# Long-Chain Reasoning Distillation via Adaptive Prefix Alignment

**Zhenghao Liu**[1*†], **Zhuoyang Wu**[1*], **Xinze Li**[1], **Yukun Yan**[2],
**Shuo Wang**[2], **Zulong Chen**[3], **Yu Gu**[1], **Ge Yu**[1], **Maosong Sun**[2]
[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3]Alibaba Group, Hangzhou, China

## Abstract

Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities, particularly in solving complex mathematical problems. Recent studies show that distilling long reasoning trajectories can effectively enhance the reasoning performance of small-scale student models. However, teacher-generated reasoning trajectories are often excessively long and structurally complex, making them difficult for student models to learn. This mismatch leads to a gap between the provided supervision signal and the learning capacity of the student model. To address this challenge, we propose **P**refix-**ALIGN**ment distillation (P-ALIGN), a framework that fully exploits teacher CoTs for distillation through adaptive prefix alignment. Specifically, P-ALIGN adaptively truncates teacher-generated reasoning trajectories by determining whether the remaining suffix is concise and sufficient to guide the student model. Then, P-ALIGN leverages the teacher-generated prefix to supervise the student model, encouraging effective prefix alignment. Experiments on multiple mathematical reasoning benchmarks demonstrate that P-ALIGN outperforms all baselines by over 3%. Further analysis indicates that the prefixes constructed by P-ALIGN provide more effective supervision signals, while avoiding the negative impact of redundant and uncertain reasoning components. All codes are available at https://github.com/NEUIR/P-ALIGN.

## 1 Introduction

Large Language Models (LLMs) have exhibited impressive reasoning capabilities across complex tasks, especially in solving mathematical problems (Brown et al., 2020; Zhang et al., 2022). By adopting the long-form chain-of-thoughts (CoTs) paradigm (DeepSeek-AI et al., 2025), LLMs can engage in deeper, iterative thinking, which substantially improves their performance on challenging reasoning tasks (Yang et al., 2025; Chen et al., 2025a). Despite these successes, enabling small-scale models to generate outputs that exhibit such reflective and structured reasoning behavior remains a significant challenge when tackling complex mathematical problems (Luo et al., 2025a).

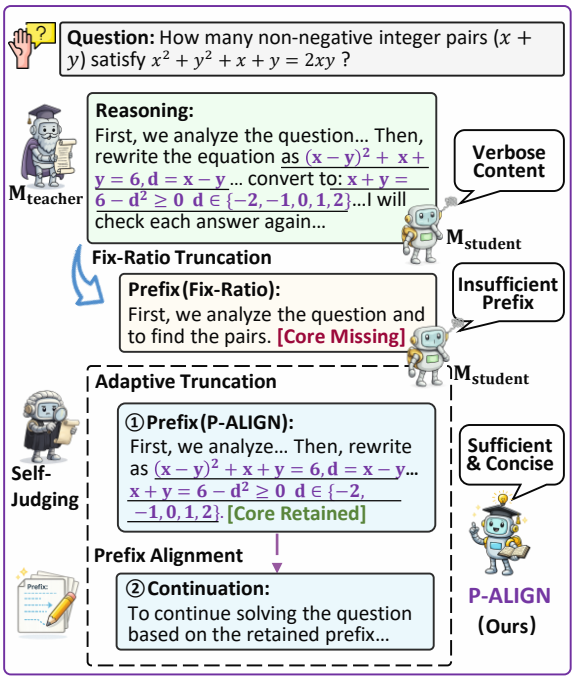



Figure 1: The Framework of Our P-ALIGN Model. Later steps in traces are often more uncertain and harder for the student to learn. P-ALIGN therefore performs adaptively selecting a sufficient prefix and aligning student supervision to it for more effective distillation.

Recent work has explored distilling reasoning abilities from long-form CoTs to improve the performance of small-scale models on mathematical reasoning benchmarks (Ye et al., 2025; DeepSeek-AI et al., 2025). In these approaches, small-scale models are treated as student models and trained via Supervised Fine-Tuning (SFT) on full long-

* indicates equal contribution.
† indicates corresponding author.



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18041–18059
July 2-7, 2026 

form reasoning trajectories generated by Reasoning Language Models (RLMs) (Hsieh et al., 2023). However, such intricate and reflective reasoning trajectories often exceed the learning capacity of student models (Li et al., 2025), making it difficult for them to internalize the underlying reasoning patterns and, in some cases, even leading to degraded reasoning performance (Luo et al., 2025a). To address this issue, several recent studies attempt to refine teacher-generated trajectories using LLM-based refinement techniques to reduce redundancy and improve the compatibility of distilled reasoning with student models (Wu et al., 2025; Zeng et al., 2025b). Nevertheless, refinement-based methods introduce an additional and weakly constrained transformation stage, where the refined trajectories are not explicitly optimized to match the capacity of the student model. To better exploit teacher-generated reasoning trajectories, existing works (Ji et al., 2025; Chen et al., 2025b) incorporate prefixes into the SFT process, as these prefixes help preserve structural knowledge while mitigating uncertainty and redundancy that tend to emerge in later reasoning steps. However, as shown in Figure 1, using a fixed ratio for prefix truncation renders this strategy suboptimal: prefixes may still fail to retain sufficient information for complex tasks.

In this paper, we introduce the Adaptive **P**refix-**ALIGN**ment reasoning distillation method (P-ALIGN), which adaptively truncates prefixes from teacher-generated reasoning trajectories and optimizes student models to align with the retained teacher prefixes. Specifically, P-ALIGN enables the student model to adaptively assess the sufficiency of a truncated prefix and identify a minimal sufficient prefix boundary via binary search, ensuring that the retained prefix is both effective and concise. Building on this adaptive prefix truncation strategy, P-ALIGN further performs prefix-based alignment by treating the retained teacher prefix as prior reasoning context and guiding the student to generate a complete reasoning trajectory, which serves as the supervision signal for SFT.

Our experimental results demonstrate that P-ALIGN consistently outperforms all baselines across a range of mathematical tasks, highlighting its effectiveness in distilling long-form reasoning into student models. Further analysis reveals that reasoning content from later steps introduces increased uncertainty into student models, which in turn degrades distillation performance. Therefore, to fully exploit the teacher-generated reasoning trajectories, P-ALIGN adaptively applies prefix truncation to the reasoning trajectories, retaining more informative content while reducing unnecessary uncertainty and bias when guiding the student model on problems of varying difficulty. In addition, the prefix alignment strategy in P-ALIGN further demonstrates its effectiveness under the SFT setting by enabling the student model to generate complete reasoning chains autonomously. This design avoids overfitting to incomplete or overly short reasoning patterns, a common issue in prefix-based SFT methods (Ji et al., 2025).

## 2 Related Work

Large Language Models (LLMs) have demonstrated strong capabilities in mathematical reasoning (Cobbe et al., 2021; Hendrycks et al., 2021). Chain-of-Thought (CoT) (Wei et al., 2022) elicits step-by-step reasoning and has been shown to substantially improve performance on mathematical reasoning benchmarks (Li et al., 2023; Qin et al., 2023; Luo et al., 2023). To enhance small-scale student models, prior work commonly leverages a stronger LLM as a teacher to generate long-form reasoning trajectories and uses these trajectories as supervision signals to fine-tune student models (DeepSeek-AI et al., 2025; Yang et al., 2025). However, the uncertainty and redundancy of teacher-generated long-form CoTs often exceed the capacity of student models, making it challenging to faithfully distill such reasoning capabilities into them (Chen et al., 2025a; Li et al., 2025).

To mitigate the challenges of distilling long-form reasoning trajectories, recent work has increasingly focused on constructing more efficient and higher-quality SFT data. One line of research investigates quality-based data selection, where sophisticated criteria are employed to identify a small set of high-quality examples for training (Ye et al., 2025; Muennighoff et al., 2025). Another line of work refines teacher-generated reasoning trajectories through prompting strategies or structural editing, producing shorter yet still effective trajectories (Xu et al., 2025; Wu et al., 2025; Jin et al., 2025). However, these selection and refinement approaches typically involve multiple sequential stages, such as iterative improvements (Zelikman et al., 2022) and verification, to refine or select high-quality long-form reasoning trajectories.

Unlike these sophisticated CoT selection or refinement methods, recent studies have proposed

unsupervised approaches to fully exploit the effectiveness of teacher-generated reasoning trajectories. Specifically, Ji et al. (2025) observe that the prefixes of reasoning trajectories are typically consistent; they then extract such prefixes and mix them with full CoTs as supervision during SFT. Subsequent studies (Chen et al., 2025b; Sun et al., 2025) further explore the use of randomly truncated prefixes from teacher-generated reasoning trajectories to either enhance reasoning performance or improve trajectory sampling efficiency in RL training. However, employing a fixed truncation ratio lacks adaptability across problem difficulties, often failing to preserve reasoning information that is both sufficient for complex tasks and concise for simpler ones.

