DeepPrune: Parallel Scaling without Inter-trace Redundancy

Shangqing Tu[1*], **Yaxuan Li**[2*], **Yushi Bai**[1], **Lei Hou**[1†], **Juanzi Li**[1]
[1]Tsinghua University, [2]ShanghaiTech University
{tsq25,bys22}@mails.tsinghua.edu.cn
liyx12023@shanghaitech.edu.cn,{houlei,lijuanzi}@tsinghua.edu.cn
https://deepprune.github.io

Abstract

Parallel scaling has emerged as a powerful paradigm to enhance reasoning capabilities in large language models (LLMs) by generating multiple Chain-of-Thought (CoT) traces simultaneously. However, this approach introduces significant computational inefficiency due to *inter-trace redundancy*—our analysis reveals that over 80% of parallel reasoning traces yield identical final answers, representing substantial wasted computation. To address this critical efficiency bottleneck, we propose **DeepPrune**, a novel framework that enables efficient parallel scaling through dynamic pruning. Our method features a specialized judge model trained with out-of-distribution data (AIME 2022, AIME 2023, and MATH 500) using oversampling techniques to accurately predict answer equivalence from partial reasoning traces, achieving 0.7072 AUROC on unseen reasoning models. Combined with an online greedy clustering algorithm that dynamically prunes redundant paths while preserving answer diversity. Comprehensive evaluations across three challenging benchmarks (AIME 2024, AIME 2025, and GPQA) and multiple reasoning models demonstrate that DeepPrune achieves remarkable token reduction of 65.73%–88.50% compared to conventional consensus sampling, while maintaining competitive accuracy within 3 percentage points. Our work establishes a new standard for efficient parallel reasoning, making high-performance reasoning more efficient. Our code and data are here: https://deepprune.github.io/.

1 Introduction

Large language models (LLMs) (OpenAI, 2024; Anthropic, 2024; Reid et al., 2024) have made remarkable progress on reasoning tasks (Guo et al., 2025; Team et al., 2025; Zeng et al., 2025), especially when equipped with long Chain-of-Thoughts (CoT) that can mimic human's thinking processes (Wei et al., 2022; Sprague et al., 2024). This advancement is driven by inference-time scaling (Jaech et al., 2024), a new paradigm that enhances LLM's reasoning capabilities via more computing in the test stage (Snell et al., 2025).

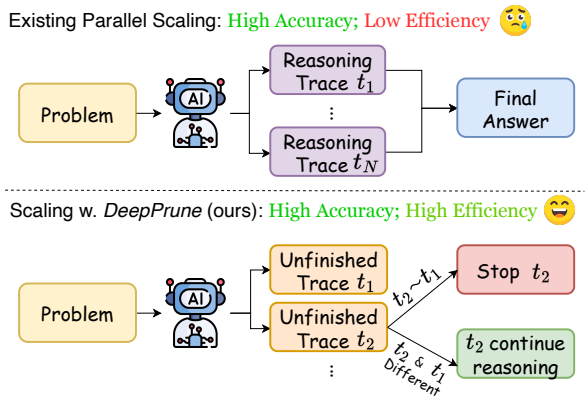


Figure 1: *DeepPrune* conducts early stopping based on the similarity between reasoning treaces to enhance the efficiency of parallel scaling and save diverse traces.

Generally, there are two types of inference-time scaling: sequential scaling and parallel scaling (Venkatraman et al., 2025). Sequential scaling (Muennighoff et al., 2025) focuses on increasing the computation in one reasoning trace like expanding the output length to 128k. While parallel scaling (e.g. best-of-n sampling) encourages generating multiple reasoning traces simultaneously, further pushing the total token cost to 100M or higher (Moshkov et al., 2025). However, beneath these advances lies a practical question: **How to achieve high performance with low token cost?**

Existing efficient reasoning methods mainly focus on alleviating the over-thinking of sequential scaling (Chen et al., 2024b; Hou et al., 2025; Zhang et al., 2025). There are few works designed for parallel scaling (Madaan et al., 2025), which typically adopt the LLM's internal signal like confidence (Fu

[*] Equal Contribution.
[†] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 13389–13403
July 2-7, 2026

et al., 2025b) for early stopping to improve the sampling efficiency. However, these confidence-based methods suffer from two fundamental limitations: (1) they fail to reduce redundancy *between* parallel reasoning paths, and (2) they risk prematurely terminating correct reasoning traces.

In this paper, we propose **DeepPrune**, as shown in Figure 1, a novel method that proactively prunes redundant parallel CoTs while preserving traces with diverse answers. Our approach is motivated by a key observation from preliminary experiments: approximately 80% of parallel reasoning traces yield identical final answers, while only 20% produce distinct solutions. This reveals significant redundancy in current parallel reasoning paradigms.

We further investigate whether early-stage trace similarity can predict final answer equivalence. Surprisingly, shallow semantic similarity measures (e.g., SentenceBERT on first 500 tokens) achieve only random-level performance (AUROC=0.58), while deeper LLM-based comparison (`Qwen3-4B-Instruct`) shows moderate improvement (AUROC=0.66) but remains suboptimal for practical deployment. This finding underscores the necessity for specialized models capable of understanding reasoning processes at a deeper level.

Inspired by this analysis, we train a LLM-based judge model that predicts redundancy between truncated reasoning traces. To enable accurate early stopping, we explore two truncation strategies including fixed-length prefixes and reasoning-step aligned segments. To address class imbalance and preserve answer diversity, we employ focal loss and oversampling techniques for training the judge model. For efficient online inference, we design a greedy clustering algorithm that dynamically prunes redundant paths during generation.

