DRP: Distilled Reasoning Pruning with Mathematical Skill-aware Step Decomposition for Efficient Large Reasoning Models

Yuxuan Jiang[1] **Dawei Li**[2] **Francis Ferraro**[1]
[1]University of Maryland, Baltimore County
[2]Arizona State University
yuxuanj1@umbc.edu

Abstract

While Large Reasoning Models (LRMs) excel at complex tasks via long Chain-of-Thought (CoT) reasoning, their outputs are often excessively verbose, leading to inefficiency. This problem is amplified when the student's long-form reasoning mismatches the concise outputs of smaller teacher models—common in LLM distillation to avoid using costly large teachers. To address this issue, we propose **Distilled Reasoning Pruning (DRP)**, a hybrid framework that combines inference-time pruning with tuning-based distillation. DRP adopts a trajectory-aware distillation paradigm, in which the teacher operates directly on the student's reasoning paths by performing *skill-aware step decomposition* and pruning. The resulting refined trajectories are then distilled back into the student model, providing targeted supervision that aligns closely with the student's policy and facilitates more effective learning. This trajectory-aware design anticipates recent advances in On-Policy Distillation, as it grounds supervision in the student's own reasoning trajectories while retaining the efficiency of offline distillation. Across challenging math datasets, DRP significantly reduces token usage without sacrificing accuracy—for instance, cutting tokens on GSM8K from 917 to 328 while improving accuracy from 91.7% to 94.1%, and reducing AIME tokens by 43% with no performance drop. Further analysis shows that aligning training CoT structure with the student's capacity is key to effective knowledge transfer.

Code is available at:
https://github.com/YuxuanJiang1/DRP

1 Introduction

Although Large Reasoning Models (LRMs) (Xu et al., 2025a), like OpenAI's o1 (OpenAI, 2024b) and DeepSeek-R1 (Guo et al., 2025), have advanced the state of the art in complex reasoning tasks (Li et al., 2025e), a critical limitation of these models is their tendency toward *overthinking*—the generation of excessively verbose reasoning trajectories containing redundant or unnecessary steps (Chen et al., 2024b; Cui et al., 2025; Fu et al., 2024). This can lead to substantial inference overhead and misguide the model toward incorrect conclusions (Sui et al., 2025).

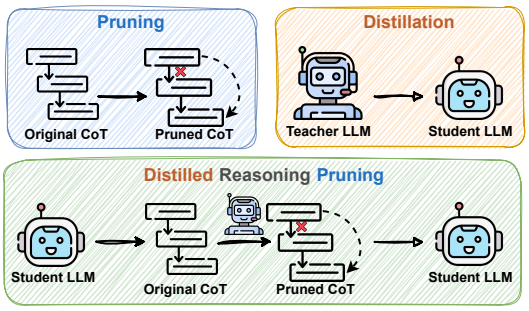

Figure 1: An overview of our proposed Distilled Reasoning Pruning (DRP) framework, which unifies pruning and distillation. Unlike traditional distillation, DRP uses a teacher LLM to prune the student model's Long-CoT reasoning chains into concise CoTs, which are then distilled back into the student. This design addresses the reasoning style mismatch between verbose student models and concise teacher models, improving efficiency without sacrificing accuracy.

Existing solutions predominantly follow two paradigms: *inference-time pruning*, which attempts to terminate generation early to avoid redundant reasoning steps (Cui et al., 2025; Muennighoff et al., 2025; Fu et al., 2024; Zeng et al., 2025c), and *distillation-based compression*, where smaller models are trained on teacher-generated reasoning paths to imitate the concise reasoning behavior of larger models (Tan et al., 2024; Zhu et al., 2024; Xu et al., 2025b). However, both approaches can be at the cost of accuracy: pruning methods risk prematurely halting the reasoning process, while distillation methods tend to underperform due to the learnability gap (Xu et al., 2024; Li et al., 2025a). This gap

Findings of the Association for Computational Linguistics: ACL 2026, pages 4020–4039

becomes especially pronounced when the teacher adopts a Short-CoT style (concise, polished reasoning), and the student follows a Long-CoT style (verbose reasoning with reflective self-corrections). Such style mismatch introduces a compatibility issue that hinders effective learning, as discussed in prior work (Xu et al., 2024).

To address these limitations, we propose **Distilled Reasoning Pruning (DRP)** (Fig. 1), a hybrid framework that combines the strengths of both pruning and distillation (student/teacher) paradigms. In particular, rather than simply distilling down a response from the teacher model, we use the teacher model to prune an initial, lengthy, CoT reasoning trajectory from the student model. To facilitate this, and to encourage shorter yet informative resulting pruned trajectories, we introduce a **skill-based step decomposition** method. The teacher model uses this to prune the trajectory, which produces more stable and semantically coherent reasoning units. Unlike prior methods that rely solely on either teacher-generated (Xu et al., 2025b; Zhu et al., 2024) or self-sampled (Chen et al., 2024b; Ma et al., 2025) concise trajectories for distillation, DRP takes advantage of the teacher's pruning within the student model's original reasoning structure. This trajectory-aware design is conceptually related to recent advances in on-policy distillation, as supervision is grounded in the student's own reasoning paths. However, DRP operates in an offline distillation setting, avoiding the computational overhead of fully on-policy training while still aligning teacher feedback with the student's policy. This design reduces the learnability gap and enables student models to achieve efficient reasoning without compromising performance.

Consider the example in Fig. 2. For this word problem, the student model generates its initial CoT trajectory (shown in the `<think>` block). This trajectory is provided to a teacher model, which segments that CoT into steps, with a high-level description of the *skill* that that step demonstrates or exercises. The teacher model then prunes and curates these skill-segmented steps, such as by merging similar ones (if the student was redundant or verbose) or deleting steps (e.g., for backtracking). This pruned CoT trajectory is then provided to the student model for supervised fine-tuning.

We conduct extensive experiments across multiple student models, teacher models, and mathematical benchmarks. DRP consistently improves both reasoning efficiency and accuracy, even on substantially harder out-of-distribution tasks. Our ablation studies show that DRP significantly outperforms direct distillation, whose aggressive compression often harms generalization. Moreover, DRP remains effective across diverse teacher choices, including relatively weaker models, while higher teacher strength or similarity alone does not guarantee larger gains. These findings highlight that effective pruning depends on preserving and utilizing reasoning structure rather than on raw teacher capability. Our key contributions are summarized as follows:

- We propose **Distilled Reasoning Pruning (DRP)**, a unified framework that combines step-level pruning and distillation to improve both reasoning efficiency and accuracy in small-scale LLMs.
- We introduce a **skill-based step decomposition** that segments reasoning traces into semantically coherent units, providing a stable foundation for effective pruning.
- Through systematic ablations, we show that **effective pruning is driven by preserving meaningful reasoning structure rather than finer granularity, aggressive compression, or stronger teachers alone**, clarifying the robustness and operating boundaries of DRP.

2 Related Work

2.1 The Overthinking Problem

Large reasoning models , such as OpenAI o1 (OpenAI, 2024b) and DeepSeek-R1 (Guo et al., 2025), are a subclass of LLMs trained to iteratively generate and refine intermediate steps (Chen et al., 2025a; Sui et al., 2025; Jiang and Ferraro, 2026b), internalizing Chain-of-Thought (CoT) reasoning (Tong et al., 2024). This leads to strong performance on complex tasks in math and code. However, LRMs—especially smaller ones—often overgenerate verbose reasoning chains with unnecessary tracebacks and redundant paths (Chen et al., 2024b; Cui et al., 2025; Fu et al., 2025), increasing token usage and causing reasoning drift that may harm accuracy (Hou et al., 2025; Sui et al., 2025).

Our method addresses this *overthinking* by pruning redundant steps and promoting concise, effective reasoning.