## 3 Methodology

In this section, we present **P**refix-**ALIGN**ment distillation (P-ALIGN), a framework designed to distill long-chain reasoning capabilities into small-scaled models efficiently. As shown in Figure 2, we first describe the standard supervised fine-tuning (SFT) for reasoning distillation, along with its truncated-prefix SFT variant (Sec. 3.1). We then introduce our P-ALIGN model (Sec. 3.2), in which the student model evaluates reasoning trajectories and uses binary search to truncate them into minimal, sufficient prefixes aligned with its needs to solve the given problem. Then P-ALIGN treats the retained prefixes as prior reasoning context to guide prefix-aligned generation of full reasoning sequences for effective supervision.

### 3.1 Distilling Long-Chain Reasoning via Supervised Fine-Tuning

Given a mathematical problem $q$, a Large Language Model (LLM) is typically prompted with a problem-solving instruction ($\text{Instruct}_{\text{QA}}$) to generate a complete solution. In reasoning distillation, we treat the small-scale model as the student model $\mathcal{M}_{\text{student}}$ and employ different Supervised Fine-Tuning (SFT) strategies to optimize it using signals derived from a stronger teacher model.

**Distillation from Long-Form CoT.** To distill the reasoning capability of a teacher model $\mathcal{M}_{\text{teacher}}$ into a student model $\mathcal{M}_{\text{student}}$, existing approaches (Wang et al., 2023; Yin et al., 2025; DeepSeek-AI et al., 2025) typically use the reasoning trajectories generated by $\mathcal{M}_{\text{teacher}}$ as supervision signals for SFT. Specifically, given a question $q$, we adopt a Reasoning Language Model (RLM), such as DeepSeek-R1 (DeepSeek-AI et al., 2025), as the teacher model and prompt it to produce a long-form reasoning response $R$:

$$R = \mathcal{M}_{\text{teacher}}(\text{Instruct}_{\text{QA}}(q)). \tag{1}$$

We then collect the query–response pairs as the training dataset $\mathcal{D} = \{(q^1, R^1), \ldots, (q^n, R^n)\}$, which consists of $n$ examples. The training objective minimizes the negative log-likelihood of the teacher-generated reasoning trajectory $R^i$:

$$\mathcal{J} = -\sum_{i=1}^{n}\sum_{t=1}^{|R^i|} \log P(R_t^i | R_{<t}^i, \text{Instruct}_{\text{QA}}(q^i); \mathcal{M}_{\text{student}}), \tag{2}$$

where $|R^i|$ denotes the token number of the generated reasoning trajectory $R^i$. Although distilling long-form CoTs from the teacher model substantially improves LLM problem-solving accuracy (Hsieh et al., 2023), the student model may struggle to effectively learn from these long-form reasoning trajectories (Luo et al., 2025b; Gudibande et al., 2023). This limitation has motivated recent efforts toward constructing higher-quality SFT datasets (Wettig et al., 2024).

**SFT with Truncated Prefixes.** To further improve the quality of supervision signals, recent studies (Ji et al., 2025; Chen et al., 2025b) demonstrate that incorporating truncated prefixes during SFT can enhance model performance, since different solution paths often share a common initial reasoning trajectory. Accordingly, they construct a prefix-truncated dataset $\mathcal{D}_{\text{Prefix}}$ to encourage the student model to better learn from the early stages of reasoning trajectories, which is then mixed with the original training dataset $\mathcal{D}$ for SFT.

To construct the prefix-truncated dataset $\mathcal{D}_{\text{Prefix}}$, the $\mathcal{M}_{\text{teacher}}$-generated reasoning trajectory $R^i$ is truncated using the function $\text{Truncate}(\cdot)$:

$$\tilde{R}^i = \text{Truncate}(R^i, \lambda \cdot |R^i|), \tag{3}$$

where $\lambda \in (0, 1)$ controls the truncation ratio and $\text{Truncate}(\cdot)$ retains the first $\lambda \cdot |R^i|$ tokens of $R^i$. The resulting prefix $\tilde{R}^i$ is directly used as the supervision signal for training. We then construct the prefix SFT dataset $\mathcal{D}_{\text{Prefix}} = \{(q^1, \tilde{R}^1), \ldots, (q^n, \tilde{R}^n)\}$ by pairing each input query $q^i$ with its corresponding truncated reasoning prefix. However, such ratio-based truncation strategies may still discard necessary information from the reasoning trajectories, potentially leading to incomplete or incorrect reasoning outcomes.

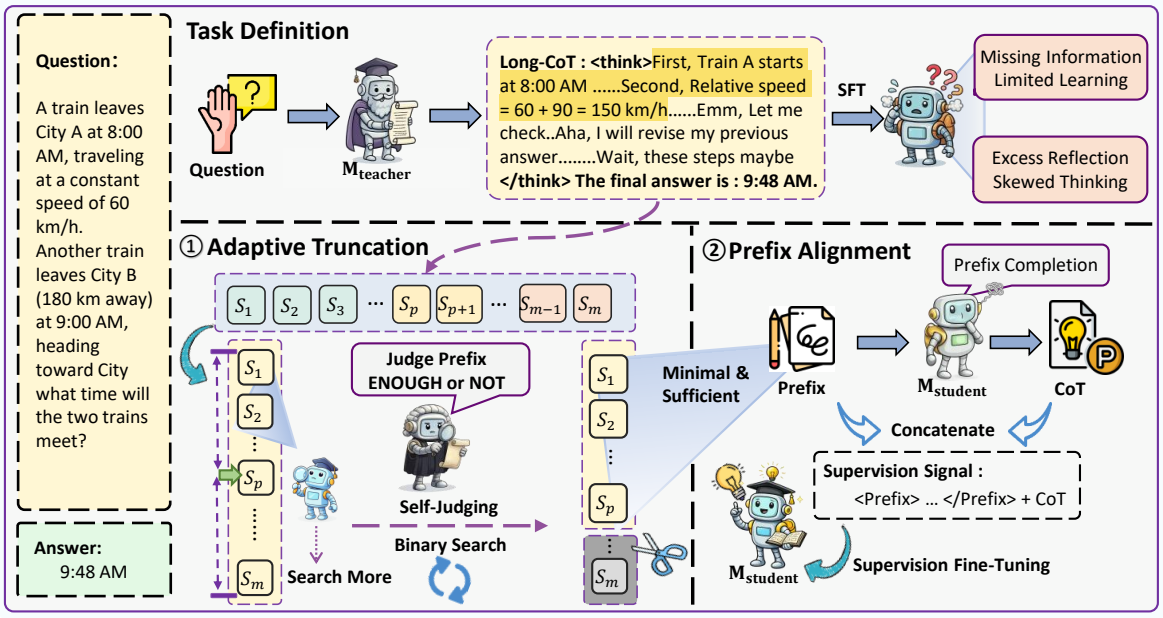



Figure 2: Illustration of P-ALIGN Model.

### 3.2 Adaptive Prefix Alignment based Reasoning Capability Distillation

To further enhance the distillation of the reasoning capability from $\mathcal{M}_{\text{teacher}}$, P-ALIGN introduces an adaptive prefix alignment strategy, which encourages the student model $\mathcal{M}_{\text{student}}$ to dynamically truncate prefixes from reasoning trajectories. Specifically, P-ALIGN prompts $\mathcal{M}_{\text{student}}$ to assess the sufficiency of a prefix for truncation, while ensuring minimality so as to avoid the introduction of redundant or noisy information. Subsequently, the student model $\mathcal{M}_{\text{student}}$ is optimized via SFT to align with the truncated prefixes generated by the teacher model $\mathcal{M}_{\text{teacher}}$.

#### 3.2.1 Efficient Minimal Prefix Truncation via Binary Search

To adaptively truncate prefixes from reasoning trajectories, P-ALIGN prompts the student model $\mathcal{M}_{\text{student}}$ to progressively read prefix sentences from a teacher-generated reasoning trajectory $R^i$ and determine whether the truncated prefix contains sufficient information to solve the problem, while maintaining a minimal length. To enable efficient identification of such a prefix, we also employ a binary search algorithm to locate a tailored truncation point.

**Prefix Evaluation by Self-Judging.** For the $i$-th reasoning trajectory $R^i$ generated by the teacher model $\mathcal{M}_{\text{teacher}}$, P-ALIGN first segments $R^i$ into $m$ sentences:

$$R^i = \{r_1^i, r_2^i, \ldots, r_m^i\}. \tag{4}$$

These sentences serve as the basic units for prefix truncation, rather than individual tokens (Chen et al., 2025b), in order to preserve the semantic completeness of the truncated prefixes. We then prompt the student model $\mathcal{M}_{\text{student}}$ to self-evaluate whether the current truncated prefix $\tilde{R}^i$ contains sufficient information to solve the problem $q^i$:

$$L = \mathcal{M}_{\text{student}}(\text{Instruct}_{\text{Eval}}(q^i, \tilde{R}^i)), \tag{5}$$

where $\text{Instruct}_{\text{Eval}}$ denotes an evaluation instruction that guides $\mathcal{M}_{\text{student}}$ to make a sufficiency judgement. The judgment label $L \in \{\texttt{ENOUGH}, \texttt{NOT\_ENOUGH}\}$ represents the predicted sufficiency of the current prefix $\tilde{R}^i$.

