget constraints?

3.1 Experimental Setup

Base Model and Datasets We follow the setup of L1 (Aggarwal and Welleck 2025) and use *DeepScaleR-1.5B-Preview* (Luo et al. 2025) as our base LRM. We fine-tune our models on the *DeepScaleR-Preview-Dataset*, a high-quality dataset of 40,000 mathematical problems sourced

	Accuracy			Generation Length		
	GPQA	H-Eval	Avg.	GPQA	H-Eval	Avg.
DeepScaleR-1.5B						
	36.4	86.6	61.5	5287.6	4807.1	5047.3
L1-Exact						
3600	32.8	79.9	56.3	2472.1	2172.1	2322.1
2048	30.8	79.9	55.3	1102.9	1903.9	1503.4
1024	30.3	78.7	54.5	823.5	1579.1	1201.3
L1-Max						
3600	32.3	82.3	57.3	1730.3	2009.8	1870.1
3072	32.8	79.3	56.0	1455.0	1777.3	1616.1
2048	30.8	82.9	56.9	950.0	1441.8	1195.9
AdaReason-Trained						
T_1	34.4	84.1	59.3	4357.1	4035.5	4196.3
T_2	34.8	85.4	60.1	3332.8	3103.2	3218.0
T_3	33.8	84.8	59.3	2917.2	2546.5	2731.8
T_4	33.2	84.1	58.7	2067.5	1938.1	2002.8
T_5	34.6	81.1	57.9	1301.2	1606.0	1453.6
AdaReason-Merge						
$M_{1.5}$	35.0	85.4	60.2	3687.7	3428.0	3557.8
$M_{2.5}$	35.7	83.5	59.6	3029.0	2708.3	2868.7
$M_{3.5}$	33.0	83.5	58.3	2353.6	2219.8	2286.7
$M_{4.5}$	34.0	82.9	58.5	1719.4	1832.7	1776.1

Table 1: Performance of AdaReason on non-mathematical datasets.

from AIME, AMC, Omni-MATH (Gao et al. 2024), and STILL (Min et al. 2024). For evaluation, we assess performance on a diverse set of benchmarks:

- **Mathematical Reasoning:** GSM8k (Cobbe et al. 2021), MATH500 (Hendrycks et al. 2021), AIME2025, AMC2023, and Olympiad Bench (He et al. 2024).
- **Non-mathematical Reasoning:** GPQA (Rein et al. 2024) and HumanEval (Chen et al. 2021).

All evaluations are performed in a zero-shot setting to test the models' intrinsic reasoning capabilities, with a maximum generation length of 16k tokens. On HumanEval we report `pass@4` for more stable results.

Baselines To comprehensively evaluate the performance of AdaReason, we benchmark it against a carefully curated set of baseline models.

1. **DeepScaleR-1.5B-Preview:** This is the original, frozen large reasoning model used as the foundation for our fine-tuning (Luo et al. 2025).
2. **L1-Max and L1-Exact:** A variant of DeepScaleR-1.5B-Preview fine-tuned to follow prompt-based length constraints (Aggarwal and Welleck 2025).

Implementation Details of AdaReason We train AdaReason using the verl framework with the GRPO algorithm (Sheng et al. 2025). We use a global batch size of 128 and a learning rate of 1e-5 with a linear warmup and decay schedule. For each prompt during RL training, we generate 8 rollouts to ensure sufficient exploration of the policy space.

Our training methodology employs a progressive length curriculum over 200 steps, during which the target budget is systematically reduced from an initial 4096 tokens to a final 2048 tokens. For AdaReason, we instantiate $K = 5$ parallel LoRA adapters. To efficiently generate a suite of budget-specialized adapters, we derive them from a single training run by saving model checkpoints at various points in the curriculum. This approach avoids the significant overhead of multiple independent training procedures. Consequently, each adapter becomes specialized for the budget constraint active at its corresponding training stage, capturing a distinct reasoning style from a single, unified training process.