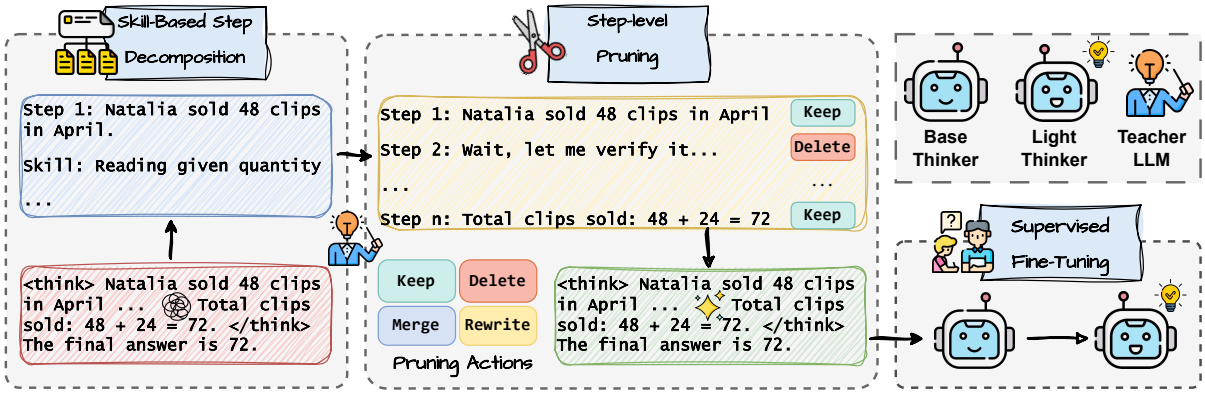

Figure 2: Overview of DRP framework. The student model generates Long-CoT reasoning traces, which are first decomposed into skill-based steps, then pruned and refined with help from a teacher LLM (e.g., GPT-4o). The concise CoTs align better with the student's learning, improving efficiency without sacrificing accuracy.

2.2 Token-Efficient Reasoning Methods

Current token-efficient reasoning methods fall into three categories (Sui et al., 2025; Jiang et al., 2025):

(1) **Prompt-based methods** constrain token budgets at the prompt level to encourage brevity without retraining. TALE (Han et al., 2024), for example, estimates per-instance budgets to reduce output length. However, these methods depend on handcrafted prompts and struggle with complex tasks. (2) **Supervised fine-tuning (SFT)** trains models on compressed traces to internalize efficiency (Xia et al., 2025; Munkhbat et al., 2025), as in CoT–Valve (Ma et al., 2025). These methods typically require task-specific data and retraining. (3) **Reinforcement learning (RL)** introduces rewards to penalize long outputs (Team et al., 2025; Chen et al., 2024b), sometimes with early-exit mechanisms (Muennighoff et al., 2025; Fu et al., 2024; Zeng et al., 2025c; Dai et al., 2025; Yang et al., 2025). ThinkPrune (Hou et al., 2025) sets target lengths and tightens constraints over time.

Despite these advances, many approaches reduce accuracy—especially on out-of-domain (OOD) tasks—when optimizing for brevity. By contrast, our framework uses external teacher models to perform *skill-aware pruning*, reducing tokens while improving robustness under distribution shift.

2.3 LLM Self-Refinement

Recent work explores *self-refinement*, where a model iteratively revises its own outputs to improve accuracy (Madaan et al., 2023; Li et al., 2024b; Jiang and Ferraro, 2026b), often through self-feedback or selective re-generation.

In contrast, our approach introduces external teacher supervision to prune redundant steps during reasoning. Rather than focusing on iterative correction of final answers, our *skill-aware pruning* directly optimizes the reasoning structure, thereby improving both token efficiency and reasoning quality.

3 Methodology

We propose **D**istilled **R**easoning **P**runing (DRP), a method to improve the efficiency of Long-CoT student models by refining their reasoning traces using a concise Short-CoT teacher. This asymmetric setup is key: the student (e.g., R1-Distill-Qwen-7B) generates verbose, reflective reasoning, while the teacher (e.g., GPT-4o) produces polished, compact CoTs. To bridge this gap, we let the student generate initial traces, which are selectively revised under teacher guidance.

As shown in Figure 2, DRP involves three stages: (1) decomposing reasoning into fine-grained, skill-based steps; (2) pruning and rewriting via a teacher LLM; and (3) fine-tuning the student on the revised traces. This process yields token-efficient supervision that enhances both accuracy and reasoning efficiency.

3.1 Mathematical Problem-solving Skill-Based Step Decomposition

To solve a math problem, a reasoning model produces a response consisting of two parts: a structured reasoning trace T enclosed in `<think>` `</think>` tags, and a final answer summarization A. We denote the response as $R = (T, A)$. For example, in Fig. 2, the trace may include "Natalia sold 48 clips in ... Total clips sold: 48 + 24

= 72," while the answer summarization is "The answer is 72."

We extract the trace T and prompt a teacher model to decompose it into non-overlapping segments, each aligned with a functional mathematical problem-solving skill (e.g., arithmetic, comparison, logical inference). This skill-based segmentation supports step-level pruning and distillation. Compared with naïve sentence splitting, it yields more stable step boundaries, preserves semantic coherence, and provides finer-grained supervision signals (see §6.1). The segmentation quality is validated through pairwise evaluation with Gemini 2.0 Flash; results and examples appear in Appendix G.

Formally, given a response $R = (T, A)$, we define a decomposition function D:

$$D(T) \mapsto \{(s_1, k_1), (s_2, k_2), \ldots, (s_m, k_m)\},$$

where each s_i is a contiguous token span representing one reasoning step, and k_i is the corresponding skill label assigned by the teacher model. For instance, "Natalia sold 48 clips in April" corresponds to the skill "Reading given quantity." Skills also cover broader tasks such as "Algebraic representation" or "Interpreting fractions of a subset." Complete decomposition prompts are provided in Appendix E.

3.2 Step-Level Pruning

For each reasoning step–skill pair (s_i, k_i), the teacher model is prompted to revise the step without altering the essential structure or logical intent of the original reasoning. The teacher selects one of the following actions for each step:

- **Keep:** Retain the step unchanged.
- **Delete:** Remove the step if it is redundant or uninformative.
- **Rewrite:** Replace the step with a more concise version conveying the same logic.
- **Merge:** Combine the step with adjacent ones if they form a coherent atomic unit.

This yields a revised step $\hat{s}_i = \text{Revise}(s_i)$, and a pruned reasoning trace: $\hat{T} = \{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{m'}\}$, where $m' \leq m$, which reduces redundancy and increases the overall information density.

Finally, the teacher model rewrites $\hat{T}$ into a fluent, coherent reasoning trace that preserves the tone and speaking style of the student model. To ensure consistency between the revised reasoning and the final answer, we prompt the teacher to optionally revise the original answer segment A, yielding an updated final answer summarization $\hat{A}$. The final output becomes a concatenation $\hat{R} = (\hat{T}, \hat{A})$, which we use as the target for supervised fine-tuning. Complete prompt are in Appendix E.

Self-Revision Prompt

Given a list of reasoning steps labeled with their respective skills, your task is to evaluate and revise each step according to one of the four ACTIONs.
Ensure the resulting reasoning path is concise, fluent, and logically sound. At the end, synthesize the revised steps into a coherent explanation that matches the speaker's tone.

3.3 Supervised Fine-Tuning

We construct the training dataset using pairs $(x, \hat{R})$, where x is the input question and $\hat{R}$ is the complete revised response.

We fine-tune the model using teacher-forced decoding to encourage the generation of concise reasoning traces. The training objective minimizes the negative log-likelihood of the revised response:

$$\mathcal{L}_{\text{SFT}} = -\sum_{i=1}^{n} \log P_\theta(y_i \mid x, y_{<i}),$$

where $\{y_1, \ldots, y_n\}$ are the tokens in $\hat{R}$, and θ denotes the model parameters. This supervision enables the model to internalize skill-aligned, token-efficient reasoning strategies while maintaining consistency with the final answer.

4 Experimental Setup

4.1 Datasets

Our training corpora consist of the training split of GSM8K (Cobbe et al., 2021) and the full PRM12K (Lightman et al., 2023) dataset. From these, we generate initial reasoning paths, which are then processed through skill-based step decomposition and teacher-guided step-level pruning to create supervision signals for fine-tuning. To evaluate complex mathematical reasoning and generalization ability, we select a broad set of out-of-domain benchmarks including MATH500 (Hendrycks et al., 2021), AIME24 (AI-MO Team, 2024a), and AMC23 (AI-MO Team, 2024b).