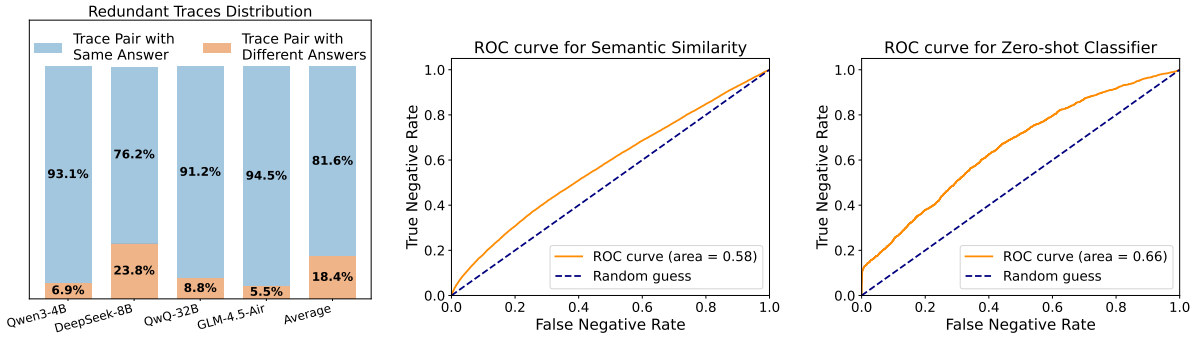
We conduct comprehensive experiments to prove the effectiveness of DeepPrune across diverse settings. To ensure robust generalization, we train our judge model on fully out-of-distribution data (AIME 2022, AIME 2023, and MATH 500) and evaluate it on unseen reasoning models and benchmarks. In offline evaluation, our judge model achieves an average AUROC of 0.7072 and TPR of 0.5063 when using first-800 tokens with oversampling, demonstrating strong cross-model generalization to three unseen reasoning models. More importantly, in online reasoning tasks across three challenging benchmarks (AIME 2024, AIME 2025, and GPQA) and three state-of-the-art reasoning models (DeepSeek-8B, Qwen3-32B, and GPT-OSS-20B), DeepPrune reduces token consumption by 65.73% to 88.50% compared to cons@512 which samples 512 traces and conducts majority voting, while maintaining comparable accuracy (within 3.4 percentage points). Notably, on AIME24 and AIME25 datasets, DeepPrune achieves up to 88.5% token reduction while maintaining or improving accuracy, substantially outperforming strong baselines like DeepConf (Fu et al., 2025b).

Our contributions are threefold: (1) We identify and quantify the pervasive problem of *inter-trace redundancy* in parallel reasoning, revealing that over 80% of computational resources are wasted on generating equivalent reasoning paths. (2) We propose DeepPrune, a novel framework that combines a trained judge model with online greedy clustering to efficiently prune redundant reasoning traces while preserving answer diversity. (3) Extensive offline and online experiments show that our method reduces token consumption by up to 88.5% without compromising accuracy, significantly outperforming existing baselines across multiple reasoning benchmarks and model architectures.

2 Related Work

Parallel Scaling. Parallel scaling has emerged as a pivotal paradigm for enhancing reasoning performance through concurrent generation of multiple reasoning traces (Chen et al., 2024a; Pan et al., 2025; Zheng et al., 2025). Self-Consistency (Wang et al., 2022) pioneered majority voting over diverse reasoning paths, while Best-of-N sampling (Brown et al., 2024) extended this concept through explicit candidate ranking. There are also tree-based exploration methods like ToT (Yao et al., 2023) that dynamically branch reasoning paths.

Efficient Reasoning. Efficient reasoning methods (Feng et al., 2025; Sui et al., 2025; Liu et al., 2025; Qu et al., 2025) aim to optimize the accuracy-compute trade-off during inference (Kang et al., 2025; Srivastava et al., 2025; Li et al., 2025). Prior research has explored reducing token usage in individual reasoning traces, such as through length-conscious fine-tuning (Liu et al., 2024; Arora and Zanette, 2025; Aggarwal and Welleck, 2025; Xia et al., 2025) or training-free prompting techniques (Renze and Guven, 2024; Han et al., 2024; Xu et al., 2025; Fu et al., 2025a; Aytes et al., 2025). Another line of work improves parallel scaling efficiency by early-stopping redundant samples via



(a) Redundant Traces Distribution (b) ROC of Semantic Similarity (c) ROC of LLM Judge

Figure 2: Analysis of Inter-trace Redundancy. (a) Distribution of same vs. different answer pairs of reasoning traces, revealing severe redundancy. (b) ROC curve for shallow semantic similarity (SentenceBERT) to distinguish traces with same answers from those with different ones, which shows limited predictive power (AUROC=0.58). (c) ROC curve for LLM-based deep comparison (Qwen3-4B-Instruct) achieves moderate improvement (AUROC=0.66).

confidence estimates (Fu et al., 2025b; Yang et al., 2025b) or by refining aggregation strategies (Wang et al., 2024, 2025b). While these methods address intra-trace verbosity or sample quantity reduction, they do not explicitly model redundancy *between* parallel reasoning paths. Our work directly targets this inter-trace redundancy, enabling proactive pruning while preserving answer diversity.

3 Preliminaries

3.1 Problem Definition

Given a set of n parallel reasoning traces $S_1 = \{t_1, t_2, ..., t_n\}$ generated concurrently for the same query, our objective is to reduce inter-trace redundancy while preserving answer diversity. We define a pruning process P that selects a subset of traces:

$$P(S_1) = S_2, \quad \text{where } S_2 \subseteq S_1$$

The pruned set S_2 should satisfy:

$$S_2 = \left\{ t_{k_1}, t_{k_2}, \ldots, t_{k_m} \;\middle|\; \begin{array}{l} \text{sim}(t_{k_i}, t_{k_j}) < \tau, \\ \forall i, j \leq m \end{array} \right\}$$

where $\text{sim}(t_{k_i}, t_{k_j})$ is the similarity, τ is a similarity threshold. The traces in S_2 continue reasoning to produce final answers $\{o_{k_1}, o_{k_2}, \ldots, o_{k_m}\}$.

To operate this process, we need to model the *Similarity* function. Since parallel reasoning focuses on answer accuracy, we simplify the judgment to predicting whether two incomplete traces will yield the same final answer. Formally, for any pair of unfinished traces (t_i, t_j), we predict whether their corresponding results (o_i, o_j) will be identical, which can be defined as the binary similarity function based on final answer equivalence:

$$\text{sim}(t_i, t_j) = \begin{cases} 1 & \text{if } R(o_i, o_j) = 1 \\ 0 & \text{if } R(o_i, o_j) = 0 \end{cases}$$

where $R(o_i, o_j)$ is the reward function for answer equivalence based on verifable rules or reward models. In this paper, we only consider queries with verifable answers, leaving others for future works.

3.2 Inter-trace Redundancy: The Efficiency Bottleneck of Parallel Reasoning

Recent advances in parallel scaling have significantly improved reasoning performance, but at the cost of substantial computational overhead. We identify **inter-trace redundancy** as the primary efficiency bottleneck: when generating multiple reasoning traces in parallel, a large proportion of tokens are wasted on producing semantically equivalent reasoning paths that lead to identical answers.