Evaluation Protocol To rigorously evaluate the adaptive merging capability of AdaReason, we assess its performance on budget targets that lie *between* the anchor points of our specialized adapters. This setup directly tests the model's ability to interpolate its reasoning strategy and generalize to unseen computational constraints.

For clarity in our results, we adopt the following notation. The five specialized adapters derived from our progressive training curriculum are denoted as $\{T_1, \ldots, T_5\}$. The merged models, created by applying our adaptive merging mechanism to adjacent adapters T_i and T_{i+1}, are denoted as $M_{i.5}$. Collectively, we refer to these two sets of models as *AdaReason-Trained* and *AdaReason-Merge*, respectively. To ensure a fair comparison, the L1-Max baseline is evaluated against the same numerical budget targets, which are provided via its instruction-prompting mechanism.

3.2 Main Results

AdaReason establishes a new state-of-the-art in the performance-efficiency trade-off. Figure 3 illustrates that AdaReason consistently operates on a superior Pareto frontier compared to existing baselines on mathematical reasoning tasks. For any given computational budget, both our specialized adapters and interpolated models achieve higher accuracy, achieving an average accuracy improvement of 1% over L1-Max and 5% over L1-Exact on mathematical tasks under comparable length preferences. This advantage is a direct result of our multi-adapter framework, which mitigates the gradient conflicts that typically degrade performance in single-policy models.

Runtime merging enables precise and effective budget adaptation. Another key finding is the effectiveness of our runtime adaptation mechanism. As shown in Figure 3, our adaptive merging strategy allows for fine-grained control, as the merged models generate outputs with lengths that lie predictably between their constituent adapters. This emergent benefit empirically validates our theoretical analysis, suggesting that interpolating specialized reasoning strategies can yield more robust and generalized solutions.

AdaReason can generalize to non-mathematical domains. AdaReason's advantages are not confined to mathematical reasoning. As detailed in Table 1, the framework demonstrates strong performance and precise length control on non-mathematical reasoning benchmarks like GPQA and HumanEval. For example, our merged model $M_{4.5}$ achieves an average accuracy of 58.5%, substantially outperforming the L1-Max baseline (57.3% at a comparable budget). While the correlation between reasoning length and accuracy is