**Binary Search for Efficient Truncation.** To efficiently identify the minimal sufficient truncation point for $q^i$, we perform binary search over the sentence sequence $R^i = \{r_1^i, r_2^i, \ldots, r_m^i\}$ to locate the shortest prefix that satisfies the sufficiency criterion. In the binary search procedure, the search boundaries $\ell$ and $r$ are initialized as $\ell = 1$ and $r = m$. The truncation point $p$ is selected as the midpoint of the current interval:

$$p = \left\lfloor \frac{\ell + r}{2} \right\rfloor. \tag{6}$$

We truncate $R^i$ to its first $p$ sentences to obtain the candidate prefix $\tilde{R}^i = R^i_{1:p}$. If the evaluation result

is ENOUGH, the current truncation point $p$ is deemed feasible, and we continue searching for a shorter sufficient prefix by updating the right boundary to $r = p$. Conversely, if the label is NOT_ENOUGH, the prefix lacks sufficient information, and we extend the search to the right half by setting $\ell = p + 1$. Formally, the boundary update rule is defined as:

$$(\ell, r) = \begin{cases} (\ell, p), & \text{if } L = \texttt{ENOUGH}, \\ (p+1, r), & \text{if } L = \texttt{NOT\_ENOUGH}. \end{cases} \quad (7)$$

This binary search process is repeated on the updated interval $[\ell, r]$ by recomputing the midpoint in Eq. 6 and adjusting the boundaries according to Eq. 7. The search terminates when $\ell = r$, yielding the final truncation point $p^*$ that is equal to $\ell$. The resulting minimal prefix $\tilde{R}^i = R^i_{1:p^*}$ thus consists of the first $p^*$ sentences of $R^i$. For clarity, we provide a complete procedure of this binary search in Appendix A.10.

#### 3.2.2 Prefix-based Alignment for Effective Supervised Fine-Tuning

To better align the student model with the retained prefixes $\tilde{R}^i$ introduced in Sec. 3.2.1, we first collect all query-prefix pairs into a dataset $\mathcal{D}_{\text{Prefix}} = \{(q^1, \tilde{R}^1), \ldots, (q^n, \tilde{R}^n)\}$. Based on this dataset, we further complete each retained prefix into a full reasoning trajectory, thereby constructing the alignment dataset $\mathcal{D}_{\text{align}}$ for training the student model $\mathcal{M}_{\text{student}}$.

Specifically, for each pair $(q^i, \tilde{R}^i)$, an instruction $\text{Instruct}_{\text{Align}}$ prompts the student model $\mathcal{M}_{\text{student}}$ to condition on $\tilde{R}^i$ as prior reasoning context and then continue the reasoning process to generate a complete reasoning trajectory that is consistent with the given prefix:

$$y^i = \mathcal{M}_{\text{student}}(\text{Instruct}_{\text{Align}}(q^i, \tilde{R}^i)). \quad (8)$$

To enforce prefix-based alignment during SFT, we concatenate the retained prefix $\tilde{R}^i$ with the student-generated continuation $y^i$ as the supervision signal. To ensure the correctness of the constructed supervision signals, we conduct the $\mathcal{D}_{\text{Align}}$ dataset by retaining only those samples whose final answers match the ground-truth answer $a^i_*$:

$$\mathcal{D}_{\text{align}} = \{(q^i, \tilde{R}^i \oplus y^i) \mid 1 \le i \le n,\ \text{Ans}(y^i) = a^i_*\}, \quad (9)$$

where $\text{Ans}(\cdot)$ denotes an answer extraction function. The resulting dataset $\mathcal{D}_{\text{align}}$ is then used for fine-tuning the student model $\mathcal{M}_{\text{student}}$ using Eq. 2, enabling the student model to align its reasoning prefixes with those of the teacher model.

## 4 Experimental Methodology

This section first introduces the datasets, baselines, and evaluation metrics, and then details the implementation settings used in our experiments.

**Datasets.** We construct the training data using the training split of the s1K-1.1 dataset (Muennighoff et al., 2025), which contains high-quality solution traces generated by the DeepSeek-R1 model (DeepSeek-AI et al., 2025). For evaluation, we consider four mathematical problem datasets that span a wide range of difficulty levels. MATH500 (Godahewa et al., 2021) is a benchmark composed of competition-level mathematics problems with varying degrees of complexity. AIME (AIM, 2025) and AMC (AMC, 2025) assess mathematical problem-solving capabilities across arithmetic, algebra, counting, geometry, number theory, probability, and other topics in secondary school mathematics.

**Baselines.** We compare P-ALIGN against both zero-shot and SFT-based baselines. In the zero-shot setting, the model is prompted with the question and directly generates a solution without task-specific fine-tuning. We further include three SFT baselines: SFT (Label), SFT (Long-CoT) (DeepSeek-AI et al., 2025), and UPFT (Ji et al., 2025). SFT (Label) trains the student model using ground-truth labels provided in the dataset. Following prior work (Wang et al., 2024), SFT (Long-CoT) fine-tunes the student model using full-length reasoning traces produced by the teacher model. In addition, UPFT selects prefixes based on self-consistency and conducts training on truncated prefixes, using the first 32 tokens in accordance with the original experimental setup.

**Evaluation Metrics.** Following prior work (Zhang et al., 2024), we evaluate model performance using Pass@1 and Pass@3. Pass@1 measures the accuracy of a single generated answer, while Pass@3 denotes the proportion of problems for which at least one of three sampled outputs matches the ground-truth answer.

**Implementation Details.** All experiments employ Qwen2.5-7B-Instruct (Yang et al., 2024) and Qwen3-8B (Yang et al., 2025) as student models, with DeepSeek-R1 (DeepSeek-AI et al., 2025) serving as the teacher model. We train the models for 3 epochs with a learning rate of $5 \times 10^{-5}$ and adopt LoRA (Hu et al., 2022) for parameter-efficient fine-tuning. Our P-ALIGN is implemented

| | AIME25 | | AIME24 | | AMC12 | | MATH500 | | Avg. | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 |
| **Qwen2.5-7B-Instruct** | | | | | | | | | | |
| Zero-Shot | 6.67 | 10.00 | 10.00 | 20.00 | 40.76 | 49.70 | 69.80 | 84.80 | 31.81 | 41.13 |
| SFT (Label) | 3.33 | 6.67 | 3.33 | 10.00 | 32.92 | 34.62 | 62.20 | 71.00 | 25.45 | 30.57 |
| SFT (Long-CoT) | 10.00 | 20.00 | 10.00 | 26.67 | 48.19 | 56.63 | 75.60 | **86.60** | 35.95 | 47.68 |
| UPFT (2025) | 13.33 | 20.00 | 13.33 | 23.33 | 47.58 | 59.04 | 74.80 | 84.60 | 37.26 | 46.74 |
| P-ALIGN | **16.67** | **26.67** | **16.67** | **26.67** | **49.40** | **63.86** | **75.80** | 85.20 | **39.64** | **50.60** |
| **Qwen3-8B** | | | | | | | | | | |
| Zero-Shot | 20.00 | 23.33 | 26.67 | 36.67 | 59.84 | 71.08 | 83.60 | 91.20 | 47.53 | 55.57 |
| SFT (Label) | 10.00 | 13.33 | 10.00 | 10.00 | 48.51 | 59.04 | 73.20 | 84.60 | 35.43 | 41.74 |
| SFT (Long-CoT) | 23.33 | 30.00 | 26.67 | **40.00** | 58.81 | 67.04 | 84.40 | 90.80 | 48.30 | 56.96 |
| UPFT (2025) | 23.33 | 26.67 | 23.33 | 36.67 | 61.74 | 75.38 | **85.60** | 92.40 | 48.50 | 57.78 |
| P-ALIGN | **23.33** | **33.33** | **30.00** | 36.67 | **66.27** | **77.11** | 84.80 | **92.80** | **51.10** | **59.98** |

Table 1: Overall Performance.

based on TRL[1] and LLaMA Factory[2]. Additional experimental details are provided in Appendix A.2, and the prompt templates are presented in Appendix A.3.

## 5 Evaluation Results

In this section, we first evaluate the overall performance of P-ALIGN through the main experiments. Next, we conduct ablation studies to examine the contribution of different components in P-ALIGN. Furthermore, we analyze the effectiveness of distilled models under different prefix truncation strategies. Finally, we investigate positional effects in long-form reasoning trajectories, aiming to analyse how prefix-based designs contribute to improved model performance. The case study is conducted in Appendix A.11.

### 5.1 Overall Performance

As shown in Table 1, we compare P-ALIGN with several baseline models across different mathematical reasoning tasks.

