FROST: FILTERING REASONING OUTLIERS WITH ATTENTION FOR EFFICIENT REASONING

Haozheng Luo[†♮*] **Md Zahid Hasan**[♮‡*] **Zhuolin Jiang**[♮] **Yan Chen**[†] **Soumalya Sarkar**[♮]

[†]Department of Computer Science, Northwestern University, Evanston, IL 60208 USA
[♮]RTX Technology Research Center (RTRC), East Hartford, CT 06118 USA
[‡]Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA

hluo@u.northwestern.edu {zhuolin.jiang,soumalya.sarkar}@rtx.com
zahid@iastate.edu ychen@northwestern.edu

ABSTRACT

Large reasoning models often spend many tokens on low-value self-checking steps that barely affect the final answer. We formalize these steps as **reasoning outliers** and target them directly. Our method, **FROST**, modifies the attention mechanism by replacing vanilla $\mathrm{Softmax}$ with $\mathrm{Softmax}_1$, which contracts low-attention tails while keeping high-attention signals. In practice, this shifts probability mass toward the few reasoning sentences that drive answer prediction and suppresses redundant traces at the sentence level. To make this change usable for existing models, we pair the activation swap with lightweight SFT, avoiding expensive retraining from scratch. We provide a theoretical analysis showing why sentence-level outlier suppression can be achieved without degrading core reasoning ability. Across four math benchmarks and two strong reasoning backbones (Phi-4-Reasoning and GPT-oss-20B), FROST consistently improves the accuracy–efficiency tradeoff over prior SOTA methods (e.g., TALE and ThinkLess). On average, it reduces the number of generated tokens by **69.68%** and improves accuracy by **26.70%** relative to the base model. It also yields cleaner attention statistics, lowering maximum activation infinity norm by **15.97%** and average kurtosis by **91.09%**.

1 INTRODUCTION

Large reasoning models (LRMs) have made rapid progress on complex reasoning tasks, including mathematics (Luo et al., 2025a; Yang et al., 2024; Shao et al., 2024), coding (Ding et al., 2024a;b), and scientific question answering (Comanici et al., 2025; Hurst et al., 2024). Despite these gains, they often produce unnecessarily long reasoning traces. In many cases, a substantial fraction of the generated text consists of repetitive self-verification or low-impact intermediate steps that increase latency and token usage without improving correctness. This inefficiency becomes especially problematic as reasoning models are deployed under strict inference-time and budget constraints.

Recent work on efficient reasoning has explored both token-level and sentence-level compression. Token-level methods such as TALE (Han et al., 2025) and R2R (Fu et al., 2025) can reduce length aggressively, but they may remove tokens inside otherwise useful reasoning sentences. Sentence-level methods such as DRP (Jiang et al., 2025b) and GRPO-S (Tan & Pan, 2025) provide a more structured alternative, yet often rely on iterative refinement or extra optimization loops that increase training and inference overhead.

In this work, we ask a simple question: can we improve reasoning efficiency by removing low-impact reasoning *before* generation becomes excessively long? We formalize these low-impact steps as **reasoning outliers**: reasoning sentences that receive weak attention and contribute little to final answer formation. This definition is consistent with findings that critical reasoning traces tend to

[*]This work was carried out at the RTX Technology Research Center during a Summer 2025 internship. These authors contributed equally to this work.

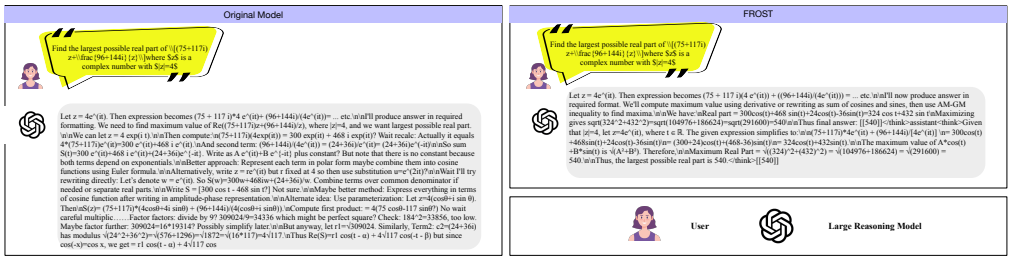

Figure 1: **Illustration of reasoning outliers in GPT-OSS-20B.** The example highlights long reasoning traces with low-contribution steps, motivating sentence-level outlier removal.

show higher entropy (Tan & Pan, 2025), while low-value traces are often associated with weaker informativeness (Wang et al., 2025; Fu et al., 2025).