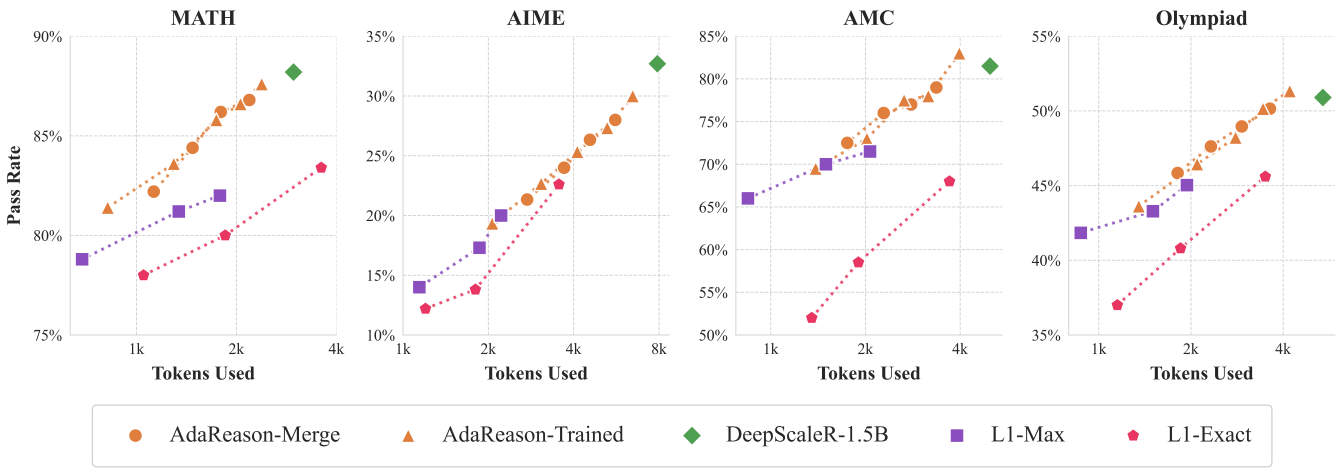

Figure 3: Performance of AdaReason versus generation length on mathematical reasoning benchmarks. AdaReason consistently achieves a better performance-to-token ratio than baseline methods. The merged models effectively interpolate between the specialized adapters, demonstrating fine-grained budget control.

	Accuracy					Generation Length				
	MATH	AIME	AMC	Olympiad	Avg.	MATH	AIME	AMC	Olympiad	Avg.
Navie-Merge										
$T_1 + T_2$	87.2	28.7	79.0	50.4	61.3	2186.8	5849.0	3455.0	3792.0	3820.7
$T_2 + T_3$	85.8	27.3	76.0	50.3	59.9	1920.7	4716.3	2896.0	2961.2	3123.5
$T_3 + T_4$	84.6	25.3	74.5	45.8	57.6	1466.4	3707.6	2291.4	2306.9	2443.1
$T_4 + T_5$	80.8	20.7	69.5	44.8	53.9	1044.1	2542.5	1743.9	1652.8	1745.8
AdaReason-Merge										
$M_{1.5}$	86.8	28.0	79.0	50.1	61.0	2186.7	5606.0	3360.5	3620.4	3693.4
$M_{2.5}$	86.2	26.3	77.0	49.0	59.6	1794.7	4566.8	2795.1	2930.1	3021.7
$M_{3.5}$	84.4	24.0	76.0	47.6	58.0	1475.5	3700.6	2286.5	2324.1	2446.7
$M_{4.5}$	82.2	21.3	72.5	45.8	55.5	1127.3	2740.8	1750.7	1809.9	1857.2

Table 2: Comparison of AdaReason versus naive pairwise merging on mathematical benchmarks.

less strict in GPQA, our method's ability to adhere to arbitrary budgets while maintaining a high level of performance underscores its versatility and broad applicability.

3.3 Ablation Studies

We further validate the runtime budget adaptation mechanism by comparing it to naive pairwise merging of adjacent adapters. As shown in Table 2, AdaReason outperforms naive merging at short generation lengths, with an average accuracy gain of up to 1.6% under the shortest budget. This demonstrates the effectiveness of our adaptive merging strategy in achieving smooth and optimal budget adaptation.

3.4 AdaReason Behavior Analysis

To understand how AdaReason adapts to varying budget constraints, we analyzed the composition of its reasoning process. Our findings indicate that the model learns to dynamically shift its strategy from expansive exploration to focused exploitation as the budget decreases.

We first performed a qualitative analysis of the model's outputs on the MATH dataset by categorizing generated text into four key behaviors: Verification, Exploration, Computation, and Conclusion, using a keyword-based methodology. As illustrated in Figure 4, there is a systematic shift in the model's reasoning style as the budget tightens. Specifically, the relative proportion of tokens allocated to "Verification" and "Exploration" diminishes significantly under stricter constraints. For instance, the token share for "Verification" decreased by 5.6% when the budget was reduced from $M_{1.5}$ to $M_{4.5}$. Conversely, the proportion of tokens dedicated to the final "Conclusion" and core "Computation" increases. This demonstrates that AdaReason learns to prioritize essential calculations and a direct path to the answer when resources are limited, effectively pruning its own divergent thought processes.