Method	GSM8K		MATH500 (OOD)		AIME24 (OOD)		AMC (OOD)	
	Pass@1	#Tokens	Pass@1	#Tokens	Pass@1	#Tokens	Pass@1	#Tokens
R1-Distill-Qwen-1.5B								
Base	70.7%	1443	80.4%	3276	6/30	10484	23/40	6516
+TALE	70.1%	1170	76.2%	3107	6/30	8915	22/40	6235
+Cot Valve	70.4%	805	76.5%	2705	7/30	**5601**	22/40	4574
+Thinkprune	80.0%	**712**	79.2%	**2006**	9/30	5745	25/40	**3291**
+DRP	**83.4%**	721 (-50%)	**82.0%**	2122 (-35%)	**10/30**	6135 (-42%)	**27/40**	3657 (-44%)
R1-Distill-Qwen-7B								
Base	91.7%	917	92.4%	2486	15/30	8674	31/40	4845
+TALE	91.0%	522	91.6%	2530	10/30	8602	31/40	3998
+Cot Valve	90.8%	364	89.4%	1975	13/30	6315	30/40	**3157**
+DRP	**94.1%**	**328** (-64%)	**93.0%**	**1781** (-28%)	**15/30**	**4966** (-43%)	**33/40**	3258 (-33%)

Table 1: Pass@1 accuracy and average token usage on R1-Distill-Qwen models across various math benchmarks, comparing our DRP method with Cot Valve (Ma et al., 2025), TALE (Han et al., 2024), and ThinkPrune (Hou et al., 2025).

4.2 Models

We use DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) as student models for supervised fine-tuning. Both are distilled variants of DeepSeek-R1 optimized for efficient inference. Our primary teacher model is GPT-4o (OpenAI, 2024a), which performs step-level decomposition and pruning to generate supervision signals. For ablation studies, we additionally explore alternative teacher models, including Gemini 2.0 Flash (DeepMind, 2025), Deepseek V3 (Liu et al., 2024) and ChatGPT (OpenAI, 2022).

4.3 Compared Methods

We select three representative methods that span the major paradigms for token-efficient reasoning: (1) **TALE** (Han et al., 2024): a prompt-based method incorporates a soft token budget constraint into the prompt to encourage concise generation. (2) **CoT-Valve** (Ma et al., 2025): a SFT-based method which enerates multiple chains-of-thought of varying lengths for the same problem, and performs supervised fine-tuning in multiple rounds—each time using shorter CoTs as training targets. (3) **ThinkPrune** (Hou et al., 2025): a tuning-based method uses reinforcement learning to iteratively reduce chain-of-thought length by optimizing under a target token constraint.

4.4 Implementation Details

Supervised fine-tuning is performed using the LLaMA-Factory[1] framework with LoRA adaptation. All models are trained for 3 epochs with cosine learning rate scheduling. Full hyperparameters are listed in Appendix B.

4.5 Evaluation Protocol

We use the lm-evaluate-harness[2] framework for unified evaluation across tasks. Each model is evaluated in a zero-shot setting. For inference, we use the vLLM backend with the maximum generation length set to 131,072 tokens—the upper limit supported by the Qwen models.

4.6 Evaluation Metrics

We report **Pass@1** as the accuracy metric, averaged over five independent runs to account for randomness in decoding. For efficiency, we measure the number of reasoning tokens generated per completion using the HuggingFace-compatible Qwen tokenizer.[3]

We also observe that models occasionally fall into degenerate loops, repeatedly generating parts of their responses until reaching the maximum generation limit (e.g., 130k tokens), far exceeding the typical average length (e.g., 5k). In most of these cases, the model fails to answer correctly. Such outliers significantly inflate the average token count.

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

[2] https://github.com/EleutherAI/lm-evaluation-harness

[3] https://huggingface.co/Qwen/Qwen-tokenizer

To mitigate their impact, we set a cutoff threshold of 12k tokens, which empirically covers 99% of correct responses across all benchmarks for the models we evaluate. Detailed token length distributions by task are provided in Appendix A.

5 Main Results

5.1 Accuracy and Token Efficiency

Table 1 presents the main results on our DRP and other compared methods. Our key findings are:

DRP consistently reduces token usage across all benchmarks and model sizes. Our proposed DRP method achieves substantial reductions in average token usage on both in-domain and out-of-domain tasks. On GSM8K, DRP reduces token count by up to 64% with the 7B model. For out-of-domain datasets, DRP yields 28%–44% reductions across all benchmarks, demonstrating strong generalization beyond the training distribution.

DRP improve the accuracy by mitigating the over-thinking problem in LRMs. Despite significantly reducing token usage, DRP preserves or improves Pass@1 accuracy on nearly all benchmarks. Notably, DRP improves accuracy even on harder datasets such as AMC and MATH500. The only exception is AIME24 under the 7B setting, where accuracy remains unchanged, suggesting the inherent difficulty of this benchmark.

Accuracy gains are more pronounced for smaller models. DRP delivers particularly strong improvements with the 1.5B model. On GSM8K, accuracy increases by 12.7%, while on AIME24 and AMC, DRP answers 4 more problems correctly compared to the base model. This indicates DRP's effectiveness in compensating for limited model capacity through more efficient supervision.

5.2 Comparison to Prior Work

We compare our approach with three representative baselines covering prompt-based, SFT-based, and RL-based token-efficient reasoning strategies (details can be found in Section 4.3).

Prompt-based methods offer limited control on Complex Reasoning Tasks. TALE is simple to implement and demonstrates moderate effectiveness on short-answer tasks such as GSM8K. It achieves token reductions with minimal accuracy degradation (less than 1%) on both model sizes. However, its effectiveness diminishes significantly on more challenging benchmarks. For example, on AIME24 with the 7B model, TALE causes a notable performance drop. Overall, TALE's prompt constraints provide limited control over fine-grained reasoning behaviors, which we hypothesize leads to its suboptimal performance on tasks requiring deeper or longer reasoning chains. In contrast, our DRP method remains effective even on such difficult tasks.

Existing SFT Methods struggle to balance accuracy and efficiency. CoT-Valve achieves consistent token reductions across benchmarks by training on compressed CoT. However, this often comes with accuracy trade-offs. For instance, on MATH500 and AMC, we observe accuracy degradation despite reduced token usage. By comparison, our DRP method achieves both stronger compression and accuracy improvements across most benchmarks. Notably, on AMC (1.5B), DRP improves accuracy from 22/40 to 27/40 while also reducing tokens from 4574 to 3657.

Fine-grained supervised pruning improves reasoning accuracy more effectively than RL-based methods. ThinkPrune demonstrates strong performance on the 1.5B model, suggesting that pruning is effective in eliminating distracting or redundant reasoning steps, particularly for smaller-capacity models. Notably, it achieves solid gains on benchmarks like AIME24 and AMC. However, our DRP method yields higher accuracy improvements, likely due to its use of high-quality teacher-guided pruning. While ThinkPrune achieves slightly better compression on a few tasks—due to its explicit optimization for token length, DRP achieves notably higher accuracy across the board, while still maintaining competitive compression rates.

5.3 Token Usage Distribution Shift

Figure 3 shows the distribution of normalized token lengths before and after applying DRP across all benchmarks using R1-Distill-Qwen-7B. Detailed observations are discussed below:

Concise Reasoning Across Tasks. DRP effectively compresses the overall reasoning length across all benchmarks. The main density of the token distribution shifts leftward, especially on harder out-of-domain tasks like MATH500, AIME24, and AMC. This demonstrates that DRP encourages more concise reasoning behavior.