Parallel Reasoning Trace Collection. To uncover the phenomenon of inter-trace redundancy, we conduct a reasoning trace collection process. We select four widely-used reasoning models: `Deepseek-R1-Distill-Llama-8B`, `Qwen3-4B-Thinking-2507`, `GLM-4.5-Air` and `QWQ-32B` . These models are evaluated on over 100 problems from reasoning benchmarks including Math500 (Hendrycks et al., 2021), AIME24, AIME25 (AIME, 2025), and GPQA (Rein et al., 2024). For each problem, we generate 16 parallel reasoning traces per model. The traces for each problem are paired exhaustively, resulting in $\binom{16}{2} = 120$ pairs per problem. Thus, for each model, we obtain around 12,000 pairs in total. To

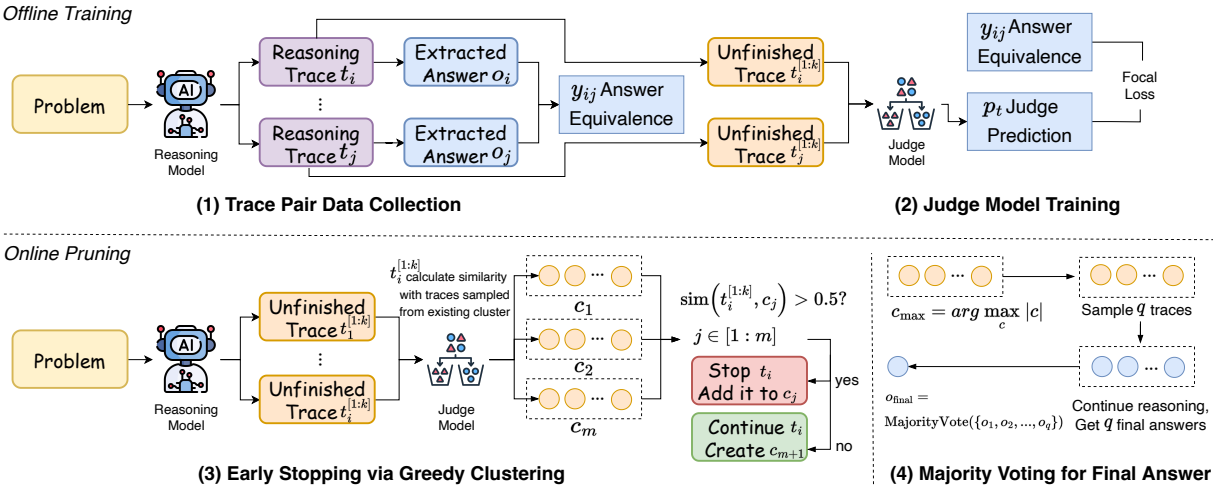


Figure 3: Overview of the **DeepPrune** framework. The *offline training* phase (top) involves constructing trace pair datasets with binary labels indicating answer equivalence, then training a judge model using focal loss and oversampling to address class imbalance. The *online pruning* phase (bottom) leverages the trained judge model to perform dynamic pruning via greedy clustering where traces are assigned to existing clusters or new ones based on similarity predictions, and concludes with majority voting on selected traces to determine the final answer.

determine answer equivalence, we use the rule-based reward function from DeepScaleR (Luo et al., 2025) to verify whether the final answers of each pair are identical. More details are in Appendix A.

The results, summarized in Figure 2(a), reveal a striking dominance of same-answer pairs across all models, with ratios exceeding 80% in most cases. Specifically, `GLM-4.5-Air` (Zeng et al., 2025) has 94.5% same-answer pairs. This severe class imbalance highlights that even with a modest number of samples (16 per problem), a large proportion of computational resources are wasted on generating redundant reasoning paths. The high prevalence of same-answer pairs (over 80% on average) underscores inter-trace redundancy as a critical efficiency bottleneck in parallel reasoning.

Preliminary Experiment. We next investigate whether the similarity between unfinished traces can predict the similarity of their final answers. This capability is crucial for early pruning of redundant paths. We evaluate two approaches for this prediction task: (1) *Shallow Semantic Similarity:* First, we use SentenceBERT (`all-MiniLM-L6-v2` model (Reimers and Gurevych, 2019)) to compute cosine similarity between the first 700 tokens of two traces. This similarity score serve as a feature for binary classification. As shown in Figure 2(b), the ROC curve achieves an AUROC of only 0.58, which is barely better than random guessing (AUROC=0.5). This indicates that surface-level semantic features are insufficient for predicting answer equivalence. (2) *LLM-based Deep Comparison:* To leverage deeper understanding for reasoning traces, we employ `Qwen3-4B-Instruct` in a zero-shot setting. We design a prompt that instructs the model to compare two unfinished traces with 700 tokens and judge whether they will yield the same final answer. The ROC curve for this classifier achieves an AUROC of 0.68, a notable improvement over SentenceBERT. However, this performance remains suboptimal for practical deployment.

These results demonstrate that while LLM-based judgment captures logical equivalence better than semantic similarity, there is significant room for improvement. The suboptimal performance of both approaches motivates our proposed method.

4 DeepPrune

To address the inter-trace redundancy problem in parallel scaling, we propose **DeepPrune**, a two-stage framework that includes offline training of a specialized judge model and online inference-time pruning. As demonstrated in Figure 3, the core idea is that by accurately predicting whether two incomplete reasoning traces will yield identical final answers, we can efficiently prune redundant paths while preserving answer diversity.

4.1 Offline Training

4.1.1 Trace Pair Data Collection

To train our judge model, we construct a dataset of reasoning trace pairs with binary labels indicating

whether they lead to identical final answers. For each input query q, we generate n parallel reasoning traces $\{t_1, t_2, \ldots, t_n\}$ using the same reasoning model. The traces are paired exhaustively, resulting in $\binom{n}{2}$ pairs per query. The similarity label y_{ij} of each pair (t_i, t_j) is based on answer equivalence:

$$y_{ij} = R(o_i, o_j)$$

where $R(o_i, o_j)$ is a reward function that verifies answer equivalence using rule-based methods from DeepScaler, and o_i, o_j are the final answers derived from traces t_i and t_j respectively.

A key challenge is determining how to extract meaningful segments from unfinished traces for early redundancy prediction. We explore two truncation strategies: (1) **Fixed-length prefix:** Truncate the first k tokens from each trace: $t_i^{[1:k]}$ and $t_j^{[1:k]}$. (2) **Reasoning-step alignment:** Extract segments containing the same number of reasoning steps, which can be represented by first k reasoning words like *wait*, *thus*, and *since* that drive the direction of reasoning pathways (Wang et al., 2025a).

Our training data is collected exclusively from `Deepseek-R1-Distill-Llama-8B` outputs using out-of-distribution datasets (AIME 2022, AIME 2023, and MATH 500), while traces from other models and different datasets are reserved for testing cross-model generalization.