Furthermore, we analyzed how budget reductions affect the structure of the model's output. Figure 5 depicts the average word counts in thinking and solution segments across

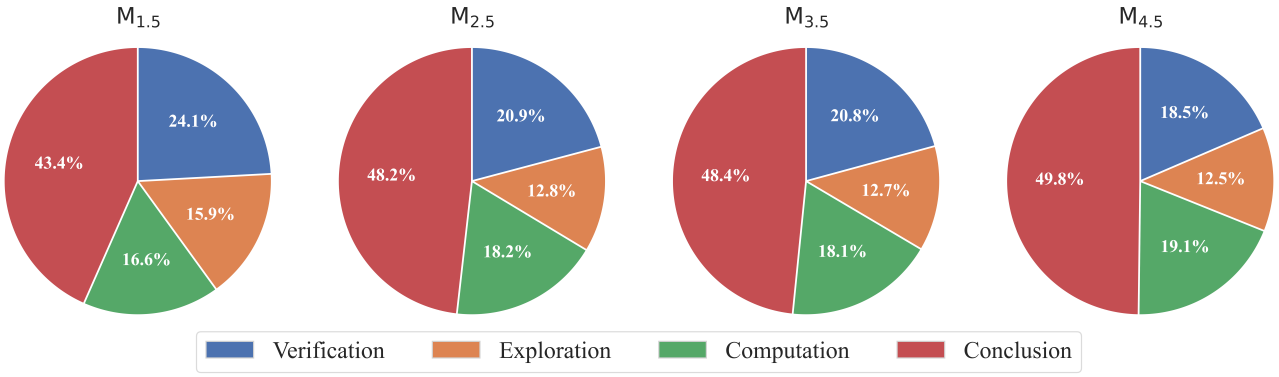

Figure 4: AdaReason thinking behavior across different length preferences.

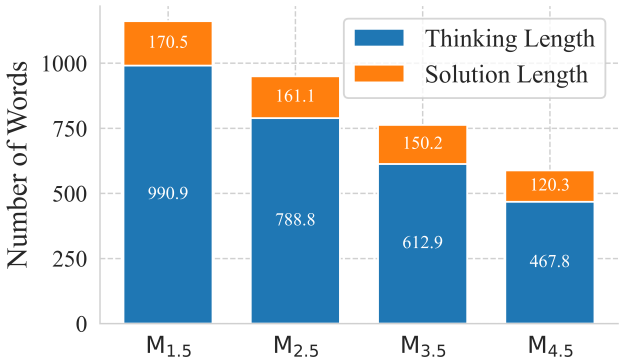

Figure 5: AdaReason thinking and solution lengths across different budgets.

budgets. Reducing the budget primarily shortens the thinking portion, preserving the solution's integrity. This suggests that AdaReason effectively eliminates redundant reasoning while maintaining essential outputs.

4 Related Work

4.1 Length-Controlled Reasoning

Research in length-controlled reasoning seeks to balance the accuracy and computational cost of Chain-of-Thought models (Xu et al. 2025a; Feng et al. 2025; Kang et al. 2024). Initial efforts were training-free, relying on prompt instructions, token budgets, or hard truncation to shorten outputs, but these methods often yielded inconsistent control and degraded reasoning quality (Lee, Che, and Peng 2025; Han et al. 2024a; Muennighoff et al. 2025). This motivated a shift toward training-based solutions, including supervised fine-tuning (SFT) on datasets with variable-length reasoning paths (Kang et al. 2024; Liu et al. 2024b; Ma et al. 2025) and teaching models to strategically skip non-essential steps (Xia et al. 2025).

Reinforcement learning methods recently emerged as the primary approach for controlling the length of language model outputs due to their simplicity and effectiveness (Liu et al. 2025). Most methods involve reward shaping, where models are incentivized to produce shorter responses by associating higher rewards with more concise outputs (Arora and Zanette 2025; Liu et al. 2025). Most methods incorporate length penalties during RL training to reduce reasoning verbosity (Team et al. 2025; Shen et al. 2025).

However, these methods require different models for different length preferences. L1 (Aggarwal and Welleck 2025) uses reinforcement learning with length penalties to train models that follow user-specified constraints. However, it suffers from training instability when dealing with diverse budgets in large context windows.