The evaluation results indicate that P-ALIGN consistently outperforms all baselines, highlighting the effectiveness of our prefix-alignment framework for reasoning distillation. Notably, P-ALIGN achieves consistent improvements across different backbone models, demonstrating its generalization ability. Among different SFT strategies, SFT (Label) performs substantially worse than the zero-shot model, suggesting that supervising the student model solely with annotated ground-truth targets may lead to overfitting and consequently degrade its reasoning capability. In contrast, SFT (Long-CoT) consistently improves the reasoning performance of student models, as teacher-generated long-form reasoning trajectories provide richer and more informative patterns for imitation. Building upon the distillation of teacher-generated long-form reasoning trajectories, P-ALIGN further constructs higher-quality supervision signals by adaptively truncating a minimal sufficient prefix through student self-judging and performing prefix-based alignment for distillation, yielding an additional 3% improvement over SFT (Long-CoT). Compared with UPFT, P-ALIGN achieves an improvement of more than 2%, demonstrating the crucial role of adaptive prefix alignment, rather than relying on brute-force prefix truncation with a fixed token budget for SFT. In addition, we report results for student models at different scales in Appendix A.9.

### 5.2 Ablation Study

This subsection conducts ablation studies to investigate the contributions of different components in our P-ALIGN model.

As shown in Table 2, we first evaluate two variants, SFT w/ Teacher Prefix and SFT w/ Student CoT, to analyze the impact of different supervision signals. Specifically, SFT w/ Teacher Prefix only leverages the teacher-generated prefix, while SFT w/ Student CoT relies solely on student-generated reasoning conditioned on the teacher-generated prefix. We then conduct three additional variants to examine the effectiveness of the prefix truncation strategy in P-ALIGN. The P-ALIGN (InfoGain) model adopts InfoGain (Wang et al., 2025) as the criterion for prefix truncation. P-ALIGN w/o Binary Search removes the binary search procedure for identifying the truncation point. In contrast, P-ALIGN w/o Adaptive Read directly feeds the entire teacher-generated trajectory to the student model without adaptive truncation.

Compared with P-ALIGN, both SFT w/ Teacher Prefix and SFT w/ Student CoT decrease the an-

[1] https://github.com/huggingface/trl
[2] https://github.com/hiyouga/LLaMA-Factory

| | AIME25 | | AIME24 | | AMC12 | | MATH500 | | Avg. | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 |
| **Qwen2.5-7B-Instruct** | | | | | | | | | | |
| SFT (Long-CoT) | 10.00 | 20.00 | 10.00 | 26.67 | 48.19 | 56.63 | 75.60 | **86.60** | 35.95 | 47.68 |
| P-ALIGN | **16.67** | **26.67** | **16.67** | **26.67** | **49.40** | 63.86 | **75.80** | 85.20 | **39.64** | **50.60** |
| SFT w/ Teacher Prefix | 13.33 | 20.00 | 13.33 | 23.33 | 46.71 | 54.22 | 70.20 | 82.80 | 35.89 | 45.09 |
| SFT w/ Student CoT | 13.33 | 16.67 | 13.33 | 20.00 | 40.55 | 56.63 | 71.60 | 83.80 | 34.70 | 44.28 |
| P-ALIGN (InfoGain) | 13.33 | 20.00 | 10.00 | 20.00 | 42.17 | 55.90 | 73.80 | 85.40 | 34.83 | 45.33 |
| P-ALIGN w/o Adaptive Read | 13.33 | 16.67 | 16.67 | 23.33 | 47.53 | 60.19 | 73.60 | 84.80 | 37.78 | 46.25 |
| P-ALIGN w/o Binary Search | 16.67 | 23.33 | 16.67 | 26.67 | 49.26 | **64.07** | 75.00 | 85.20 | 39.40 | 49.82 |
| **Qwen3-8B** | | | | | | | | | | |
| SFT (Long-CoT) | 23.33 | 30.00 | 26.67 | 40.00 | 58.81 | 67.04 | 84.40 | 90.80 | 48.30 | 56.96 |
| P-ALIGN | 23.33 | **33.33** | **30.00** | 36.67 | **66.27** | **77.11** | 84.80 | **92.80** | **51.10** | **59.98** |
| SFT w/ Teacher Prefix | 16.67 | 30.00 | 26.67 | 33.33 | 60.24 | 72.29 | 83.60 | 92.20 | 46.80 | 56.96 |
| SFT w/ Student CoT | 16.67 | 30.00 | 23.33 | 30.00 | 60.04 | 71.08 | 84.60 | 91.20 | 46.16 | 55.57 |
| P-ALIGN (InfoGain) | 23.33 | 26.67 | 23.33 | 33.33 | 63.58 | 73.21 | 84.20 | 88.40 | 48.61 | 55.40 |
| P-ALIGN w/o Adaptive Read | 23.33 | 26.67 | 26.67 | **40.00** | 62.75 | 75.64 | 82.20 | 92.80 | 48.74 | 58.36 |
| P-ALIGN w/o Binary Search | **26.67** | 26.67 | 26.67 | 40.00 | 62.63 | 76.34 | **85.00** | 91.40 | 50.24 | 58.60 |

Table 2: Ablation Study.

swering accuracy by more than 3%, highlighting the effectiveness of incorporating both supervision components. Each component contributes to constructing higher-quality supervision, which helps distill the teacher's reasoning patterns into the student model. The main reason may lie in the fact that the teacher prefix guides the student model to align with the teacher's reasoning process, while the student-generated CoT provides more complete and prefix-guided reasoning trajectories for SFT. We then ablate different prefix truncation strategies to further verify the effectiveness of our truncation procedure. The InfoGain-based variant determines prefix boundaries based on changes in information entropy (Wang et al., 2025). Although it improves over the vanilla LLM, it underperforms SFT (Long-CoT), demonstrating the advantage of using student self-judging to assess prefix sufficiency. Next, when directly truncating teacher-generated reasoning trajectories (P-ALIGN w/o Adaptive Read), the performance of P-ALIGN decreases, indicating that reading the entire reasoning trajectories makes it difficult to localize a concise yet sufficient boundary. Finally, although P-ALIGN w/o Binary Search achieves performance comparable to P-ALIGN, it incurs substantially higher computational cost, resulting in more than 20 times higher latency than P-ALIGN when searching for the truncation point (More details are provided in Appendix A.5).

### 5.3 Distillation Performance with Different Prefix Truncation Strategies

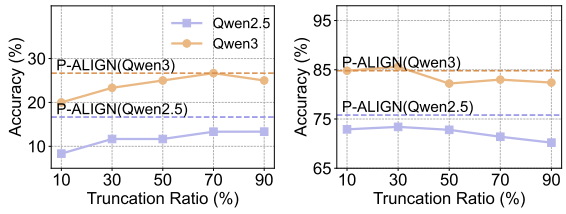
In this section, we investigate the distillation performance under different prefix truncation strategies. As shown in Figure 3, we first compare fixed-ratio truncation to motivate the necessity of adaptive prefix selection, and then analyze the length and quality of responses generated by models optimized using different prefix truncation strategies.

**Optimization with Fixed Prefix Ratios.** To evaluate the effectiveness of our adaptive prefix truncation method in P-ALIGN, we compare P-ALIGN with fixed ratio-based truncation strategies, where the truncation ratio ranges from 10% to 90%.

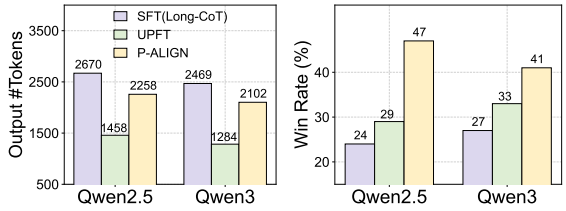
As shown in Figure 3(a), on competition-level benchmarks (AIME24&25), the performance of optimized LLMs generally improves as longer prefixes are retained. In contrast, on the relatively easier dataset MATH500, performance degrades when longer prefixes are used (Figure 3(b)). This suggests that, for difficult problems, longer reasoning trajectories provide informative and effective supervision signals for student model learning. Conversely, for easier problems, extended reasoning trajectories may introduce the overthinking issue (Chen et al., 2024), which can mislead the student model and degrade its reasoning capability (Luo et al., 2025a). Benefiting from adaptive prefix truncation, P-ALIGN consistently outperforms models trained with fixed-ratio truncation strategies across different settings, highlighting the critical role of adaptive truncation in balancing insufficient and excessive prefixes to deliver more effective supervision signals.

**Distillated Models with Different Truncation Strategies.** To more thoroughly evaluate the effectiveness of optimized student models under different truncation strategies, we further analyze both the response lengths and the quality of the generated reasoning trajectories produced by these optimized student models.