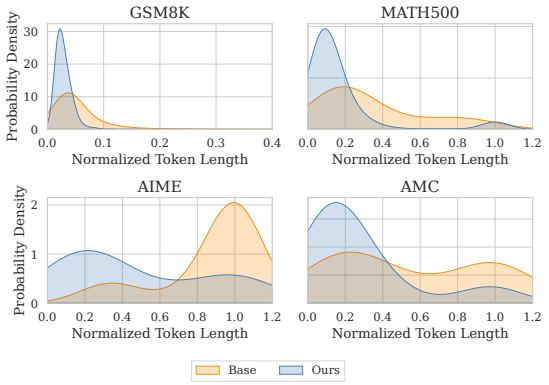

Figure 3: Normalized token length distributions across GSM8K, MATH500, AIME24, and AMC before and after SFT using the DeepSeek-R1-Distill-Qwen-7B model. The horizontal axis indicates the normalized token length (token count divided by the maximum allowed length), and the vertical axis represents the probability density. Blue curves correspond to our method (DRP), and orange curves denote the baseline. The reduction in long-tail completions and high-token outliers indicates that DRP mitigates verbose and degenerate reasoning failures, resulting in more robust and efficient inference.

Eliminating Verbose and Degenerate Reasoning Failures. DRP significantly reduces long-tail completions that typically result from verbose or degenerate reasoning. On AMC, the baseline exhibits a bimodal distribution, where the secondary peak reflects unnecessarily lengthy yet still valid reasoning paths. More critically, we observe a clear drop in density near the upper token limit—especially in AIME and AMC—indicating that DRP mitigates degenerate cases where the model previously generated excessively long or looping outputs due to failure to converge (Section 4.6). This results in improved *robustness*, defined here as the model's ability to terminate reasonably when it cannot reach a correct solution.

Further Compression of Already Efficient Reasoning Paths. On datasets such as GSM8K and MATH500, where baseline models already produce relatively short completions, DRP still yields measurable compression gains. This indicates that DRP not only removes verbosity but also optimizes reasoning even in high-performing regimes.

Dataset	Method	Accuracy	Tokens	Avg. Units
GSM8K	7B Base	91.7%	917	–
	No decomposition	91.0%	434	1
	Step-based	92.7%	350	8.3
	Skill-based	**94.1%**	328	12.6
	Sentence-based	91.3%	511	18.1
OOD Tasks				
MATH500	7B Base	92.4%	2486	–
	No decomposition	88.6%	2102	1
	Step-based	92.0%	1905	15.8
	Skill-based	**93.0%**	1781	23.6
	Sentence-based	89.2%	2301	34.8
AMC	7B Base	31/40	4845	–
	No decomposition	29/40	4028	1
	Step-based	31/40	4975	34.8
	Skill-based	**33/40**	3258	49.6
	Sentence-based	30/40	3881	75.6
AIME24	7B Base	15/30	8674	–
	No decomposition	13/30	6201	1
	Step-based	14/30	4678	54.6
	Skill-based	**15/30**	4966	69.2
	Sentence-based	13/30	5100	117.6

Table 2: Comparison of different reasoning structure representations under the same DRP framework. Avg. Units denotes the average number of decomposition units per example.

6 Ablation Studies

6.1 RQ1: Is Skill-Based Decomposition Effective Due to Finer Granularity?

Our skill-based decomposition provides the structural basis for subsequent pruning. We therefore examine how different decomposition variants affect pruning behavior and reasoning performance under the same DRP framework, including skill-based steps, sentence-based, default step-based, and no decomposition.

As shown in Table 2, **sentence-based decomposition**, despite producing the finest granularity, consistently degrades performance across benchmarks. Further analysis indicates that sentence-level segmentation induces more aggressive pruning (see Appendix H), frequently removing reflective or transitional phrases (e.g., "let me think again"). Although linguistically redundant, such phrases often support logical continuity, and their removal results in less coherent reasoning traces.

Overall, these results indicate that **decomposition granularity alone does not determine effectiveness**: excessively fine-grained segmentation can be counterproductive, while skill-based decomposition better balances granularity and semantic integrity.

Method	GSM8K		MATH500		AIME24		AMC	
	Accuracy	Tokens	Accuracy	Tokens	Accuracy	Tokens	Accuracy	Tokens
7B-base student model	91.7%	917	92.4%	2486	15/30	8674	31/40	4845
with GPT-4o teacher	94.1%	328	93.0%	1781	15/30	4966	33/40	3258
with Gemini 2.0 Flash teacher	93.2%	419	91.8%	2245	14/30	4860	33/40	2835
with ChatGPT teacher	89.2%	386	88.1%	2122	14/30	5265	31/40	3101
with DeepSeek-V3 teacher	94.7%	340	92.4%	1908	15/30	4640	33/40	2583

Table 3: Impact of different teacher models on DRP performance. We compare GPT-4o, Gemini 2.0 Flash, ChatGPT, and DeepSeek-V3 as pruning teachers, evaluating downstream accuracy and average token usage.

Dataset	Method	Accuracy	Tokens
GSM8K	7B Base	91.7%	917
	Distill	90.7%	425
	DRP	94.1%	328
OOD Tasks			
MATH500	7B Base	92.4%	2486
	Distill	88.6%	2152
	DRP	93.0%	1781
AIME24	7B Base	15/30	8674
	Distill	13/30	6417
	DRP	15/30	4966
AMC	7B Base	31/40	4845
	Distill	28/40	4279
	DRP	33/40	3258

Table 4: Comparison between 7B base model, direct distillation from GPT-4o, and our DRP method across both in-distribution and OOD math benchmarks.

Comparison	Trace Pattern Sim.	Style Sim.
Base vs. Distill	0.24	0.34
Base vs. DRP	0.58	0.61
Distill vs. DRP	0.41	0.47

Table 5: Trace pattern and style similarity between different generation methods, averaged across all evaluation tasks.

6.2 RQ2: Does Preserving Reasoning Structure Matter More Than Shortening CoTs?

We compare DRP with a direct distillation baseline, where the student is trained on concise CoTs generated by GPT-4o. While direct distillation substantially shortens reasoning traces, it generalizes poorly under distribution shift. As shown in Table 4, it achieves large token reductions on GSM8K with minimal accuracy loss, but suffers notable degradation on OOD benchmarks such as MATH500, AIME24, and AMC. This suggests that aggressive compression alone is insufficient, as it often removes intermediate reasoning steps critical for robust transfer. In contrast, DRP preserves structured reasoning while removing redundancy, resulting in more stable performance across both in-distribution and OOD settings.

Trace Pattern and Style Similarity. To explain this gap, we analyze response similarity in terms of trace pattern and style (definitions in Appendix F). Direct distillation significantly alters both reasoning trajectories and linguistic expression, whereas DRP remains closer to the base model along both dimensions. This indicates that DRP achieves effective pruning by reducing redundancy without collapsing the underlying reasoning structure, which helps explain its superior generalization.

6.3 RQ3: Teacher–student similarity vs. teacher strength for learnable distillation

To evaluate the sensitivity of our pruning framework to the choice of teacher model, we experiment with four different large language models (LLMs): GPT-4o, Gemini 2.0 Flash, ChatGPT, and DeepSeek-V3. As shown in Table 3, all teachers consistently improve both accuracy and token efficiency over the 7B-base student, demonstrating the robustness of DRP across diverse teacher choices.

Distillation gain is defined as the average accuracy improvement across four benchmarks relative to the base model. Here, *Strength* denotes the teacher's average task accuracy, and *Similarity* measures teacher–student alignment as defined in Appendix F. Jointly considering Tables 3 and 6, we observe that DRP remains effective even when using a weaker teacher such as ChatGPT, while higher teacher strength or same-family similarity alone does not necessarily yield larger gains. In particular, DeepSeek-V3 achieves high similarity but relatively limited improvement, indicating that effective pruning depends on how structural information is leveraged rather than on raw capability alone. This result complements RQ2 by identifying a boundary condition: although skill-based decomposition induces a stable and learnable structure,

Teacher Model	Strength	Similarity	Norm. Gain
GPT-4o	0.62	0.59	+3.23
Gemini 2.0 Flash	0.60	0.47	+1.07
ChatGPT	0.41	0.52	-6.18
DeepSeek-V3	0.74	0.71	+2.70

Table 6: Teacher strength, teacher–student similarity, and normalized distillation gain.

its benefit is not automatically realized without effective utilization during pruning.