4.1.2 Judge Model Training Strategy

We fine-tune `Qwen3-4B-Instruct` as our generative judge model J_θ to predict the similarity label y_{ij} given a pair of unfinished traces (t_i, t_j). The model takes the concatenated trace pair as input and outputs a binary prediction:

$$\hat{y}_{ij} = J_\theta(\text{concat}(t_i, t_j))$$

To address the severe class imbalance where same-answer pairs constitute approximately 80% of the data, we employ two complementary techniques:

Focal Loss. We use focal loss (Lin et al., 2017) to focus training on hard negative examples (different-answer pairs), improving the true negative rate:

$$\mathcal{L}_{focal} = -\alpha_t (1 - p_t)^\gamma \log(p_t)$$

where p_t is the model's estimated probability for the true class, γ modulates the rate at which easy examples are down-weighted, and α_t balances class importance for the 80/20 distribution of our labels.

Oversampling. We oversample the minority class by a factor of 2 to achieve balanced class distribution during training, ensuring the model receives sufficient exposure to diverse reasoning patterns.

4.2 Online Pruning

4.2.1 Early Stopping via Greedy Clustering

During online pruning, we generate multiple parallel traces simultaneously and dynamically prune redundant paths. Let $S = \{t_1, t_2, \ldots, t_N\}$ be the set of N parallel reasoning traces. Our goal is to select a diverse subset $S' \subseteq S$ that maximizes answer diversity while minimizing computational cost.

We propose a greedy clustering algorithm that operates with unfinished traces. The algorithm maintains a set of clusters $C = \{c_1, c_2, \ldots, c_m\}$, where each cluster represents traces that are predicted to yield identical answers $c_j = \{t_1, t_2, \ldots, t_{|c_j|}\}$. For each new trace $t_i \in S$, we compute its average similarity with representative traces sampled from the existing cluster $c_j \in C$:

$$\text{sim}(t_i, c_j) = \frac{1}{p} \sum_{h=1}^{p} J_\theta(t_i, t_h^{(j)})$$

where $t_h^{(j)}$ are randomly sampled top-p traces from cluster c_j with $p = \min(20, |c_j|)$. If $\max_j \text{sim}(t_i, c_j) > \tau$, we assign t_i to the most similar cluster $\arg\max_j \text{sim}(t_i, c_j)$; otherwise, we create a new cluster if the maximum number of clusters K has not been reached. If K is reached, we will terminate the clustering process.

Our approach reduces the number of similarity judgments compared to exhaustive pairwise comparisons, making it suitable for real-time inference.

4.2.2 Majority Voting for Final Answer

After clustering, we need to select a final answer from the remaining traces. Since our judge model may give wrong prediction, we observe two kinds of errors: (1) Most pairs are classified as equivalent, so the largest cluster has too many traces. (2) All trace pairs are predicted as different, therefore each cluster only has one trace. To deal with these situations, we first select the largest cluster c_{max} ($|c|$ means the number of reasoning traces in c):

$$c_{\text{max}} = \arg\max_{c \in C} |c|$$

To conduct voting without too many identical traces, we only let top-q_1 traces in c_{max} to finish reasoning, where $q_1 = \min(|c_{\text{max}}|, 20)$. Besides,

Judge Training Method	Average		Qwen3-4B-Thinking		QwQ-32B		GLM-4.5-Air	
	AUROC	TNR@0.2	AUROC	TNR@0.2	AUROC	TNR@0.2	AUROC	TNR@0.2
Top-500 Tokens	0.8556	0.7720	0.8582	0.7720	0.8435	0.7632	0.8652	0.7808
+ Focal loss	0.8360	0.7327	0.8369	0.7134	0.8309	0.7373	0.8401	0.7473
+ Oversampling	0.7610	0.5232	0.7587	0.4910	0.7509	0.5154	0.7733	0.5632
+ Focal loss & Oversampling	0.8608	0.7698	0.8710	0.7869	**0.8586**	0.7629	0.8528	0.7595
Top-25 Reasoning Words	0.8326	0.6647	0.7948	0.5236	0.8190	0.6587	0.8841	0.8117
+ Focal loss	0.8559	0.7403	0.8434	0.6846	0.8253	0.6842	**0.8989**	0.8522
+ Oversampling	0.7983	0.6762	0.8095	0.6677	0.7917	0.6521	0.7938	0.7089
+ Focal loss & Oversampling	**0.8701**	**0.8186**	**0.8705**	**0.8100**	0.8512	**0.7905**	0.8886	**0.8554**

Table 1: The offline evaluation results of the judge model across different truncation methods and training strategies, which reports the average AUROC and TNR@0.2 metrics for three reasoning models using two truncation types: top-500 tokens and top-25 reasoning words, with the combination of focal loss and oversampling.

if all clusters are singletons, i.e. $|c| = 1, \forall c \in C$, which means the judge model is highly likely wrong, we just sample $q_2 = 32$ traces from S for final reasoning. Finally, we apply majority voting on the final answers of those finished traces:

$$o_{\text{final}} = \text{MajorityVote}(\{o_1, o_2, \ldots, o_q\})$$

where $q = q_1$ or $q = q_2$ depending on situations.

This approach ensures that we can invest computational resources primarily in promising reasoning paths even if the judge model may produce wrong prediction, which reduces token consumption in parallel reasoning while preserving answer quality.

5 Experiments

5.1 Experimental Setup

Settings. Our evaluation includes both offline assessment of the judge model's predictive performance and online testing of the full pruning framework during reasoning tasks. For offline evaluation, we train our judge model on reasoning traces from `Deepseek-R1-Distill-Llama-8B` and evaluate its generalization capability on three distinct reasoning models: `Qwen3-4B-Thinking-2507`, `QWQ-32B`, and `glm-4.5-air`. We use reasoning traces collected from challenging reasoning benchmarks comprising over 1,000 problems. For online reasoning experiments, we evaluate three reasoning models: `DeepSeek-8B` (Guo et al., 2025), `Qwen3-32B` (Yang et al., 2025a), and `GPT-OSS-20B` (Agarwal et al., 2025) across the three benchmarks: AIME 2024 (MAA, 2024), AIME 2025 (AIME, 2025), and GPQA (Rein et al., 2024). We generate 512 parallel reasoning traces per problem for baseline methods and apply DeepPrune with a redundancy threshold $\tau = 0.5$.