We first present the average lengths of reasoning trajectories in Figure 3(c). Among all models, UPFT produces significantly shorter trajectories,

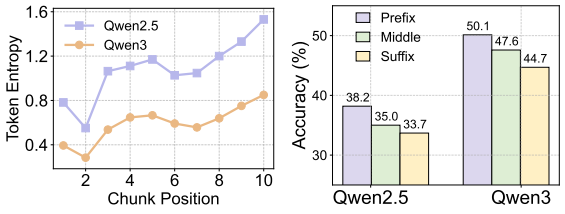



(a) AIME24&25. (b) MATH500.




(c) Response Length. (d) GLM Preference Rate.

Figure 3: Performance of Distilled Student Models under Different Prefix Truncation Strategies. We compare models distilled using fixed prefix truncation ratios and P-ALIGN (Figures 3(a) and 3(b)), analyze their response lengths (Figure 3(c)), and evaluate CoT quality with GLM-4.5 as the judge (Figure 3(d)).




(a) Chunk Uncertainty at Different Positions. (b) P-ALIGN Using Fixed Teacher-generated Prefix.

Figure 4: Effectiveness of Chunks in Long-Form CoTs at Different Positions. Figure 4(a) shows the entropy scores of student models across different chunks at varying positions. Figure 4(b) compares the performance of student models when using prefix, middle, or suffix segments as supervision.

indicating that naively using prefixes as supervision signals may cause the student model to overfit superficial patterns. Compared with SFT (Long-CoT), P-ALIGN generates more concise reasoning trajectories, demonstrating its effectiveness in mitigating the influence of redundant or uncertain reasoning content. We then employ GLM-4.5 (Figure 3(d)) to evaluate the quality of reasoning trajectories generated by different models, with detailed evaluation prompts provided in Appendix A.4. The evaluation results show that P-ALIGN achieves the highest win rate, further confirming its effectiveness in providing higher-quality supervision that enables the student model to produce more concise and higher-quality reasoning trajectories.

### 5.4 Effectiveness of Reasoning Trajectory Chunks at Different Positions

As illustrated in Figure 4, we analyze the effectiveness of teacher-generated reasoning trajectories in supervising student models, with a particular focus on evaluating the contributions of trajectory prefixes at different positions.

As shown in Figure 4(a), we assess the uncertainty of teacher-generated reasoning regions at different positions using the student model. Specifically, we evenly divide the teacher-generated reasoning trajectories into ten sequential chunks and compute the average token-level entropy score for chunks at each position. A higher entropy score indicates greater uncertainty (Kendall and Gal, 2017). The results show that entropy increases as the chunk position moves toward the rear of the trajectory, indicating that later reasoning steps contain more uncertain content and potential exploratory trails. This observation is consistent with previous work (Ji et al., 2025). Furthermore, as shown in Figure 4(b), we evenly partition the teacher-generated CoT into three segments: prefix, middle, and suffix, and directly use each segment to replace the adaptively truncated prefix in P-ALIGN for optimizing the student model. The results demonstrate that conditioning on prefixes yields substantially better performance than using either the middle or suffix segments, suggesting that prefixes provide more stable and informative context than later parts of the reasoning trajectory. This finding further validates the motivation of P-ALIGN to fully exploit prefixes for synthesizing higher-quality supervision, thereby more effectively guiding the student model through SFT.

## 6 Conclusion

This paper proposes the Adaptive **P**refix-**ALIGN**ment reasoning distillation method (P-ALIGN), which adaptively truncates prefixes from teacher-generated reasoning trajectories and optimizes student models to align with the retained teacher prefixes. Our experimental results demonstrate that P-ALIGN consistently outperforms all baselines across multiple mathematical reasoning benchmarks, highlighting its effectiveness and robustness in distillating long-form reasoning into student models.

## Acknowledgment

This work is partly supported by the National Natural Science Foundation of China (No. 62576082) and Alibaba Innovative Research Program. This work is also supported by the AI9Stars community.

## Limitation

Although P-ALIGN effectively constructs higher-quality SFT data, it still relies on powerful closed-source reasoning models to generate long-form reasoning chains, which incurs substantial computational overhead. Moreover, while adaptive prefix truncation proves effective, the self-judge process heavily depends on the judgment capability of the student model. This dependency may cause smaller-scale student models to become confused during self-judgment. Therefore, beyond self-information requirements, explicitly accounting for prefix quality is also crucial for adaptive prefix truncation.

## References

2025. AIME. aime problems and solutions. Accessed: September 23, 2025.

2025. AMC12. amc 12 problems and solutions. Accessed: September 23, 2025.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*, pages 1877–1901.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025a. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *ArXiv preprint*, abs/2503.09567.

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

Xinjie Chen, Minpeng Liao, Guoxin Chen, Chengxi Li, Biao Fu, Kai Fan, and Xinggao Liu. 2025b. From data-centric to sample-centric: Enhancing llm reasoning via progressive optimization. *ArXiv preprint*, abs/2507.06573.

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. *ArXiv preprint*, abs/2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, and Haowei Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv preprint*, abs/2501.12948.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. 2021. Monash time series forecasting archive. *ArXiv preprint*, abs/2105.06643.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *ArXiv preprint*, abs/2305.15717.

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. *ArXiv preprint*, abs/2103.03874.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Proceedings of ACL Findings*, pages 8003–8017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Ke Ji, Jiahao Xu, Tian Liang, Qiuzhi Liu, Zhiwei He, Xingyu Chen, Xiaoyuan Liu, Zhijie Wang, Junying Chen, Benyou Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. The first few tokens are all you need: An efficient and effective unsupervised prefix fine-tuning method for reasoning models. *ArXiv preprint*, abs/2503.02875.

Zhensheng Jin, Xinze Li, Yifan Ji, Chunyi Peng, Zhenghao Liu, Qi Shi, Yukun Yan, Shuo Wang, Furong Peng, and Ge Yu. 2025. ReCUT: Balancing reasoning length and accuracy in LLMs via stepwise trails and preference optimization. In *Proceedings of EMNLP Findings*, pages 14269–14282.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of NeurIPS*, pages 5574–5584.

Chengpeng Li, Zheng Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. 2023. Query and response augmentation cannot help out-of-domain math reasoning generalization. *ArXiv preprint*, abs/2310.05506.

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

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *ArXiv preprint*, abs/2308.09583.

Renjie Luo, Jiaxi Li, Chen Huang, and Wei Lu. 2025a. Through the valley: Path to effective long CoT training for small language models. In *Proceedings of EMNLP*, pages 4972–4992.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025b. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of EMNLP*, pages 20286–20332.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of EMNLP*, pages 2695–2709.

Yiliu Sun, Zicheng Zhao, Yang Wei, Yanfang Zhang, and Chen Gong. 2025. Well begun, half done: Reinforcement learning with prefix optimization for llm reasoning. *ArXiv preprint*, abs/2512.15274.

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. 2023. Making large language models better reasoners with alignment. *ArXiv preprint*, abs/2309.02144.

Renxi Wang, Haonan Li, Xudong Han, Yixuan Zhang, and Timothy Baldwin. 2024. Learning from failure: Integrating negative examples when fine-tuning large language models as agents. *ArXiv preprint*, abs/2402.11651.

Zihan Wang, Zihan Liang, Zhou Shao, Yufei Ma, Huangyu Dai, Ben Chen, Lingtao Mao, Chenyi Lei, Yuqing Ding, and Han Li. 2025. Infogain-rag: Boosting retrieval-augmented generation through document information gain-based reranking and filtering. In *Proceedings of EMNLP*, pages 4972–4992.

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 NeurIPS*, pages 24824–24837.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. *ArXiv preprint*, abs/2402.09739.

Zhuoyang Wu, Xinze Li, Zhenghao Liu, Yukun Yan, Zhiyuan Liu, Minghe Yu, Cheng Yang, Yu Gu, Ge Yu, and Maosong Sun. 2025. Enhancing long-chain reasoning distillation through error-aware self-reflection. *ArXiv preprint*, abs/2505.22131.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *ArXiv preprint*, abs/2502.18600.

An Yang, Anfeng Li, Baosong Yang, and et al. Beichen Zhang. 2025. Qwen3 technical report. *ArXiv preprint*, abs/2505.09388.

An Yang, Baosong Yang, Beichen Zhang, and Binyuan Hui. 2024. Qwen2.5 technical report. *ArXiv preprint*, abs/2412.15115.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *ArXiv preprint*, abs/2502.03387.

Huifeng Yin, Yu Zhao, Minghao Wu, Xuanfan Ni, Bo Zeng, Hao Wang, Tianqi Shi, Liangying Shao, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025. Marco-o1 v2: Towards widening the distillation bottleneck for reasoning models. *ArXiv preprint*, abs/2503.01461.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Proceedings of NeurIPS*, pages 15476–15488.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. 2025a. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *ArXiv preprint*, abs/2508.06471.

Wenhao Zeng, Yaoning Wang, Chao Hu, Yuling Shi, Chengcheng Wan, Hongyu Zhang, and Xiaodong Gu. 2025b. Pruning the unsurprising: Efficient code reasoning via first-token surprisal. *ArXiv preprint*, abs/2508.05988.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, and Xi Victoria Lin. 2022. Opt: Open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068.