Based on this view, we propose **FROST**, an attention-level intervention that replaces vanilla $\mathrm{Softmax}$ with $\mathrm{Softmax}_1$ to contract low-attention tails while preserving high-attention signals. To make this intervention practical for existing checkpoints, we combine the activation replacement with lightweight supervised fine-tuning (LoRA-based SFT), rather than full retraining. This design choice follows prior activation-based outlier-suppression insights (Luo et al., 2025b; Hu et al., 2024; Xiao et al., 2024) and adapts them to efficient reasoning.

Empirically, we evaluate FROST on four math benchmarks using two strong reasoning backbones (Phi-4-Reasoning and GPT-oss-20B), and compare against strong efficiency baselines including TALE and ThinkLess. FROST consistently improves the accuracy–efficiency tradeoff, achieving on average **69.68%** fewer generated tokens and **26.70%** higher accuracy than base models. We also observe cleaner internal attention statistics, including lower maximum activation infinity norm (**15.97%**) and lower average kurtosis (**91.09%**).

Contributions.

- We introduce **reasoning outliers** as a concrete target for efficient reasoning, defined through weak-attention, low-impact reasoning traces.
- We propose **FROST**, a simple attention-level method ($\mathrm{Softmax} \rightarrow \mathrm{Softmax}_1$) with lightweight LoRA-based SFT that can be applied to existing LRMs.
- We provide theoretical support for sentence-level outlier suppression under the proposed activation design.
- We show consistent gains in both efficiency and accuracy across multiple reasoning backbones and benchmarks, with substantial improvements in token usage and attention metrics related to outliers.

2 Related Work

Reasoning Models. Recent frontier LLMs, including DeepSeek-R1 (Guo et al., 2025), OpenAI o1 (Jaech et al., 2024), and Gemini 2.0 Pro (Team et al., 2023), demonstrate strong performance on mathematical and logical reasoning (Hao et al., 2024). Current progress is typically organized into two paradigms (Ke et al.): *inference-time scaling* and *learning-to-reason*. Inference-time scaling methods increase effective reasoning depth through prompting or search, including few-shot/in-context prompting (Brown et al., 2020), Chain-of-Thought (CoT) prompting (Wei et al., 2022), and explicit search-planning procedures (Besta et al., 2024). CoT has become a default building block because it improves intermediate trace quality and interpretability (Wei et al., 2022), and is often combined with interactive decomposition and tool-use patterns such as ReAct (Yao et al., 2023), Self-Ask (Press et al., 2023), and agentic pipelines (Pan et al., 2025; 2024).

In parallel, learning-to-reason methods optimize model behavior during post-training, e.g., RLHF (Ouyang et al., 2022), DPO (Rafailov et al., 2023), GRPO (Ramesh et al., 2024), and EBM-style reasoners (Jiang et al., 2025a). However, stronger reasoning behavior frequently comes with *overthinking*: models generate long, repetitive chains with unnecessary branches and traceback-style self-checking (Hou et al., 2026; Chen et al., 2025). This behavior increases inference cost and can degrade both answer quality and safety under practical constraints (Cuadron et al., 2025; Kumar et al.,

2025). These observations motivate methods that improve reasoning efficiency *without* sacrificing correctness.

Efficient Reasoning Methods. Prior work on efficient reasoning can be grouped into prompt-based control, supervised fine-tuning (SFT), and reinforcement learning (RL) (Sui et al., 2025). Prompt-based methods impose explicit generation budgets or stopping rules (Liu et al., 2025; Xu et al., 2025a; Han et al., 2025); TALE is a representative example that uses token-budget control per instance (Han et al., 2025). SFT methods train on compressed traces to encourage concise generation (Ma et al., 2025; Xia et al., 2025a), with DRP as a representative pruning-based distillation approach (Jiang et al., 2025b). RL methods shape length-accuracy trade-offs through reward design (Hou et al., 2026; Li et al., 2025; Yi et al., 2025); for example, Chia et al. (2024) combine reference and exploration signals to promote favorable reasoning branches.

Recent attention-centric studies (Choi et al., 2025; Cai et al.) are related but focus on KV-cache efficiency. Think Clearly (Choi et al., 2025) uses sentence-level attention spikes near the end-of-thinking token to prune redundancy, and R-KV (Cai et al.) compresses redundant attention interactions. In contrast, our analysis attributes sentence-level contributions to final-answer formation, emphasizing functional contribution rather than cache redundancy.