Metrics. We employ two categories of evaluation metrics. (1) **Offline Evaluation Metrics:** We assess the judge model's binary classification performance using AUROC (area under the receiver operating characteristic curve) to measure overall classification performance, and TNR@0.2 (true negative rate at false negative rate of 0.2) to evaluate the model's ability to identify diverse reasoning paths while controlling false negatives. (2) **Online Evaluation Metrics:** We measure the end-to-end system performance using token consumption, accuracy (final answer correctness measured by exact match with ground truth), and token reduction percentage reduction in token consumption compared to original consensus sampling (cons@512):

$$\Delta\text{Token}\% = \frac{\text{Tokens}_{\text{new}} - \text{Tokens}_{\text{origin}}}{\text{Tokens}_{\text{origin}}} \times 100\%$$

Baselines. We compare DeepPrune against several competitive baselines: (1) **Sampling Methods:** cons@512 which samples 512 parallel traces with majority voting for self-consistency (Wang et al., 2022), serves as our primary baseline for token reduction calculations. (2) **Confidence-based Pruning Methods:** DeepConf-high and DeepConf-low are confidence-based early stopping methods with high or low threshold for pruning. These baselines represent the state-of-the-art in efficient reasoning methods. We ensure fair comparison by using identical model checkpoints and experimental configurations with the DeepConf (Fu et al., 2025b).

5.2 Offline Experiment Results

As shown in Table 1, we have three observations: (1) Our best configuration, which uses top-25 reasoning words with focal loss and oversampling, achieves superior performance with an average AUROC of 0.8701 and TNR@0.2 of 0.8186 across all

Metric	DeepSeek-8B			Qwen3-32B			GPT-OSS-20B		
	AIME24	**AIME25**	**GPQA**	**AIME24**	**AIME25**	**GPQA**	**AIME24**	**AIME25**	**GPQA**
cons@512†									
Token ($\times 10^8$)	3.55	4.01	9.92	2.00	2.43	7.44	5.57	6.26	-
Accuracy	86.7%	82.3%	72.5%	84.8%	80.1%	72.2%	96.7%	95.4%	-
DeepConf-high†									
Token ($\times 10^8$)	1.45	2.37	6.90	0.88	1.61	4.16	3.07	3.18	-
ΔToken%	-59.0%	-40.9%	-30.4%	-56.0%	-33.7%	-44.1%	-44.8%	-49.2%	-
Accuracy	86.7%	81.4%	72.4%	86.4%	80.2%	72.9%	96.7%	95.3%	-
DeepConf-low†									
Token ($\times 10^8$)	0.78	1.24	3.46	0.66	1.14	3.21	1.11	1.21	-
ΔToken%	-77.9%	-69.0%	-65.1%	-66.8%	-52.9%	-56.9%	**-80.0%**	-80.7%	-
Accuracy	92.5%	86.4%	71.7%	89.5%	80.2%	73.0%	95.7%	96.1%	-
cons@512									
Token ($\times 10^8$)	3.62	4.19	10.9	1.93	2.64	6.94	2.05	2.10	4.60
Accuracy	86.7%	83.3%	66.2%	86.7%	80.0%	70.7%	93.3%	90.0%	70.7%
DeepPrune (ours)									
Token ($\times 10^8$)	**0.42**	**0.35**	**2.54**	**0.26**	**0.23**	-	**0.42**	**0.38**	**2.20**
ΔToken%	**-88.3%**	**-91.6%**	**-76.7%**	**-86.4%**	**-91.4%**	**-%**	-79.6%	**-82.2%**	**-52.5%**
Accuracy	86.7%	83.3%	63.1%	90.0%	90.0%	-%	90.0%	93.3%	68.7%

Table 2: Online experimental results showing token consumption (in $\times 10^8$) and accuracy across three reasoning models on three benchmarks. The table compares different methods including conventional sampling (cons@512), confidence-based approaches (DeepConf-high, DeepConf-low), and the proposed DeepPrune. Token savings relative to cons@512 (ΔToken%) are also provided where applicable. † indicates results taken from the DeepConf paper.

models. This represents a substantial improvement over the preliminary zero-shot LLM judgment (AUROC=0.66) reported in Figure 2(c), validating the necessity of specialized training for redundancy prediction. (2) The comparison between top-500 tokens and top-25 reasoning words reveals a clear advantage for reasoning-aligned truncation. This suggests that structural alignment with reasoning steps provides more reliable signals for predicting answer equivalence compared to fixed-length token windows. (3) The ablation study on training strategies demonstrates the critical importance of addressing class imbalance. The combination of focal loss and oversampling consistently delivers the best performance across both truncation types and all reasoning models. Notably, using oversampling alone significantly degrades performance (AUROC drops to 0.7610 for tokens and 0.7983 for reasoning words), indicating that simply balancing the dataset distribution is insufficient without proper loss weighting. Focal loss alone provides moderate improvements, but the synergistic combination with oversampling yields the most robust results.

5.3 Online Experiment Results

Table 2 presents the online evaluation results, from which we can get several key findings:

Substantial Token Reduction with Minimal Accuracy Loss. DeepPrune achieves remarkable token savings while maintaining competitive accuracy across all experimental settings. Specifically, DeepPrune reduces token consumption by 52.5% to 91.6% compared to the cons@512 sampling baseline. The most significant reductions are observed on AIME datasets, where DeepPrune achieves 79.6%-91.6% token savings with negligible accuracy drop (within 3 percentage points). For instance, on Qwen3-32B with AIME25, DeepPrune reduces tokens by 91.4% while even improving accuracy from 80.0% to 90.0%. This demonstrates that our method effectively identifies and eliminates redundant reasoning paths without compromising solution quality.

Superior Efficiency Compared to Confidence-Based Methods. DeepPrune consistently outperforms confidence-based pruning methods: DeepConf-high and DeepConf-low, in terms of token efficiency. While DeepConf-low achieves substantial token reductions, DeepPrune provides the least token consumption across different configurations. More importantly, DeepPrune maintains more stable accuracy preservation compared to DeepConf-low, e.g., DeepPrune's 90.0% vs DeepConf-low's 80.2% on AIME25 with Qwen3-

Threshold	Qwen3-32B on AIME24				Qwen3-32B on AIME25			
	Greedy Clustering		w. Majority Voting		Greedy Clustering		w. Majority Voting	
	Token($\times 10^8$)	pass@k	Token($\times 10^8$)	ACC	Token($\times 10^8$)	pass@k	Token($\times 10^8$)	ACC
0.75	0.0148	93.3%	0.33	86.7%	0.0282	96.7%	0.31	90.0%
0.63	0.0093	93.3%	0.28	93.3%	0.0265	96.7%	**0.23**	83.3%
0.5	0.0082	93.3%	0.26	90.0%	0.0142	70%	**0.23**	90.0%
0.25	**0.0043**	93.3%	**0.25**	90.0%	**0.0050**	70%	**0.23**	90.0%

Table 3: Performance of DeepPrune with varying redundancy threshold τ on AIME datasets for Qwen3-32B. Token consumption, pass rate and accuracy are reported for two pruning settings: (1) Conduct greedy clustering then retains only one trace per cluster, (2) Perform majority voting to get one final answer with the largest cluster.