Yongheng Zhang, Qiguang Chen, Jingxuan Zhou, Peng Wang, Jiasheng Si, Jin Wang, Wenpeng Lu, and Libo Qin. 2024. Wrong-of-thought: An integrated reasoning framework with multi-perspective verification and wrong information. In *Proceedings of EMNLP Findings*, pages 6644–6653.

| Dataset | Train | Test |
|---|---|---|
| s1K-1.1 (2025) | 1,000 | - |
| AIME25 (2025) | - | 30 |
| AIME24 (2025) | - | 30 |
| AMC12 (2025) | - | 83 |
| MATH500 (2021) | - | 500 |

Table 3: Data Statistics.

| Model | Long-CoT | Prefix | P-ALIGN |
|---|---|---|---|
| Qwen2.5-7B | 9,291 | 4,531 | 5,453 |
| Qwen3-8B | 9,291 | 4,402 | 5,732 |
| Llama3.2-3B | 8,755 | 4,780 | 6,035 |

Table 4: Average Token Length Statistics of Training Dataset Across Different Student Models. We evaluate the average token lengths of the original long-form CoTs, the truncated prefixes, and the final trajectories used by P-ALIGN.

# A Appendix

## A.1 License

We provide the licenses for the datasets used in P-ALIGN. s1K-1.1 is licensed under the MIT License, while MATH500, AIME24, AIME25, and AMC12 are licensed under the Apache License 2.0. All of these licenses permit the use of their data for academic purposes.

## A.2 Additional Implementation Details

In this section, we provide the detailed data statistics in P-ALIGN. We build our training set using 1,000 examples sampled from s1K-1.1 (Muennighoff et al., 2025) as the training dataset. For evaluation, we use the MATH500 with 500 examples, AIME25 with 30 examples, AIME24 with 30 examples, and AMC12 with 83 examples. All statistics are shown in Table 3.

To clarify the impact of P-ALIGN on supervision efficiency, we report average token length statistics across different student models in Table 4. Specifically, Long-CoT denotes the token length of the original teacher-generated reasoning traces, Prefix corresponds to the adaptive prefixes truncated by P-ALIGN, and P-ALIGN reports the token length of the final trajectories used for SFT. The results show that P-ALIGN substantially reduces the length of training data by retaining only concise and sufficient prefixes, which lowers training cost while maintaining or even improving model performance.

## A.3 Prompt Templates Used in P-ALIGN

We detail the instruction prompts employed at different stages of P-ALIGN. As shown in Figure 5, the $\text{Instruct}_{\text{QA}}$ prompts the model to directly answer the given question, serving as the standard question-answering setting and providing the initial reasoning. Building on this, the prefix evaluation stage utilizes the $\text{Instruct}_{\text{Eval}}$ prompt, as illustrated in Figure 6, which guides the student model in assessing whether a given reasoning prefix contains sufficient information to solve the problem. Based on the retained prefix, the prefix-based alignment stage applies the $\text{Instruct}_{\text{Align}}$ prompt shown in Figure 7, guiding the student model to continue generating a complete reasoning trajectory.

| Backbone Model | P-ALIGN | w/o Binary Search |
|---|---|---|
| Qwen2.5-7B | 7.4 | 144.6 |
| Qwen3-8B | 7.4 | 137.1 |
| Llama3.2-3B | 7.4 | 152.9 |

Table 5: Average Search Count for Binary Search Usage in Adaptive Prefix Truncation.

## A.4 Prompt Templates Used for Evaluating the Quality of CoTs

As shown in Figure 8, we present the prompt templates used to evaluate the quality of the CoTs synthesized by three methods: SFT (Long-CoT), UPFT, and P-ALIGN. The evaluation is conducted using GLM-4.5 (Zeng et al., 2025a) as the evaluator, which assesses each generated reasoning trace based on its logical completeness, reflective reasoning behavior, conciseness, and overall organization. We employ consistent prompt templates across all methods to ensure fair comparison and reproducibility.

## A.5 Verification of Binary Search Effectiveness in P-ALIGN

To further verify the effectiveness of binary search in improving the efficiency of adaptive prefix truncation, we measure the average search count for determining whether binary search is used in prefix truncation. When binary search is not used, the sufficient condition is checked in a sentence-level sequential manner. As shown in Table 5, the average search count without binary search is about 20 times higher than when binary search is used. This clearly demonstrates the effectiveness of binary search in improving efficiency. Notably, the total number of iterations for binary search is fixed at $O(\log_2 n)$, independent of both the student model and the results of the `ENOUGH` judgment.

| Method | AIME25 | | AIME24 | | AMC12 | | MATH500 | | Avg. | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 | Pass@1 | Pass@3 |
| **Qwen2.5-7B-Instruct** | | | | | | | | | | |
| P-ALIGN (Student Judge) | 16.67 | 26.67 | 16.67 | 26.67 | 49.40 | 63.86 | 75.80 | 85.20 | 39.64 | 50.60 |
| P-ALIGN (DeepSeek Judge) | 13.33 | 23.33 | 16.67 | 23.33 | 47.31 | 59.04 | 76.00 | 86.00 | 38.33 | 47.93 |
| P-ALIGN (GLM Judge) | 13.33 | 20.00 | 13.33 | 23.33 | 48.48 | 60.24 | 75.20 | 85.80 | 37.59 | 47.34 |
| **Qwen3-8B** | | | | | | | | | | |
| P-ALIGN (Student Judge) | 23.33 | 33.33 | 30.00 | 36.67 | 66.27 | 77.11 | 84.80 | 92.80 | 51.10 | 59.98 |
| P-ALIGN (DeepSeek Judge) | 20.00 | 26.67 | 23.33 | 33.33 | 65.71 | 76.82 | 85.20 | 93.00 | 48.56 | 57.46 |
| P-ALIGN (GLM Judge) | 20.00 | 30.00 | 26.67 | 30.00 | 64.30 | 74.28 | 85.00 | 92.80 | 48.99 | 56.77 |

Table 6: Judge Ablation for Prefix Truncation. Using the student model as the prefix sufficiency judge consistently outperforms using stronger teacher models (DeepSeek-R1, GLM) across student backbones and benchmarks.

### A.6 Justification of Student-Based Prefix Sufficiency Evaluation

To validate the design of the student-based prefix sufficiency evaluation in P-ALIGN, we conduct an ablation study by replacing the student judge with stronger teacher models (e.g., DeepSeek-R1 and GLM) during the binary-search truncation process. Although teacher models are more capable of evaluating reasoning completeness, their judgments may not align with student learnability, potentially retaining overly complex or redundant reasoning segments. As shown in Table 6, student-based judging consistently achieves superior downstream distillation performance, supporting the necessity of student-aligned prefix truncation.

### A.7 Analysis of Training Data Retention

To further analyze the impact of prefix alignment on training data construction, we report the size of the final training dataset after filtering. In P-ALIGN, the filtering step is applied to ensure supervision correctness rather than to perform trajectory-level data selection. As shown in Table 7, over 97% of the original training data are preserved across student backbones, indicating high data utilization. Unlike trajectory selection methods that discard entire reasoning chains, P-ALIGN operates within each trajectory to retain prefix segments that are most learnable for the student model, thereby improving supervision quality without sacrificing data efficiency.

### A.8 Evaluation with Additional Teacher Models

To further evaluate the generality of P-ALIGN under varying teacher–student capability gaps, we conduct additional experiments using two alternative teacher models, QwQ-32B and DeepSeek-R1-Distill-Qwen-7B. These teachers exhibit distinct reasoning styles and supervision characteristics compared to the primary teacher model. As shown in Table 8, P-ALIGN consistently improves student performance across all teacher settings and student backbones. The gains are comparable to those observed under DeepSeek-R1 supervision, demonstrating that the effectiveness of prefix alignment does not rely on a specific teacher model but generalizes across different teacher reasoning distributions.

| Student Model | Original Data | Final Training Data |
|---|---|---|
| Qwen2.5-7B | 1000 | 972 |
| Qwen3-8B | 1000 | 988 |

Table 7: Training Data Retention After Filtering. P-ALIGN preserves over 97% of the original data while ensuring supervision correctness.

### A.9 Additional Results with Different Student Models

To further evaluate the scalability and robustness of P-ALIGN, we conduct additional experiments on models of different sizes, including the smaller Llama3.2-3B-Instruct (Grattafiori et al., 2024) and the larger Qwen3-14B (Yang et al., 2025). All training settings remain consistent with those used for the 7B-scale experiments. As shown in Table 9, P-ALIGN yields consistent improvements across model scales, demonstrating that the proposed framework generalizes effectively to both lower-capacity and higher-capacity models.