3 Reasoning Outlier

We analyze how LRMs allocate attention over reasoning traces. We then quantify how different trace components contribute to final-answer prediction, and finally define and characterize reasoning outliers.

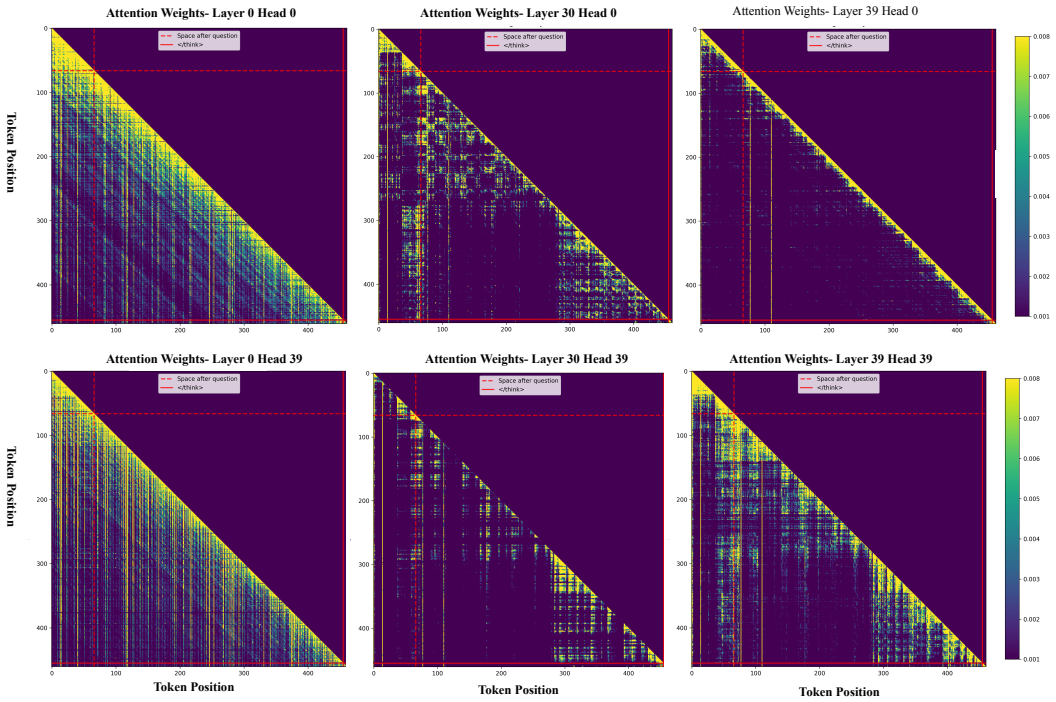

Figure 2: **Attention patterns of reasoning tokens.** Attention maps from Phi-4-Reasoning on a GSM8K example (Abdin et al., 2025; Cobbe et al., 2021) at layers 0, 30, and 39, shown for the first head (top row) and last head (bottom row). Yellow denotes higher attention, while blue indicates lower attention. Shallow layers exhibit nearly uniform contributions, whereas deeper layers and later heads focus on specific tokens with stronger influence.

3.1 Attention Patterns of Reasoning Traces

We begin by examining how attention is allocated across generated reasoning traces in representative LRMs, including DeepSeek-R1 (Guo et al., 2025), Phi-4 (Abdin et al., 2025), and GPT-4o (Hurst et al., 2024). Since these models generate text autoregressively, the attention pattern induced during prediction provides a direct view of which parts of the trace are most likely to matter.

Formally, let the reasoning trace be $T = [t_1, t_2, \ldots, t_n]$, where each t_i denotes a token. We write the attention matrix as

$$A = [a_{ij}], \qquad a_{ij} = \text{AttentionWeight}(t_i, t_j),$$

where a_{ij} measures how strongly token t_i attends to token t_j.

To make this concrete, we analyze a GSM8K example (Cobbe et al., 2021) generated by Phi-4-Reasoning (Abdin et al., 2025). The heatmaps in Figure 2 reveal a clear progression: early layers spread attention relatively evenly across the trace, whereas deeper layers and later heads concentrate attention on a much smaller set of tokens. These high-attention tokens tend to align with pivotal reasoning steps and the final answer. These patterns imply that the model gradually filters broad contextual information into a compact set of decision-relevant signals.

3.2 Impact of Reasoning Trace on Model Prediction

We next examine how different parts of a generated solution contribute to answer formation. To do so, we measure the total attention weight assigned to the final-answer token `</think>`, using it as a proxy for the influence of each trace component on the model's prediction.