32B. This highlights the advantage of our inter-trace redundancy analysis over single-trace confidence estimation methods (Fu et al., 2025b).

Cross-Model and Cross-Dataset Generalization. The consistent performance across three distinct reasoning models (DeepSeek-8B, Qwen3-32B, and GPT-OSS-20B) and three diverse benchmarks (AIME24, AIME25, GPQA) validates the generalizability of our approach. Particularly noteworthy is that the judge model of DeepPrune was trained purely on the reasoning traces of `Deepseek-R1-Distill-Llama-8B`, which means all the tests are out-of-distribution, providing a practical solution for efficient parallel reasoning

5.4 Ablation Study

To better understand the impact of different truncation strategies on the judge model's performance, we conduct an ablation study on the number of extracted top tokens and reasoning words. The results are presented in Figure 4. Comparing Figure 4(a) and Figure 4(b), it is evident that using reasoning words as features generally yields higher AUROC scores and a more pronounced optimal point compared to using raw top tokens. This confirms our hypothesis that extracting semantically rich reasoning words provides a more effective representation for the judge model, leading to better prediction of answer equivalence. The optimal performance is often found at an intermediate number of features (e.g., 500 tokens or 25 reasoning words), suggesting a sweet spot where sufficient context is provided without introducing excessive noise.

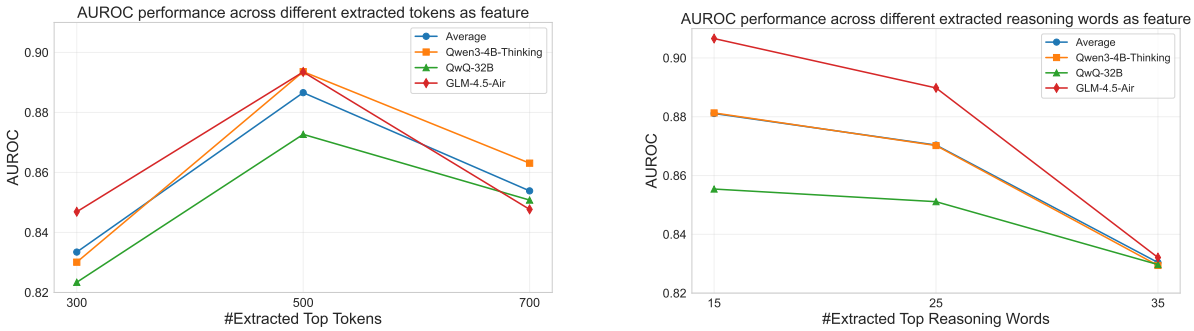
Besides, we analyze the trade-off between efficiency, answer diversity, and final accuracy by varying the redundancy threshold τ in DeepPrune (Table 3). The pass rate measures answer diversity after clustering, while accuracy reflects the voting outcome from the largest cluster. As τ decreases from 0.75 to 0.25, token consumption decreases significantly due to more aggressive pruning. However, this comes at the cost of reduced answer diversity, particularly on challenging problems. On AIME25, pass rate drops from 96.7% to 70% under greedy clustering, indicating that higher thresholds may prune valuable diverse reasoning paths. The majority voting accuracy shows the same trend and the threshold $\tau = 0.5$ provides the best balance.

6 Conclusion

We identify inter-trace redundancy as a major efficiency bottleneck in parallel reasoning, where over 80% of computational resources are wasted on generating equivalent reasoning paths. To address this, we propose DeepPrune, a novel framework that trains a judge model on out-of-distribution data for online greedy clustering to dynamically prune redundant traces while preserving answer diversity. Extensive experiments show that our OOD-trained judge model generalizes strongly to unseen reasoning models, achieving 0.7072 AUROC in offline evaluation, and DeepPrune reduces token consumption by up to 88.5% without hurting accuracy in online experiments. Our work establishes that learned similarity judgment effectively addresses redundancy in parallel scaling, paving the way for more efficient reasoning systems.

Limitations

We acknowledge several limitations of our work. First, due to limited computational resources, our judge model is trained exclusively on reasoning traces from `Deepseek-R1-Distill-Llama-8B`, which may limit its generalization to other model families with distinct reasoning styles. While our cross-model evaluations show promising results, performance could potentially degrade on architectures different from the training distribution. Second, the greedy clustering algorithm, while effi-



(a) Ablation of Token Number

(b) Ablation of Reasoning Word Number

Figure 4: Ablation study on judge model with different truncation strategies for unfinished reasoning traces. We report the classification performance on three reasoning models' trace answer equivalence with different numbers of truncated top tokens (Figure (a)) and different numbers of reasoning words in extracted segments (Figure (b)).

cient, makes locally optimal decisions that may occasionally prune beneficial diverse paths, particularly in complex reasoning scenarios where early similarity is not indicative of final answer equivalence. Third, our method introduces additional computational overhead from the judge model inferences during pruning. Although the overall token reduction is substantial, the relative efficiency gain depends on the cost ratio between the judge and reasoning models. Finally, the optimal redundancy threshold τ may be problem-dependent; while $\tau = 0.5$ works well across our benchmarks, adaptive threshold selection could further improve performance. Addressing these limitations represents promising directions for future work.

Ethical Consideration

We affirm that this work raises no significant ethical concerns. All models and datasets used in our experiments are publicly available with permissible licenses, ensuring proper attribution and compliant usage. Specifically, the reasoning models (DeepSeek-8B[1], Qwen (Yang et al., 2025a), GPT-OSS (Agarwal et al., 2025)) and benchmarks (AIME (MAA, 2024; AIME, 2025), GPQA (Rein et al., 2024)) are widely recognized resources in the research community.

Our research focuses on improving the computational efficiency of reasoning processes through redundancy reduction, without involving sensitive data generation or manipulation. The content processed consists exclusively of mathematical and scientific reasoning tasks, which are devoid of personal, biased, or harmful material.

[1] https://huggingface.co/deepseek-ai/DeepSeek-R1-0528-Qwen3-8B

Acknowledgements

This work is supported by the National Natural Science Foundation of China (625B2101, 62476150). This work is also supported by a grant from the Institute for Guo Qiang, Tsinghua University (2019GQB0003).

References

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

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.

AIME. 2025. Aime problems and solutions.

Anthropic. 2024. Anthropic: Introducing claude 3.5 sonnet.

Daman Arora and Andrea Zanette. 2025. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*.

Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. *arXiv preprint arXiv:2503.05179*.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. 2024a. Are more llm calls all you need? towards

scaling laws of compound inference systems. *arXiv preprint arXiv:2403.02419*.

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*.

Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*.

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

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025b. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

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. *NeurIPS*.

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*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Chen Li, Nazhou Liu, and Kai Yang. 2025. Adaptive group policy optimization: Towards stable training and token-efficient reasoning. *arXiv preprint arXiv:2503.15952*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024. Can language models learn to skip steps? *Advances in Neural Information Processing Systems*, 37:45359–45385.

Yue Liu, Jiaying Wu, Yufei He, Ruihan Gong, Jun Xia, Liang Li, Hongcheng Gao, Hongyu Chen, Baolong Bi, Jiaheng Zhang, et al. 2025. Efficient inference for large reasoning models: A survey. *arXiv preprint arXiv:2503.23077*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. `https://github.com/rllm-org/rllm`. GitHub.

MAA. 2024. American invitational mathematics examination - aime.

Lovish Madaan, Aniket Didolkar, Suchin Gururangan, John Quan, Ruan Silva, Ruslan Salakhutdinov, Manzil Zaheer, Sanjeev Arora, and Anirudh Goyal. 2025. Rethinking thinking tokens: Llms as improvement operators. *arXiv preprint arXiv:2510.01123*.

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

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*.

OpenAI. 2024. Openai: Hello gpt-4o.

Jiayi Pan, Xiuyu Li, Long Lian, Charlie Snell, Yifei Zhou, Adam Yala, Trevor Darrell, Kurt Keutzer, and Alane Suhr. 2025. Learning adaptive parallel reasoning with language models. *arXiv preprint arXiv:2504.15466*.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, et al. 2025. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

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

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*.

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

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

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*.

Siddarth Venkatraman, Vineet Jain, Sarthak Mittal, Vedant Shah, Johan Obando-Ceron, Yoshua Bengio, Brian R Bartoldson, Bhavya Kailkhura, Guillaume Lajoie, Glen Berseth, et al. 2025. Recursive self-aggregation unlocks deep thinking in large language models. *arXiv preprint arXiv:2509.26626*.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024. Soft self-consistency improves language model agents. *arXiv preprint arXiv:2402.13212*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. 2025a. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

Weiqin Wang, Yile Wang, and Hui Huang. 2025b. Ranked voting based self-consistency of large language models. *arXiv preprint arXiv:2505.10772*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

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

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

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

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

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*.

Tong Zheng, Hongming Zhang, Wenhao Yu, Xiaoyang Wang, Xinyu Yang, Runpeng Dai, Rui Liu, Huiwen Bao, Chengsong Huang, Heng Huang, et al. 2025. Parallel-r1: Towards parallel thinking via reinforcement learning. *arXiv preprint arXiv:2509.07980*.

Appendix

A Detailed Analysis of Inter-trace Redundancy

In Section 3.2, Figure 2(a) presents the distribution of redundant traces, showing the "same-answer" pair ratios for four different models. The average percentage of these models might not directly align with a simple arithmetic average of the individual percentages. This is because the amount of parallel reasoning trace data collected for each model varies significantly.

During the trace collection process, we initially sampled 16 responses for each query. However, some Chain-of-Thought traces failed to produce valid answers or did not terminate within the ‘max_length‘ limit. These invalid traces were excluded from the pair calculation. Furthermore, while our full dataset comprised over 760 problems from benchmarks including GPQA, AIME24, AIME25, and Math500, due to computational resource constraints, we were only able to run all 760 queries on `Deepseek-R1-Distill-llama-8b`. For other models, we sampled approximately over 100 problems. This led to the following total pair counts and similarity ratios for each model in Table 4.

Model	Total Pairs	Same Answer Pairs	Similarity Ratio
GLM-4.5-Air	11,870	11,213	0.9447
QwQ-32B	13,785	12,569	0.9118
Deepseek-R1-Distill-llama-8b	80,760	61,505	0.7616
Qwen3-4B-Thinking-2507	13,800	12,852	0.9313
Average	120,215	98,139	0.8164

Table 4: Detailed statistics of collected reasoning trace pairs and their similarity ratios across different models.

Given the large disparity in the number of total pairs, particularly the substantial contribution from `Deepseek-R1-Distill-llama-8b`, a weighted average would be necessary to accurately reflect the overall inter-trace redundancy across the combined dataset. Our analysis in Section 3.2 is based on the aggregate distribution, rather than a simple average of individual model ratios.

B Reconciliation of Online Experiment Results

B.1 Differences in cons@512 and DeepConf Baselines

In Table 2 of our online experiments, the reported cons@512 and DeepConf baseline results might show slight differences compared to those originally published in the `DeepConf` paper. We conducted our experiments using the exact same three reasoning models (`DeepSeek-8B`, `Qwen3-32B`, and `GPT-OSS-20B`) and the same three benchmarks (AIME 2024, AIME 2025, and GPQA) as in the `DeepConf` paper.

The cons@512 method, which involves sampling 512 responses, inherently has a degree of randomness. To minimize these discrepancies and ensure a fair comparison, we meticulously aligned our experimental setup with that described in the `DeepConf` codebase. This included using identical sampling temperatures and ‘max_length‘ settings for each model’s hyperparameters, and employing vllm (Kwon et al., 2023) as the inference engine.

Despite these efforts, minor differences in the final evaluation results persisted. To further align our results, we initially sampled 640 responses (512 + 128) and then re-sampled 512 responses. This re-sampling process was performed to match the pass@1 metrics for each model and dataset as closely as possible to those reported in the original `DeepConf` paper, ensuring the differences were within 3 percentage points. The aligned pass@1 values are presented in Table 5.

This careful alignment allows for a more direct comparison of DeepPrune’s performance against established baselines.

B.2 Missing DeepConf Result for GPT-OSS-20B on GPQA

In Table 2, the entry for DeepConf on GPT-OSS-20B for the GPQA dataset is marked as empty (-). This is because the original `DeepConf` paper did not report experimental results for this specific model-dataset

Model	Dataset	DeepPrune Pass@1	DeepConf Pass@1
Qwen3-32B	AIME_2025	69.28	71.7
Qwen3-32B	AIME_2024	80.10	80.6
Qwen3-32B	GPQA	68.24	68.9
DeepSeek-R1-0528-Qwen3-8B	AIME_2025	74.57	76.9
DeepSeek-R1-0528-Qwen3-8B	AIME_2024	83.08	83.0
DeepSeek-R1-0528-Qwen3-8B	GPQA	59.74	62.8
GPT-OSS-20B	AIME_2025	77.54	-
GPT-OSS-20B	AIME_2024	80.72	-
GPT-OSS-20B	GPQA	66.18	-

Table 5: Comparison of pass@1 metrics with the same values taken from the DeepConf paper after alignment procedure. Differences are within 3 percentage points.

combination. Due to our limited computational resources, we were unable to conduct this additional experiment to fill the gap.