### A.10 The Case of Binary Search for Minimal Sufficient Prefix Selection

As shown in Figure 9, P-ALIGN employs the binary search procedure for adaptive prefix evaluation by self-judging, which improves both the efficiency and precision of locating a minimal sufficient prefix. Specifically, when the self-judging evaluation returns `ENOUGH`, the current prefix already contains sufficient information to solve the problem, and we continue searching for a shorter

| Methods | AIME25 | AIME24 | AMC12 | MATH500 |
|---|---|---|---|---|
| **QwQ-32B** | | | | |
| **Qwen2.5-7B** | | | | |
| Zero-Shot | 6.67 | 10.00 | 40.76 | 69.80 |
| SFT (Long-CoT) | 10.00 | 10.00 | 47.64 | 73.80 |
| P-ALIGN | **16.67** | **16.67** | **49.40** | **75.80** |
| **Qwen3-8B** | | | | |
| Zero-Shot | 20.00 | 26.67 | 59.84 | 83.60 |
| SFT (Long-CoT) | 23.33 | 23.33 | 59.73 | 83.60 |
| P-ALIGN | **23.33** | **30.00** | **66.27** | **84.80** |
| **DeepSeek-R1-Distill-Qwen-7B** | | | | |
| **Qwen2.5-7B** | | | | |
| Zero-Shot | 6.67 | 10.00 | 40.76 | 69.80 |
| SFT (Long-CoT) | 6.67 | 13.33 | 42.17 | 71.20 |
| P-ALIGN | **16.67** | **16.67** | **49.40** | **75.20** |
| **Qwen3-8B** | | | | |
| Zero-Shot | 20.00 | 26.67 | 59.84 | 83.60 |
| SFT (Long-CoT) | 20.00 | 23.33 | 59.73 | 84.00 |
| P-ALIGN | **23.33** | **30.00** | **66.27** | **84.80** |

Table 8: Evaluation Across Additional Teacher Models with Varying Capability Gaps. P-ALIGN consistently improves student performance under different teacher supervision sources.

| Methods | AIME24 | AMC12 | MATH500 | Avg. |
|---|---|---|---|---|
| ***Llama3.2-3B*** | | | | |
| Zero-Shot | 6.67 | 18.81 | 39.80 | 21.76 |
| SFT (Long-CoT) | 6.67 | 20.19 | 43.60 | 23.49 |
| UPFT (2025) | 10.00 | 26.50 | 44.60 | 27.03 |
| P-ALIGN | **16.67** | **30.12** | **48.40** | **31.73** |
| ***Qwen3-14B*** | | | | |
| Zero-Shot | 33.33 | 72.29 | 87.40 | 64.34 |
| SFT | 36.67 | 73.26 | 88.60 | 66.18 |
| UPFT (2025) | 36.67 | **74.69** | **89.60** | 66.99 |
| P-ALIGN | **43.33** | 73.26 | 89.20 | **68.60** |

Table 9: Performance of P-ALIGN on Models of Different Sizes. We provide additional results on Llama3.2-3B and Qwen3-14B to evaluate the generalization of P-ALIGN across different parameter scales.

prefix to obtain a more concise and learnable supervision signal. Otherwise, if the evaluation returns `NOT_ENOUGH`, the current prefix is still insufficient, and a longer prefix with richer information is required to support solving the problem. In this case, P-ALIGN identifies the minimal sufficient prefix with only 6 self-evaluation iterations. In contrast, the sentence-by-sentence sequential scan (P-ALIGN w/o Binary Search) requires 12 evaluations to reach the same boundary. This comparison demonstrates that binary search substantially reduces the prefix-selection cost and avoids unnecessary evaluation overhead.

## A.11 Case Study

In this section, we present a detailed case study using two tables to illustrate the key steps and the qualitative benefits of P-ALIGN. As shown in Table 10, we illustrate the full pipeline of P-ALIGN, including the prefix truncation and prefix-based alignment process. Building on this, Table 11 compares the final response generated by P-ALIGN and baseline methods, highlighting the improved quality and clarity of P-ALIGN outputs.

First, we examine a representative mathematical problem to analyze the behavior of P-ALIGN. As shown in Table 10, the key reasoning that directly supports the correct answer("Therefore, $P \equiv 109 \pmod{125}$") appears early in the teacher's reasoning trajectory, within the first 30% of the full reasoning length. The subsequent steps often contain more reflective and exploratory reasoning, which can be less learnable for the student model. In this case, the vanilla student model initially fails and produces an incorrect result ("thus the remainder is 999"). In contrast, when conditioned on the retained prefix, the student can follow the preserved reasoning structure and complete the remaining reasoning steps, ultimately arriving at the correct solution ("Therefore, the remainder is boxed {109}."). This comparison highlights the advantage of prefix-aligned supervision in stabilizing the student's reasoning and helping it recover correct solutions that are otherwise difficult to obtain.

Then, we compare student outputs under different baselines on the same question. As shown in Table 11, SFT (Label) provides little explicit reasoning and directly outputs an incorrect answer. The vanilla model attempts step-by-step derivations, but fails to identify the key insight of applying Vieta's formula and instead attempts to solve the cubic equation explicitly, which is error-prone due to heavy algebraic computation. Furthermore, SFT (Long-CoT) performs a deeper analysis and recognizes the crucial insight early ("That reminds me of Vieta's formula"), realizing that the problem can be solved without computing each root. However, excessive reflection in later steps still leads to a calculation error, despite having found the correct approach. Compared with these baselines, P-ALIGN distills the reasoning patterns through prefix-based supervision, enabling the student to follow the correct high-level structure without overthinking. As a result, it produces a cleaner solution trajectory and reaches the correct final answer.

**Instruction for Direct Question Answering**

Please reason step by step, and put your final answer within boxed{}.

*Problem*: {Problem}

Figure 5: The Prompt Templates Used for Direct Question Answering.

**Instruction for Prefix Evaluation by Self-Judging**

You are a reasoning evaluator.
Your task is to judge whether the prefix already includes the essential logical and computational steps required to confidently reach the correct answer easily.

- Reply "[ENOUGH]" if the reasoning already contains the key logic and main transformations, and the remaining work is mostly routine.
- Reply "[NOT_ENOUGH]" if any essential step or reasoning link is still missing, making it uncertain to reach the correct answer.

Reply only with one word: either [ENOUGH] or [NOT_ENOUGH].
Do not add any explanation.

Question: {question}

Prefix: {prefix}

Figure 6: The Prompt Templates Used for Prefix Evaluation.

**Instruction for Prefix-based Alignment**

Please continue from the prior knowledge and solve the problem step by step, and put your final answer within \\boxed{}.

I will provide you with the prefix as the prior knowledge to assist you in solving the question.

Question: {question}

Prefix: {prefix}

Figure 7: The Prompt Templates Used for Prefix-based Alignment.

**Instruction for GLM-based Evaluation**

You will be given one question and four reasoning chains A / B / C .

All of them reach a correct answer, but the reasoning quality differs. Your task is to evaluate the three reasoning processes and select the best one.

Evaluation Criteria:
1.The reasoning should be complete and logically structured.
2.It should include reflective or self-checking reasoning.
3.No redundant or repetitive thinking steps.
4.Reasoning should be concise, efficient, and well-organized.

Question: {question}

Chain A: {a}  Chain B: {b}  Chain C:{c}
Only output the final choice without explanations, in the following format:
Best Reasoning Chain: {A/B/C/}

Figure 8: The Prompt Templates Used for GLM-based Evaluation.

**The Case of Binary Search for Adaptive Prefix Truncation**

[Starting Binary Search for Minimal Sufficient Prefix]
Total sentences: 48
Initial search interval: [1, 48]
==========

[Round 1 Evaluation]
Current search interval: left=1, right=48
Prefix sentences checked: 24/48
Model output: [ENOUGH]
Action: Current prefix is sufficient → Try shorter prefix

[Round 2 Evaluation]
Current search interval: left=1, right=23
Prefix sentences checked: 12/48
Model output: [ENOUGH]
Action: Current prefix is sufficient → Try shorter prefix

[Round 3 Evaluation]
Current search interval: left=1, right=11
Prefix sentences checked: 6/48
Model output: [NOT_ENOUGH]
Action: Current prefix is insufficient → Increase prefix length

[Round 4 Evaluation]
Current search interval: left=7, right=11
Prefix sentences checked: 9/48
Model output: [NOT_ENOUGH]
Action: Current prefix is insufficient → Increase prefix length

[Round 5 Evaluation]
Current search interval: left=10, right=11
Prefix sentences checked: 10/48
Model output: [NOT_ENOUGH]
Action: Current prefix is insufficient → Increase prefix length

[Round 6 Evaluation]
Current search interval: left=11, right=11
Prefix sentences checked: 11/48
Model output: [NOT_ENOUGH]
Action: Current prefix is insufficient → Increase prefix length

---
[Binary Search Completed]
Minimal sufficient prefix length: 12/48
Prefix proportion: 0.2500

Figure 9: The Case of Adaptive Prefix Truncation via Binary Search.