Here, *Strength* denotes the teacher's average accuracy across four evaluation tasks, *Similarity* measures the alignment between teacher and student responses, combining trace pattern and style similarity as defined in Appendix F. Spearman ρ quantifies the monotonic relationship between teacher–student similarity and DRP gain under each teacher, indicating how strongly similarity predicts learnability.

7 Conclusion

We propose skill-based Distilled Reasoning Pruning, a structure-aware framework for distilling smaller reasoning models using pruned reasoning traces. DRP outperforms direct distillation by preserving consistent reasoning structure while removing redundancy, highlight that effective reasoning compression depends on structural consistency rather than granularity or stronger teachers alone.

8 Acknowledgments

We wish to thank the anonymous reviewers for their helpful comments, feedback, and suggestions. Some experiments were conducted on the UMBC HPCF, supported by the National Science Foundation under Grant No. CNS-1920079. This material is also based on research that is in part supported by DARPA for the SciFy program under agreement number HR00112520301. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of DARPA or the U.S. Government.

9 Limitations

While our method demonstrates strong performance on existing small-scale reasoning models, it remains uncertain how well it generalizes to other architectures. Currently, there is a limited number of publicly available small-sized LRMs with strong reasoning capabilities, making broad validation challenging. In addition, the training paradigms for reasoning models are evolving rapidly, and it is unclear whether the overthinking and inefficiency issues we target will persist in future model generations. We view our work as a step toward addressing current bottlenecks, but acknowledge that its relevance may shift as the landscape of LLM training continues to change.

10 Ethics

This work relies on publicly available datasets and uses an LLM-based judge accessed via the OpenAI API for reward evaluation (OpenAI, 2024). We do not access, attempt to access, or infer any proprietary training data or internal components of the underlying models. All experiments are conducted using standard model inference and optimization procedures, without collecting or processing personal or sensitive user data.

Risks The several datasets in our experiment are sourced from publicly available sources. However, we cannot guarantee that they are devoid of socially harmful or toxic language. We use ChatGPT [4] to correct grammatical errors in this paper.

References

AI-MO Team. 2024a. AIMO Validation Set - AIME Subset. https://huggingface.co/datasets/AI-MO/aimo-validation-aime.

AI-MO Team. 2024b. AIMO Validation Set - AMC Subset. https://huggingface.co/datasets/AI-MO/aimo-validation-amc.

Meng Cao et al. 2024. Enhancing reinforcement learning with dense rewards from language model critic. In *EMNLP*.

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 arXiv:2503.09567*.

Xi Chen, Wei Xue, and Yike Guo. 2026a. Actormind: Emulating human actor reasoning for speech role-playing. *Preprint*, arXiv:2604.11103.

[4] https://chatgpt.com/

Xi Chen and Min Zeng. 2025. Prototype conditioned generative replay for continual learning in NLP. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12754–12770, Albuquerque, New Mexico. Association for Computational Linguistics.

Xiaohong Chen et al. 2024a. Confusion-resistant federated learning via diffusion-based data harmonization on non-iid data. *NeurIPS*.

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

Zhangquan Chen et al. 2024c. A three-phases-lora fine-tuned hybrid llm integrated with strong prior module in the education context. In *ICANN*.

Zhangquan Chen et al. 2025b. Dv-matcher: Deformation-based non-rigid point cloud matching guided by pre-trained visual features. In *CVPR*.

Zhangquan Chen et al. 2025c. Sifthinker: Spatially-aware image focus for visual reasoning. *arXiv preprint arXiv:2508.06259*.

Zhangquan Chen et al. 2025d. Think with 3d: Geometric imagination grounded spatial reasoning from limited views. *arXiv preprint arXiv:2510.18632*.

Zhangquan Chen et al. 2025e. Visrl: Intention-driven visual perception via reinforced reasoning. *arXiv preprint arXiv:2503.07523*.

Zhangquan Chen et al. 2026b. Omnivideo-r1: Reinforcing audio-visual reasoning with query intention and modality attention. *arXiv preprint arXiv:2602.05847*.

Kexin Chu et al. 2025a. Dynamic expert quantization for scalable mixture-of-experts inference. *arXiv preprint arXiv:2511.15015*.

Kexin Chu et al. 2025b. Mcam: Efficient llm inference with multi-tier kv cache management. In *ICDCS*.

Kexin Chu et al. 2025c. Selective kv-cache sharing to mitigate timing side-channels in llm inference. *arXiv preprint arXiv:2508.08438*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, et al. 2025. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2502.13260*.

Muzhi Dai, Chenxu Yang, and Qingyi Si. 2025. S-grpo: Early exit via reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.07686*.

Google DeepMind. 2025. Introducing gemini 2.0 flash. https://blog.google/technology/google-deepmind/gemini-model-updates-february-2025/.

Hang Ding et al. 2026. Dynaweb: Model-based reinforcement learning of web agents. *arXiv preprint arXiv:2601.22149*.

Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Aurick Qiao, and Hao Zhang. 2024. Efficiently serving llm reasoning programs with certaindex. *arXiv preprint arXiv:2412.20993*.

Yichao Fu, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. 2025. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*.

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

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

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 arXiv:2103.03874*.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*.

Fanding Huang, Guanbo Huang, Xiao Fan, Yi He, Xiao Liang, Xiao Chen, Qinting Jiang, Faisal Nadeem Khan, Jingyan Jiang, and Zhi Wang. 2026. Semantic-space exploration and exploitation in rlvr for llm reasoning. *arXiv preprint arXiv:2509.23808*.

Fanding Huang, Jingyan Jiang, Qinting Jiang, Hebei Li, Faisal Nadeem Khan, and Zhi Wang. 2025. Cosmic: Clique-oriented semantic multi-space integration for robust clip test-time adaptation. In *CVPR*, pages 9772–9781.

Fanding Huang, Zihao Yao, and Wenhui Zhou. 2023. Dtbs: Dual-teacher bi-directional self-training for domain adaptation in nighttime semantic segmentation. In *ECAI*, pages 1084–1091.

Ziwei Ji et al. 2025. Calibrating verbal uncertainty as a linear feature to reduce hallucinations. In *EMNLP*.

Jiantong Jiang, Peiyu Yang, Rui Zhang, and Feng Liu. 2025. Towards efficient large language model serving: A survey on system-aware kv cache optimization. *Authorea Preprints*.

Yuxuan Jiang and Francis Ferraro. 2026a. Beyond math: Stories as a testbed for memorization-constrained reasoning in llms. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5590–5607.

Yuxuan Jiang and Francis Ferraro. 2026b. Scribe: Structured mid-level supervision for tool-using language models. *arXiv preprint arXiv:2601.03555*.

Guangchen Lan et al. 2025a. Contextual integrity in llms via reasoning and reinforcement learning. In *NeurIPS*.

Guangchen Lan et al. 2025b. Mappo: Maximum a posteriori preference optimization with prior knowledge. *arXiv preprint arXiv:2507.21183*.

Yiming Lei et al. 2026. A multi-scale graph learning framework for financial fraud detection.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, et al. 2024a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. 2024b. Selective reflection-tuning: Student-selected data recycling for llm instruction-tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 16189–16211.

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

Yuqi Li et al. 2025b. Efficient medical image segmentation via reinforcement learning-driven k-space sampling. *IEEE TETCI*.

Yuqi Li et al. 2025c. Frequency-aligned knowledge distillation for lightweight spatiotemporal forecasting. In *ICCV*.

Yuqi Li et al. 2025d. A preference-driven methodology for efficient code generation. *IEEE Transactions on Artificial Intelligence*.

Yuqi Li et al. 2026a. A comprehensive survey of interaction techniques in 3d scene generation. *Authorea*.

Yuqi Li et al. 2026b. Sepprune: Structured pruning for efficient deep speech separation. In *AAAI*.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Dingyuan Liu et al. 2026. The health-wealth gradient in labor markets. *Computation*.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*.

OpenAI. 2022. Chatgpt: Openai's conversational ai. https://chat.openai.com/.

OpenAI. 2024a. Gpt-4o: Openai's multimodal model. https://openai.com/index/hello-gpt-4o/.

OpenAI. 2024b. https://openai.com/index/learning-to-reason-with-llms/.