4.2 Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning methods like LoRA (Hu et al. 2022; Han et al. 2024b; Xu et al. 2023) enables efficient fine-tuning by learning low-rank updates to frozen base models. Building upon these foundations, researchers have explored multi-adapter architectures that leverage multiple LoRA modules for enhanced model capabilities (Wang et al. 2023; Yang et al. 2025; Zhong et al. 2024). E.g., ComposLoRA frameworks (Zhong et al. 2024) demonstrate that simultaneously incorporates all LoRAs can guide more cohesive image synthesis. LoRAExit (Liu et al. 2024a) introduces early exit strategies combined with adapter selection, enabling dynamic computation allocation based on input complexity. In contrast, our work is the first to leverage multiple LoRA adapters for reasoning length control without significant training overhead.

5 Conclusion

We presented AdaReason, a novel framework for training efficient reasoning models with flexible computational budgets. By decomposing the problem into multiple specialized adapters and introducing progressive training with length-adaptive rewards, we achieve superior performance while enabling unprecedented flexibility in runtime budget adaptation. Supported by theoretical analysis guaranteeing graceful performance scaling, AdaReason demonstrates superior accuracy at comparable token counts while offering unprecedented flexibility.

Acknowledgments

We sincerely thank all the anonymous reviewers for their valuable comments and feedback. This work is supported by the National Natural Science Foundation of China No.62441225, and in part by Young Elite Scientists Sponsorship Program by CAST (No.YESS20240529). Jiacheng Liu is the co-responding author.

References

Aggarwal, P.; and Welleck, S. 2025. L1: Controlling How Long A Reasoning Model Thinks With Reinforcement Learning. *arXiv preprint arXiv:2503.04697*.

Arora, D.; and Zanette, A. 2025. Training Language Models to Reason Efficiently. *arXiv preprint arXiv:2502.04463*.

Chen, L.; Ye, Z.; Wu, Y.; Zhuo, D.; Ceze, L.; and Krishnamurthy, A. 2024. Punica: Multi-tenant lora serving. *Proceedings of Machine Learning and Systems*, 6: 1–13.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H. P.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; Ray, A.; Puri, R.; Krueger, G.; Petrov, M.; Khlaaf, H.; Sastry, G.; Mishkin, P.; Chan, B.; Gray, S.; Ryder, N.; Pavlov, M.; Power, A.; Kaiser, L.; Bavarian, M.; Winter, C.; Tillet, P.; Such, F. P.; Cummings, D.; Plappert, M.; Chantzis, F.; Barnes, E.; Herbert-Voss, A.; Guss, W. H.; Nichol, A.; Paino, A.; Tezak, N.; Tang, J.; Babuschkin, I.; Balaji, S.; Jain, S.; Saunders, W.; Hesse, C.; Carr, A. N.; Leike, J.; Achiam, J.; Misra, V.; Morikawa, E.; Radford, A.; Knight, M.; Brundage, M.; Murati, M.; Mayer, K.; Welinder, P.; McGrew, B.; Amodei, D.; McCandlish, S.; Sutskever, I.; and Zaremba, W. 2021. Evaluating Large Language Models Trained on Code.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

Feng, S.; Fang, G.; Ma, X.; and Wang, X. 2025. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*.

Gao, B.; Song, F.; Yang, Z.; Cai, Z.; Miao, Y.; Dong, Q.; Li, L.; Ma, C.; Chen, L.; Xu, R.; et al. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

Han, T.; Wang, Z.; Fang, C.; Zhao, S.; Ma, S.; and Chen, Z. 2024a. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

Han, Z.; Gao, C.; Liu, J.; Zhang, J.; and Zhang, S. Q. 2024b. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

He, C.; et al. 2024. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. *arXiv preprint arXiv:2402.14008*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Kang, Y.; Sun, X.; Chen, L.; and Zou, W. 2024. C3oT: Generating Shorter Chain-of-Thought without Compromising Effectiveness. *arXiv preprint arXiv:2412.11664*.