**Case:** The Case of Prefix-Alignment Distillation in P-ALIGN

**Question:** Find the remainder when $9 \times 99 \times 999 \times \cdots \times \underbrace{99\cdots9}_{999\ 9\text{'s}}$ is divided by 1000.

**Ground Truth:** 109

**Vanilla Generation:** To find the remainder of the question... we can use properties of modular arithmetic and the Chinese Remainder Theorem... Each number in the product is of the form $10^k - 1$. For any $n \geq 3$, $10^n \equiv 0 \pmod 8$, hence $10^n - 1 \equiv -1 \pmod 8$. Thus the product satisfies $9 \times 99 \times 999 \times \cdots \equiv (-1)^{999} \equiv -1 \pmod 8$... Similarly, since each $10^k - 1$ for $k \geq 3$ ends in 999, the product modulo 125 is claimed to be $9 \times 99 \times 999 \times \cdots \equiv (-1)^{999} \equiv -1 \pmod{125}$. ... and thus the remainder is 999.

**Original Long-Form Reasoning:** Alright, so I have this problem here: I need to find the remainder when the product of numbers like $9, 99, 999$, and so on, up to a number with 999 nines, is divided by 1000. Hmm, okay, let's... Therefore, the product is $\prod_{k=1}^{999}(10^k - 1)$... Since $10 \equiv 2 \pmod 8$, so $10^k \pmod 8$ is $2^k \pmod 8$... All terms from $k = 3$ onward are $-1 \pmod 8$.Therefore, $P \equiv 5 \pmod 8$... Therefore, $(10^k - 1) \equiv -1 \pmod{125}$ for $k \geq 3$ ... **Therefore,** $P \equiv 109 \pmod{125}$**...**

First, the problem is asking for the remainder of a very large product divided by 1000. When dealing with remainders, especially modulo 1000, it often helps to look at the problem modulo 8 and modulo 125 separately, since 1000 is 8 * 125 and 8 and 125 are coprime. Then, using the Chinese Remainder Theorem, combine the results... A number consisting of n 9s can be written as 10^n - 1. ... The number with k 9s is 10^k - 1. Therefore, the product is $\Pi_{k=1}^{999}(10^k - 1)$. ... We need to compute $P \bmod 1000$... Compute $P \bmod 8$. Since $10 \equiv 2 \pmod 8$, 10^k (mod 8) is 2^k (mod 8), so each term (10^k - 1) (mod 8) is (2^k - 1) (mod 8). ... For k=1: (2^1 - 1) = 1 (mod 8). For k=2: (4 - 1) = 3 (mod 8). For $k \geq 3$: 2^k $\equiv 0 \pmod 8$, so (2^k - 1) $\equiv$ -1 (mod 8). ... Therefore, $P \equiv 5 \pmod 8$...

Alright, now onto modulo 125. ... 10^3 = 1000 $\equiv 0 \pmod{125}$. Therefore, for all k $\geq 3$, 10^k $\equiv 0 \pmod{125}$, so (10^k - 1) $\equiv$ -1 (mod 125). ... Thus, $P \equiv 9 \cdot 99 \cdot (-1)^{997} \pmod{125}$. ... 9*99 = 891, and 891 (mod 125) = 16, hence $P \equiv -16 \equiv 109 \pmod{125}$... So summary so far: $P \equiv 5 \pmod 8$ and $P \equiv 109 \pmod{125}$. Now need to find x such that $x \equiv 5 \pmod 8$ and $x \equiv 109 \pmod{125}$. ... Let x = 125m + 109. Then 125m + 109 $\equiv 5 \pmod 8$. ... This gives 5m $\equiv 0 \pmod 8$, so m $\equiv 0 \pmod 8$. Therefore, x = 1000n + 109, hence $x \equiv 109 \pmod{1000}$...

...

**Therefore, the boxed answer is boxed{109}.**

**Truncation Prefix:** Alright, so I have this problem here: I need to find the remainder when the product of numbers like $9, 99, 999$, and so on, up to a number with 999 nines, is divided by 1000. Hmm, okay, let's... Therefore, the product is $\prod_{k=1}^{999}(10^k - 1)$... Since $10 \equiv 2 \pmod 8$, so $10^k \pmod 8$ is $2^k \pmod 8$... All terms from $k = 3$ onward are $-1 \pmod 8$.Therefore, $P \equiv 5 \pmod 8$... Therefore, $(10^k - 1) \equiv -1 \pmod{125}$ for $k \geq 3$ ... **Therefore,** $P \equiv 109 \pmod{125}$.

**Prefix-based Alignment:** To continue answering the question based on existing prior knowledge ... First, consider the product modulo 8... the product modulo 8 is $1 \cdot 3 \cdot (-1)^{997} \equiv -3 \equiv 5 \pmod 8$... Next, consider the product modulo 125... the product modulo 125 is $9 \cdot 99 \cdot (-1)^{997} \equiv -891 \equiv 16 \cdot (-1) \equiv 109 \pmod{125}$... Finally, by the Chinese Remainder Theorem, solving $x \equiv 5 \pmod 8$ and $x \equiv 109 \pmod{125}$ yields $x = 125m + 109$ with $m \equiv 0 \pmod 8$, hence $x \equiv 109 \pmod{1000}$. **Therefore, the remainder is boxed{109}.**

**Truncation Ratio:** 0.27

Table 10: Case Study. Text in **red** highlights erroneous parts in the reasoning and their underlying causes, while text in **green** denotes correct reasoning content, and text in **blue** indicates the truncation point of the informative prefix.

**Case:** The Case of Outputs Under Different Distillation Methods

**Question:** The roots of the polynomial $10x^3 - 39x^2 + 29x - 6$ are the height, length, and width of a rectangular box (right rectangular prism). A new rectangular box is formed by lengthening each edge of the original box by 2. What is the volume of the new box?

**Ground Truth:** 30

**Vanilla Response:** To solve the problem, we will follow these steps: first, find the roots of the polynomial $10x^3 - 39x^2 + 29x - 6$, then determine the dimensions of the original box, and finally compute the volume of the new box. Using the Rational Root Theorem, it is claimed that $x = 2$ is a root, and the polynomial is factored as $(x - 2)(10x^2 - 19x + 3)$ ... The resulting roots are $x = 2$, $x = \frac{19+\sqrt{241}}{20}$, and $x = \frac{19-\sqrt{241}}{20}$. Each dimension is increased by 2, leading to a new volume computed as $4 \cdot \frac{59+\sqrt{241}}{20} \cdot \frac{59-\sqrt{241}}{20} = 32.4$ ... **Therefore, the final answer is boxed{32.4}**.

**SFT (Label):** **Thus, the volume of the new box is {37.5}**.

**SFT (Long-CoT):** Alright, so I have this problem here where I need to find the volume of a new rectangular box formed by increasing each edge of an original box by 2 units. The original box's dimensions are the roots of the polynomial $10x^3 - 39x^2 + 29x - 6$. Hmm, okay. Let me break this down step by step ... **That reminds me of Vieta's formulas. For $10x^3 - 39x^2 + 29x - 6$, we have $h + l + w = \frac{39}{10}$, $hl + hw + lw = \frac{29}{10}$, and $hlw = \frac{3}{5}$**... Now expand the volume: $(h + 2)(l + 2)(w + 2) = hlw + 2(hl + hw + lw) + 4(h + l + w) + 8$ ...
At this point, I want to double-check the arithmetic to avoid mistakes. Converting everything to a common denominator, I rewrite 8 as $\frac{80}{10}$ and sum the terms, **obtaining** $\frac{58+156+80}{10}$... Wait, let me verify this again to be safe.... After rechecking, the total is interpreted as $\frac{294}{10}$, which simplifies to 29.4. The steps seem consistent after revision... **So the final answer is boxed{29.4}**.

**P-ALIGN:** <Begin_of_prefix> Okay, so I need to find the volume of a new rectangular box formed by increasing each edge of an original box by 2 units. The original box's dimensions (length, width, height) are the roots of the polynomial $10x^3 - 39x^2 + 29x - 6$. First, I need to find those roots, or at least use their relationships. **I remember Vieta's formulas: for a cubic equation $ax^3 + bx^2 + cx + d = 0$, the sum of the roots is $-b/a$, the sum of pairwise products is $c/a$, and the product of the roots is $-d/a$...** <End_of_prefix>
To continue solving the question based on the existing prior knowledge ... Therefore, the roots $(L, W, H)$ satisfy $L + W + H = \frac{39}{10}$, $LW + WH + HL = \frac{29}{10}$, and $LWH = \frac{3}{5}$. The new volume is $(L + 2)(W + 2)(H + 2)$, which expands to $LWH + 2(LW + WH + HL) + 4(L + W + H) + 8$. Substituting the values gives $\frac{3}{5} + 2 \cdot \frac{29}{10} + 4 \cdot \frac{39}{10} + 8 = 30$. ... **Therefore, the new volume is boxed{30}**.

Table 11: Case Study. Text in **red** highlights erroneous parts in the reasoning and their underlying causes, text in **green** denotes correct reasoning content.