OpenAI. 2024. Openai api documentation. https://platform.openai.com/docs. Accessed via official OpenAI API.

Jun Rao et al. 2025a. A data-centric perspective on the lifecycle of large language models. *TechRxiv*.

Jun Rao et al. 2025b. Dynamic sampling that adapts: Iterative dpo for self-aware mathematical reasoning. *Preprint*, arXiv:2505.16176.

Qiannan Shen et al. 2025a. Ai-enhanced disaster risk prediction with explainable shap analysis. *Research Square*.

Qiannan Shen et al. 2026a. Business resilience index (bri): Evaluating economic recovery. *Sustainability*.

Qiannan Shen et al. 2026b. Mftformer: Meteorological-frequency-temporal transformer for traffic flow prediction. *Research Square*.

Zixu Shen et al. 2025b. Expertflow: Adaptive expert scheduling and memory coordination for efficient moe inference. *arXiv preprint arXiv:2510.26730*.

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

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957.

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

Yongqi Tong, Dawei Li, Sizhe Wang, Yujia Wang, Fei Teng, and Jingbo Shang. 2024. Can llms learn from previous mistakes? investigating llms' errors to boost for reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3065–3080.

Dingzhou Wang, Lu Chang, Luyao Men, Jiajun He, Yinuo Yang, and Yefeng Liang. 2025. Improving sequential recommendations with tokenlevel llm representatio. In *2025 4th International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE)*, pages 509–514. IEEE.

Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*.

Canran Xiao et al. 2026a. Reversible primitive–composition alignment for continual vision–language learning. In *ICLR*.

ShiLin Xiao et al. 2026b. Meta-ucf: Unified task-conditioned lora generation for continual learning in large language models. In *ICLR*.

Zequn Xie. 2026. Conquer: Context-aware representation with query enhancement for text-based person search. *arXiv preprint arXiv:2601.18625*.

Zequn Xie et al. 2025. Chat-driven text generation and interaction for person retrieval. In *EMNLP*.

Zequn Xie et al. 2026a. Delving deeper: Hierarchical visual perception for robust video-text retrieval. *arXiv preprint arXiv:2601.12768*.

Zequn Xie et al. 2026b. Hvd: Human vision-driven video representation learning for text-video retrieval. *arXiv preprint arXiv:2601.16155*.

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

Jingxian Xu, Mengyu Zhou, Weichang Liu, Hanbing Liu, Shi Han, and Dongmei Zhang. 2025b. Twt: Thinking without tokens by habitual reasoning distillation with multi-teachers' guidance. *arXiv preprint arXiv:2503.24198*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, and Radha Poovendran. 2024. Stronger models are not stronger teachers for instruction tuning. *arXiv preprint arXiv:2411.07133*.

Chao Xue and Ziyuan Gao. 2025. Structcoh: Structured contrastive learning for context-aware text semantic matching. In *PRICAI*, pages 300–315.

Chao Xue, Di Liang, Pengfei Wang, and Jing Zhang. 2024. Question calibration and multi-hop modeling for temporal question answering. In *AAAI*, volume 38, pages 19332–19340.

Chao Xue, Di Liang, Sirui Wang, Jing Zhang, and Wei Wu. 2023. Dual path modeling for semantic matching by perceiving subtle conflicts. In *ICASSP*, pages 1–5.

Chao Xue et al. 2026a. Reason only when needed: Efficient generative reward modeling via model-internal uncertainty.

Chao Xue et al. 2026b. Why supervised fine-tuning fails to learn: A systematic study of incomplete learning in large language models.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. 2025. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.

Jinhua Yang, Ting Liu, Yiming Taclis Luo, Tianyue Niu, Patrick Pang, Ao Xiang, and Qin Yang. 2026a. Exploring the application boundaries of llms in mental health: A systematic scoping review. *Frontiers in Psychology*, 16:1715306.

Panqi Yang et al. 2026b. Instrucrobo: Object-centric multi-instruction decoupling model for explainable robotic manipulation. *Engineering Applications of Artificial Intelligence*, 171.

Panqi Yang et al. 2026c. Unibvr: Balancing visual and reasoning abilities in unified 3d scene understanding. *Neurocomputing*, 671.

Panqi Yang et al. 2026d. Unihoi: Unified human-object interaction understanding via unified token space. In *AAAI*.

Jiawei Yao et al. 2024. Swift sampler: Efficient learning of sampler by 10 parameters. *NeurIPS*.

Lei Yu et al. 2024a. Mechanistic understanding and mitigation of language model non-factual hallucinations. In *EMNLP*.

Lei Yu et al. 2024b. Robust llm safeguarding via refusal feature adversarial training. *arXiv preprint arXiv:2409.20089*.

Zixiong Yu et al. 2026. Mathagent: Adversarial evolution of constraint graphs for mathematical reasoning data synthesis. *Preprint*, arXiv:2604.11188.

Shuang Zeng et al. 2025a. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*.

Shuang Zeng et al. 2025b. Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. *arXiv preprint arXiv:2509.22548*.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Yunhua Zhou, and Xipeng Qiu. 2025c. Revisiting the test-time scaling of o1-like models: Do they truly possess test-time scaling capabilities? *arXiv preprint arXiv:2502.12215*.

Xiaoling Zhou et al. 2022. Understanding difficulty-based sample weighting. In *ECML PKDD*.

Xiaoling Zhou et al. 2024a. Adversarial training with anti-adversaries. *IEEE TPAMI*.

Xiaoling Zhou et al. 2024b. Boosting model resilience via implicit adversarial data augmentation. *arXiv preprint arXiv:2404.16307*.

Xunyu Zhu, Jian Li, Can Ma, and Weiping Wang. 2024. Improving mathematical reasoning capabilities of small language models via feedback-driven distillation. *arXiv preprint arXiv:2411.14698*.

A Outliers and Cutting Off

We can observe from Figure 4 that most answers are captured by 11K tokens, though some require up to 38K, revealing a long-tail pattern.

As shown in Table 7, applying the 12k token cutoff has a significant effect, particularly on harder

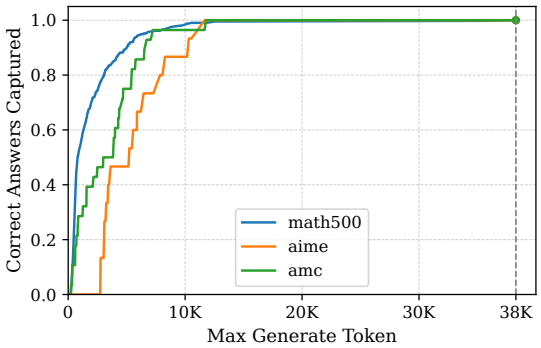

Figure 4: The x-axis denotes the model's maximum generation length, and the y-axis shows the proportion of correct answers recovered within that budget, with the R1-Distill-Qwen-7B.

benchmarks like AIME24 and AMC. On these tasks, the model occasionally fails to generate valid answers and enters degenerate loops, repeatedly producing the same text until reaching the maximum token limit. As a result, the average token usage can increase by up to 5×, which does not reflect the true distribution of reasoning length and severely skews efficiency comparisons.

B Fine-Tuning Details

We fine-tune the DeepSeek-R1-Distill-Qwen models (7B and 1.5B) using the LLAMA-FACTORY framework with LoRA adaptation. The training data includes 8,000 samples drawn from GSM8K and PRM12K. We set a cutoff length of 4096 tokens for both input and output sequences.

The models are fine-tuned for 3 epochs using the following configuration:

- **Cutoff length:** 4096
- **Max samples:** 8000
- **Batch size:** 2 (with gradient accumulation of 4)
- **Learning rate:** 3e-5 with cosine schedule
- **Precision:** bf16
- **Validation split:** 5% of training data
- **Evaluation strategy:** every 300 steps

Training is performed using two A100 80GB GPU. All experiments use `overwrite_cache=true` and 8 parallel preprocessing workers. The resulting models are

Method	GSM8K		MATH500		AIME24		AMC	
	Accuracy	Tokens	Accuracy	Tokens	Accuracy	Tokens	Accuracy	Tokens
Baseline (no cutoff)	91.7%	917	92.5%	5435	15/30	54339	32/40	18736
Baseline (12k cutoff)	91.7%	917	92.4%	2486	15/30	8674	31/40	4845

Table 7: Impact of applying a 12k token cutoff on measured token usage across benchmarks. Accuracy remains largely unaffected, while the average token count drops significantly—particularly on harder tasks like AIME and AMC. This demonstrates the necessity of outlier mitigation for fair efficiency comparison.

directly used for downstream evaluation without additional tuning.