Lee, A.; Che, E.; and Peng, T. 2025. How well do llms compress their own chain-of-thought? a token complexity approach. *arXiv preprint arXiv:2503.01141*.

Li, Z.-Z.; Zhang, D.; Zhang, M.-L.; Zhang, J.; Liu, Z.; Yao, Y.; Xu, H.; Zheng, J.; Wang, P.-J.; Chen, X.; et al. 2025. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Liu, J.; Tang, P.; Hou, X.; Li, C.; and Heng, P.-A. 2024a. LoRAExit: Empowering Dynamic Modulation of LLMs in Resource-limited Settings using Low-rank Adapters. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 9211–9225.

Liu, T.; Guo, Q.; Hu, X.; Jiayang, C.; Zhang, Y.; Qiu, X.; and Zhang, Z. 2024b. Can language models learn to skip steps? *arXiv preprint arXiv:2411.01855*.

Liu, W.; Zhou, R.; Deng, Y.; Huang, Y.; Liu, J.; Deng, Y.; Zhang, Y.; and He, J. 2025. Learn to Reason Efficiently with Adaptive Length-based Reward Shaping. *arXiv preprint arXiv:2505.15612*.

Luo, M.; et al. 2025. DeepScaleR: Surpassing O1-Preview with a 1.5B Model by Scaling RL. *Notion Blog*.

Ma, X.; Wan, G.; Yu, R.; Fang, G.; and Wang, X. 2025. CoT-Valve: Length-Compressible Chain-of-Thought Tuning. *arXiv preprint arXiv:2502.09601*.

Min, Y.; Chen, Z.; Jiang, J.; Chen, J.; Deng, J.; Hu, Y.; Tang, Y.; Wang, J.; Cheng, X.; Song, H.; et al. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413*.

Muennighoff, N.; et al. 2025. S1: Simple Test-Time Scaling. *arXiv preprint*.

OpenAI. 2024. OpenAI o1 System Card. *arXiv preprint arXiv:2412.16720*.

Plaat, A.; Wong, A.; Verberne, S.; Broekens, J.; van Stein, N.; and Back, T. 2024. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Shen, Z.; Yan, H.; Zhang, L.; Hu, Z.; Du, Y.; and He, Y. 2025. Codi: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*.

Sheng, G.; Zhang, C.; Ye, Z.; Wu, X.; Zhang, W.; Zhang, R.; Peng, Y.; Lin, H.; and Wu, C. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, 1279–1297.

Sui, Y.; Chuang, Y.-N.; Wang, G.; Zhang, J.; Zhang, T.; Yuan, J.; Liu, H.; Wen, A.; Zhong, S.; Chen, H.; et al. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Wang, Y.; Lin, Y.; Zeng, X.; and Zhang, G. 2023. Multilora: Democratizing lora for better multi-task learning. *arXiv preprint arXiv:2311.11501*.

Xia, H.; Li, Y.; Leong, C. T.; Wang, W.; and Li, W. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*.

Xu, F.; Hao, Q.; Zong, Z.; Wang, J.; Zhang, Y.; Wang, J.; Lan, X.; Gong, J.; Ouyang, T.; Meng, F.; et al. 2025a. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

Xu, L.; Xie, H.; Qin, S.-Z. J.; Tao, X.; and Wang, F. L. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Xu, S.; Xie, W.; Zhao, L.; and He, P. 2025b. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

Yang, Y.; Muhtar, D.; Shen, Y.; Zhan, Y.; Liu, J.; Wang, Y.; Sun, H.; Deng, W.; Sun, F.; Zhang, Q.; et al. 2025. Mtl-lora: Low-rank adaptation for multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 22010–22018.

Zhong, M.; Shen, Y.; Wang, S.; Lu, Y.; Jiao, Y.; Ouyang, S.; Yu, D.; Han, J.; and Chen, W. 2024. Multi-lora composition for image generation. *arXiv preprint arXiv:2402.16843*.