B.3 Dataset and Model Specifications

To ensure full transparency and reproducibility, we explicitly state the versions and sources of the datasets and models used in our experiments. If not stated below, then the name of the dataset or model should be its full name.

- All references to **GPQA** in this paper, including those in our online experiments (Table 2) and trace collection, refer specifically to the **GPQA Diamond** subset, which consists of 198 problems. This is consistent with the dataset used in the `DeepConf` paper and can be accessed at `https://huggingface.co/datasets/fingertap/GPQA-Diamond`.
- The model referred to as **DeepSeek-8B** throughout this paper, corresponds to the `DeepSeek-R1-0528-Qwen3-8B` model. This is the same model variant employed in the `DeepConf` paper and is publicly available at `https://huggingface.co/deepseek-ai/DeepSeek-R1-0528-Qwen3-8B`. And **Qwen3-4B-Thinking** refers to `https://modelscope.cn/models/Qwen/Qwen3-4B-Thinking-2507`.

C Hyperparameters

Table 6 summarizes the key hyperparameters used in the DeepPrune framework and their default values, along with a brief explanation of their meaning.

Symbol	Default Value	Description
k	500	Number of tokens for fixed-length prefix truncation
k	25	Number of reasoning words for aligned segment truncation
γ	2.0	Focal loss focusing parameter
α_t	0.25	Focal loss class-balancing parameter
τ	0.5	Similarity threshold for greedy clustering
K	16	Maximum number of clusters
p	20	Number of sampled traces for cluster similarity calculation
q_1	20	Max traces to finish reasoning from the largest cluster
q_2	32	Number of traces to sample for reasoning if all clusters are singletons

Table 6: Key hyperparameters of the DeepPrune framework. There are two k symbols because each of them will not exist when the other is used since we can only use one truncation strategy.

D Computational Resources

The majority of our experiments, including the training of the judge model and a significant portion of the testing, were conducted on 4 NVIDIA A100 GPUs with 80GB memory each. In the later stages of the experiments, we also utilized a server equipped with 8 H20 GPUs for approximately two days. Our computational resources were relatively limited, especially considering the requirement to generate 512 responses from large language models for baseline comparisons.

E Discussion on Answer Diversity and Majority Voting

E.1 Comparison of Diversity with Baselines

In the introduction, we highlight that a limitation of confidence-based early stopping is its potential impact on answer diversity, which DeepPrune aims to preserve. However, in our online experiments (Table 2), we primarily compare against baselines using accuracy rather than an explicit diversity metric like pass@k.

In reasoning tasks, answer diversity is often quantified by pass@k, which measures the proportion of samples where at least one of the top-k retained answers is correct. Prior methods like `DeepConf` and cons@512 typically aggregate multiple traces to produce a single final answer, effectively reducing k to 1 for their final output. Therefore, a direct pass@k comparison with these methods would be unfair, as our approach inherently produces a set of diverse reasoning paths (clusters) from which multiple candidates could be drawn, leading to a k value greater than 1.

Our approach, after clustering, theoretically retains distinct reasoning paths in different clusters. Each cluster's representative trace could contribute to a pass@k calculation, where k would typically be greater than one (equal to the number of clusters). This inherent diversity in our retained traces is evident in Table 3, where higher τ thresholds lead to higher pass rates under greedy clustering, indicating more diverse paths are preserved. To ensure a fair comparison on a single-answer basis, our main online evaluation focuses on the final accuracy obtained after majority voting, as this is the metric that DeepConf and cons@512 also optimize for.

E.2 Analysis of Low Pass Rate with Low Thresholds

A notable observation in Table 3 is that for Qwen3-32B on AIME25, the pass@k for "Greedy Clustering" at lower redundancy thresholds (0.5 and 0.25) is 70.0%, which appears lower than the corresponding "w. Majority Voting" setting's accuracy. This seemingly counter-intuitive result can occur because, at very low similarity thresholds, the judge model becomes highly permissive, predicting a large number of trace pairs as having the same answer.

This permissiveness causes many traces to be grouped into a few large clusters, leading to an uneven distribution of traces across clusters. When we calculate the pass@k metric for "Greedy Clustering" we typically select one representative trace from each unique cluster to contribute to the diversity count (i.e., k equals the number of distinct clusters). If most traces are concentrated in only a few clusters, the effective k becomes very small. In such scenarios, even if the individual clusters contain correct answers, the limited number of distinct clusters (and thus k value) can result in a lower pass@k percentage, as it does not fully capture the potential for correctness from the overall set of generated traces.

Conversely, for the "w. Majority Voting" results, our strategy in these low-threshold scenarios is to sample up to $q_1 = 20$ traces from the largest cluster to perform majority voting (as described in Section 4.2.2). This allows us to leverage a greater number of individual traces to reach a consensus, ensuring a more robust final answer and potentially leading to higher accuracy, even when the overall number of distinct clusters and pass@k is low. This mechanism helps maintain effective accuracy by focusing computational resources on the most prominent reasoning paths, despite a reduced perceived diversity at the cluster level when our judge model dose not act perfectly.

E.3 Rationale for Majority Voting

We employ majority voting in DeepPrune for two primary reasons:

1. **Effectiveness and Common Practice:** Majority voting is a widely adopted and empirically effective method for aggregating multiple reasoning traces to derive a robust final answer, as demonstrated by pioneering works like Self-Consistency (Wang et al., 2022). It helps mitigate errors from individual traces and leverages the collective intelligence of diverse reasoning paths.

2. **Fair Comparison with Baselines:** To enable a direct and fair comparison of final answer accuracy with methods like DeepConf that also produce a single aggregated answer, we needed a mechanism to consolidate the diverse traces retained by DeepPrune into one final prediction. While our method inherently preserves inter-trace diversity for potential pass@k evaluations (as explored in Table 3), majority voting allows us to align with the single-answer output paradigm of competitive baselines.

It is important to note that DeepPrune is highly flexible and can be integrated with other aggregation strategies. For instance, instead of simple majority voting, one could employ a selection model (Moshkov et al., 2025) to choose the best answer from the diverse set of clusters. Furthermore, DeepPrune is orthogonal to existing method; for example, after our method reduces inter-trace redundancy, a confidence-based filter like DeepConf could be applied within each remaining cluster or across selected representatives to further refine the final answer. Due to our limited resources, we leave the exploration of these alternative aggregation strategies and combinations for future work.