C Additional Related Works on LLMS Reasoning Capacity

Among various LLM abilities, foundational research in semantic matching (Xue et al., 2023; Xue and Gao, 2025) and temporal QA (Xue et al., 2024) has paved the way for complex understanding. While domain adaptation (Huang et al., 2023) and test-time integration (Huang et al., 2025) enhance robustness, reasoning ability achieved by post-training (Huang et al., 2026; Lan et al., 2025b) stands as a key frontier. Given that traditional SFT may suffer from incomplete learning (Xue et al., 2026b), recent works introduce uncertainty-aware reward modeling (Xue et al., 2026a), dense feedback from LM critics (Cao et al., 2024), and preference-driven code generation (Li et al., 2025d) to refine alignment. Moreover, recent studies have advanced role-playing reasoning through human-actor emulation (Chen et al., 2026a) and improved continual learning via prototype-conditioned generative replay (Chen and Zeng, 2025).

The scope of reasoning has expanded into multimodal domains, where models now "think with 3D" (Chen et al., 2025d) and utilize reinforced visual perception (Chen et al., 2025e,c). Advanced frameworks like OmniVideo-R1 (Chen et al., 2026b) and DV-Matcher (Chen et al., 2025b) further integrate audio-visual and geometric cues. Such multimodal intelligence significantly boosts retrieval performance, including chat-driven search (Xie et al., 2025; Xie, 2026) and vision-driven video representation (Xie et al., 2026b,a). Furthermore, safeguarding techniques (Yu et al., 2024b; Lan et al., 2025a) and hallucination mitigation (Yu et al., 2024a; Ji et al., 2025) ensure the reliability of these complex systems.

To support deployment, system efficiency and continual learning have become indispensable. Inference is optimized via multi-tier KV cache management (Chu et al., 2025b), selective sharing (Chu et al., 2025c), and adaptive MoE scheduling (Chu et al., 2025a; Shen et al., 2025b). Moreover, lightweight techniques like parameter-efficient sampling (Yao et al., 2024), structured pruning (Li et al., 2026b), and distillation (Li et al., 2025c) enable efficient task execution, from medical imaging (Li et al., 2025b) to web agents (Ding et al., 2026). Finally, reversible alignment (Xiao et al., 2026a) and unified LoRA generation (Xiao et al., 2026b; Chen et al., 2024c) allow models to learn continually, while comprehensive interaction surveys (Li et al., 2026a) and robust federated learning (Chen et al., 2024a) continue to push the boundaries of scalable AI.

Beyond core architecture, recent research emphasizes a data-centric perspective on the entire LLM lifecycle (Rao et al., 2025a), where mathematical reasoning is further enhanced by adversarial data synthesis (Yu et al., 2026) and iterative DPO with dynamic sampling (Rao et al., 2025b). To improve model robustness and learning efficiency, advanced techniques such as difficulty-based sample weighting (Zhou et al., 2022), implicit data augmentation (Zhou et al., 2024b), and adversarial training with anti-adversaries (Zhou et al., 2024a) have been introduced.

In the context of embodied AI and spatio-temporal reasoning, unified token spaces for human-object interaction (Yang et al., 2026d) and object-centric decoupling for robotic manipulation (Yang et al., 2026b) have improved explainable decision-making. These abilities are further integrated into 3D scene understanding (Yang et al., 2026c) and autonomous driving through spatio-temporal Chain-of-Thought (CoT) (Zeng et al., 2025a) and dual-memory vision-language navigation (Zeng et al., 2025b). Furthermore, the application of multi-scale graph learning and Transformers has shown significant potential in predicting complex real-world dynamics, including financial fraud detection (Lei et al., 2026), traffic flow pre-

diction (Shen et al., 2026b), disaster risk assessment (Shen et al., 2025a), mental health (Yang et al., 2026a), recommendations (Wang et al., 2025) and socio-economic metrics such as labor market gradients (Liu et al., 2026) and business resilience indices (Shen et al., 2026a).

D Decomposition Prompts

Decomposition Prompts Used in Experiments

Skill-Based Decomposition Prompt. Segment the explanation into a sequence of clear, non-overlapping steps, where each step corresponds to exactly one atomic mathematical skill. These skills may include, for example, *addition*, *subtraction*, *applying a formula*, *interpreting a quantity*, *simplifying*, *checking a condition*, and so on. Each step should focus on a single functional operation.

Sentence-Based Decomposition Prompt. Segment the explanation into a sequence of steps by splitting at sentence boundaries. Each sentence should be treated as one step, without further semantic refinement or functional labeling.

Default Step-Wise Decomposition Prompt. Segment the explanation into clear, non-overlapping steps. Each step may contain multiple operations as long as the reasoning remains coherent.

E Prompt Templates

The full prompt for skill-based step segmentation:

Skill-Based Step Segmentation Prompt (Full Version)

You are given a complete reasoning path for a math problem. Your task is to segment it into a sequence of clear, non-overlapping steps, where each step corresponds to exactly one atomic mathematical skill. These skills may include, for example, addition, subtraction, applying a formula, interpreting a quantity, simplifying, checking a condition, and so on.
Use the following format for each step:
`Step n: {{original text segment}}`
`Skill: {{name of the skill used}}`
Only segment and label the steps — do not solve or modify the original content in any way.

The full prompt for self-revision:

Self-Revision Prompt (Full Version)

You are an expert in mathematical reasoning compression.
Given a list of reasoning steps labeled with their respective skills, your task is to evaluate and revise each step according to one of the following four actions:
*1. **KEEP**: The step is necessary and already concise. Keep it unchanged.*
*2. **DELETE**: The step is unnecessary and should be removed entirely.*
*3. **SINGLE-STEP COMPRESS**: The step is necessary but verbose; rewrite it in a more concise way.*
*4. **MULTI-STEP COMPRESS**: The step can be merged with neighboring steps; write a combined, cleaner version.*
If the final step clarifies the final answer (e.g., "The answer is..."), retain it.
After completing the revision of each step, synthesize the revised steps into a coherent explanation. Ensure the output is fluent, logically sound, and matches the speaker's tone and style.

F Trace Pattern and Style Similarity

Trace Pattern Similarity. Trace pattern similarity measures whether two responses follow similar high-level solution trajectories, including the presence and extent of exploratory or reflective reasoning (e.g., intermediate checking or self-correction), without requiring step-by-step alignment. Because different models may produce reasoning traces with substantially different numbers of steps, we characterize each response using coarse-grained structural properties such as overall length, number of reasoning steps, and the distribution of content across the trace. Similarity is computed based on normalized differences in these global trajectory properties, capturing similarity in problem-solving strategies rather than surface-level form.

Style Similarity. Style similarity evaluates whether two responses exhibit similar linguistic expression styles after abstracting away task-specific content. We apply a de-contenting procedure that replaces numbers, formulas, and named entities with placeholders, while preserving discourse markers, modality, punctuation, and formatting cues. Style similarity is then computed using cosine similarity between embeddings of the resulting texts. To avoid bias introduced by unequal exploration depth, the metric is length-normalized and does not penalize additional exploratory steps that appear in only one response.

G Skill Base Decomposition Evaluation

To assess the quality of our skill-based step decomposition, we conduct a pairwise comparison experiment using **Gemini 2.0 Flash** as the judge (Gao et al., 2023; Li et al., 2024a, 2025d). For each reasoning trace T, we generate two segmentations: (1) our proposed *skill-based decomposition*, and (2) a baseline *sentence-based split* (obtained by naïvely splitting at punctuation or sentence boundaries).

We prompt Gemini to select the version with more **semantically coherent**, **structurally stable**, and **granularity-appropriate** steps, using the following instruction:

> Given two versions of step-by-step reasoning derived from the same original trace, please choose the version that shows better semantic coherence, structural consistency, and appropriate granularity. Respond with 'A', 'B', or 'Same'.

Among 50 randomly sampled traces from GSM8K, Gemini selected our skill-based decomposition in **33** cases, while preferring the baseline in only **17** cases. This result suggests that our segmentation method provides more interpretable and structurally aligned reasoning steps, which are beneficial for downstream pruning and distillation.

H Skill-based Decomposition vs. Default

In this section, we show a non-cherry picking example of how our skill based decomposition different from the default one. As shown in Table 9, our method has a more fine-grained step decomposition effect which leads to a more detailed pruning effect.

Also, table 8 shows sentence-level pruning often removes reflective or transitional sentences that connect consecutive reasoning steps. While the remaining statements may still be logically correct, the resulting reasoning trace tends to read less fluently, with weakened continuity between steps. As a result, the logical progression can appear abrupt or fragmented, even when the final answer remains unchanged.

I Teacher Strength

Here we show the origin teacher strength on Table!10.

J Comparison between different decomposition method

In this section, we do some analysis with case study on how different decomposition performs.

K Long CoT vs. Short CoT

In this section, we show a non-cherry picking example of how our skill based decomposition different from the default one. As shown in Table 11, our DRP-style CoT are more informative. That might because of LLM's unfair ability on different aspects (Jiang and Ferraro, 2026a).

Original Long-form CoT	After Sentence-level Pruning
First, I need to determine Joy's reading rate. She reads 8 pages in 20 minutes, so her reading rate is 8 pages per 20 minutes. Next, I'll calculate how many pages she reads per minute by dividing 8 pages by 20 minutes, which gives 0.4 pages per minute. Then, to find out how many minutes it takes her to read 120 pages, I'll divide 120 pages by her reading rate of 0.4 pages per minute, resulting in 300 minutes. Finally, I'll convert 300 minutes into hours by dividing by 60, which equals 5 hours.	She reads 8 pages in 20 minutes, so her reading rate is 0.4 pages per minute. Dividing 120 pages by 0.4 pages per minute gives 300 minutes. 300 minutes equals 5 hours.

Table 8: Example of sentence-level pruning. Sentence-based segmentation removes reflective and transitional sentences, resulting in shorter but less coherent reasoning traces.

Form	Decomposition Output
Problem	Joy can read 8 pages of a book in 20 minutes. How many hours will it take her to read 120 pages?
Original Long-form CoT	First, I need to determine Joy's reading rate. She reads 8 pages in 20 minutes, so her reading rate is 8 pages per 20 minutes. Next, I'll calculate how many pages she reads per minute by dividing 8 pages by 20 minutes, which gives 0.4 pages per minute. Then, to find out how many minutes it takes her to read 120 pages, I'll divide 120 pages by her reading rate of 0.4 pages per minute, resulting in 300 minutes. Finally, I'll convert 300 minutes into hours by dividing by 60, which equals 5 hours.
DRP Decomposition (Skill-Based)	**Step 1:** Determine Joy's reading rate. She reads 8 pages in 20 minutes, so her reading rate is 8 pages per 20 minutes. *Skill: Interpreting a quantity* **Step 2:** Calculate how many pages she reads per minute by dividing 8 pages by 20 minutes. *Skill: Division* **Step 3:** The result of the division is 0.4 pages per minute. *Skill: Simplifying* **Step 4:** To find out how many minutes it takes her to read 120 pages, divide 120 pages by 0.4 pages per minute. *Skill: Division* **Step 5:** The result of the division is 300 minutes. *Skill: Simplifying* **Step 6:** Convert 300 minutes into hours by dividing by 60. *Skill: Division* **Step 7:** The result of the division is 5 hours. *Skill: Simplifying*
DRP Pruning and Final Result	1. **KEEP:** The step is necessary and already concise. 2. **SINGLE-STEP COMPRESS:** Joy reads 8 pages in 20 minutes, so she reads 0.4 pages per minute (8 ÷ 20). 3. **DELETE:** This step is now included in the previous step. 4. **SINGLE-STEP COMPRESS:** To find the time for 120 pages, divide 120 by 0.4 pages per minute. 5. **DELETE:** This step is now included in the previous step. 6. **SINGLE-STEP COMPRESS:** Convert 300 minutes to hours by dividing by 60. 7. **DELETE:** This step is now included in the previous step. **Final Answer:** Okay, let's calculate the time it takes Joy to read 120 pages. Joy reads 8 pages in 20 minutes, which means she reads 0.4 pages per minute (8 ÷ 20). To find out how many minutes it takes her to read 120 pages, divide 120 by 0.4, resulting in 300 minutes. Convert 300 minutes into hours by dividing by 60, which gives us 5 hours. The answer is 5 hours.
Default Decomposition	**Step 1:** First, I need to determine Joy's reading rate. She reads 8 pages in 20 minutes, so her reading rate is 8 pages per 20 minutes. **Step 2:** Next, I'll calculate how many pages she reads per minute by dividing 8 pages by 20 minutes, which gives 0.4 pages per minute. **Step 3:** Then, to find out how many minutes it takes her to read 120 pages, I'll divide 120 pages by her reading rate of 0.4 pages per minute, resulting in 300 minutes. **Step 4:** Finally, I'll convert 300 minutes into hours by dividing by 60, which equals 5 hours.
Default Pruning and Final Result	**Step 1:** SINGLE-STEP COMPRESS Revised: Joy's reading rate is 8 pages per 20 minutes. **Step 2:** SINGLE-STEP COMPRESS Revised: Dividing 8 by 20, Joy reads 0.4 pages per minute. **Step 3:** SINGLE-STEP COMPRESS Revised: Dividing 120 by 0.4, it takes her 300 minutes to read 120 pages. **Step 4:** SINGLE-STEP COMPRESS Revised: Converting 300 minutes to hours: 300 ÷ 60 = 5 hours. **Final Answer:** Joy's reading rate is 8 pages per 20 minutes. Dividing 8 by 20, Joy reads 0.4 pages per minute. Dividing 120 by 0.4, it takes her 300 minutes to read 120 pages. Converting 300 minutes to hours: 300 ÷ 60 = 5 hours. **The answer is 5.**

Table 9: A comparison between DRP's skill-based decomposition and default segmentation with compression, for the same math problem.

Model	GSM8K	MATH500	AIME24	AMC
GPT-4o	96.2	67.6	9.3	74.6
DeepSeek-V3	89.3	90.2	39.2	76.0
Gemini	94.2	87.0	34.0	25.0
ChatGPT	89.0	54.0	55.0	10.0

Table 10: Model accuracy (%) on four math reasoning benchmarks. The last row reports the average accuracy across models for each benchmark.

Form	Reasoning Trajectory
Problem: Joy can read 8 pages of a book in 20 minutes. How many hours will it take her to read 120 pages?	
Original Long-form CoT	First, I need to determine Joy's reading rate. She reads 8 pages in 20 minutes, so her reading rate is 8 pages per 20 minutes. Next, I'll calculate how many pages she reads per minute by dividing 8 pages by 20 minutes, which gives 0.4 pages per minute. Then, to find out how many minutes it takes her to read 120 pages, I'll divide 120 pages by her reading rate of 0.4 pages per minute, resulting in 300 minutes. Finally, I'll convert 300 minutes into hours by dividing by 60, which equals 5 hours.
DRP-style (Ours)	Okay, let's calculate the time it takes Joy to read 120 pages. Joy reads 8 pages in 20 minutes, which means she reads 0.4 pages per minute ($8 \div 20$). To find out how many minutes it takes her to read 120 pages, divide 120 by 0.4, resulting in 300 minutes. Convert 300 minutes into hours by dividing by 60, which gives us 5 hours. The answer is 5 hours.
Short-form CoT	In one hour, there are 3 sets of 20 minutes. So, Joy can read $8 \times 3 = 24$ pages in an hour. It will take her $120 \div 24 = 5$ hours to read 120 pages.

Table 11: A comparison between original long-form, DRP-style, and short-form reasoning traces for the same math problem.