We decompose each generated solution into three parts: the question Q, intermediate reasoning steps $R_1, R_2, \ldots, R_m$, and the final answer A. For any component, we aggregate its attention contribution to the first token of the final answer as $W_{\text{trace}} = \sum_{t_i \in T_{\text{trace}}} a_{iA}$, where T_{trace} denotes the set of tokens belonging to that component and a_{iA} is the attention weight from token t_i to the `</think>` token. This quantity captures how much each component contributes to the final answer generation.

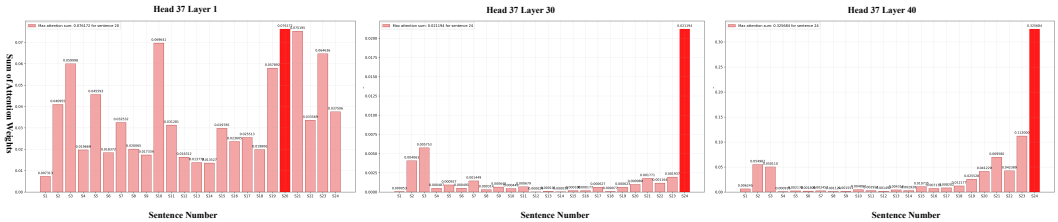

Figure 3: **Total attention directed to the final-answer token `</think>` from different reasoning-trace components.** We report component-wise total attention for a GSM8K sample using Phi-4-Reasoning at layers 1, 30, and 40. A small subset of traces contributes strongly, whereas many traces contribute marginally, especially in deeper layers.

As shown in Figure 3, reasoning components contribute unevenly to final-answer formation: a few traces dominate, while many have negligible effect.

3.3 Reasoning Outliers: Definition and Characterization

The analyses above indicate that not all generated reasoning content is equally useful. A substantial portion of the trace contributes only weakly to final-answer formation, even though it still consumes tokens and inference time. In practice, these low-impact traces often appear as verification loops, self-checking, or the repetition of earlier steps. One plausible reason (Sui et al., 2025) is that current training and prompting strategies often encourage longer chains of thought to preserve accuracy, which can inadvertently promote redundant or irrelevant reasoning.

Motivated by this observation, we define **reasoning outliers** as reasoning traces that receive low attention and make negligible contributions to final-answer prediction. To suppress such traces, we draw on their connection to attention outliers (Luo et al., 2025b; Hu et al., 2024) and adopt Softmax_1:

$$\text{Softmax}_1(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j) + 1}, \tag{1}$$

where x_i is the attention score of token t_i.

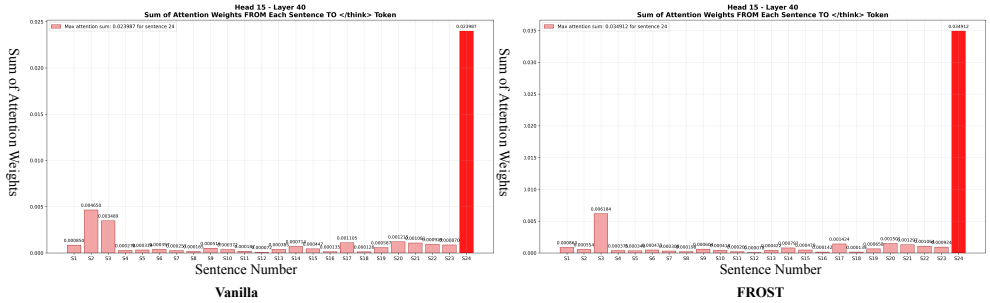

Figure 4: **Theoretical analysis of reasoning-outlier removal.** Using Phi-4-Reasoning, we compare attention distributions before and after applying $\mathrm{Softmax}_1$ (FROST). After outlier suppression, attention mass is concentrated more on critical traces while low-impact traces are reduced, indicating preserved or improved reasoning focus.

Theoretical Analysis. We next examine why $\mathrm{Softmax}_1$ can reduce reasoning outliers without compromising the model's core reasoning process. Using Phi-4-Reasoning (Abdin et al., 2025) on a GSM8K example (Cobbe et al., 2021), we compare the last-layer attention distribution of head 15 under standard attention and under $\mathrm{Softmax}_1$ (FROST). As shown in Figure 4, $\mathrm{Softmax}_1$ attenuates low-impact traces while preserving, and in some cases sharpening, attention on critical reasoning tokens. This supports its use as the central mechanism behind our efficient-reasoning approach. Full theoretical details appear in Section 5.

4 FROST

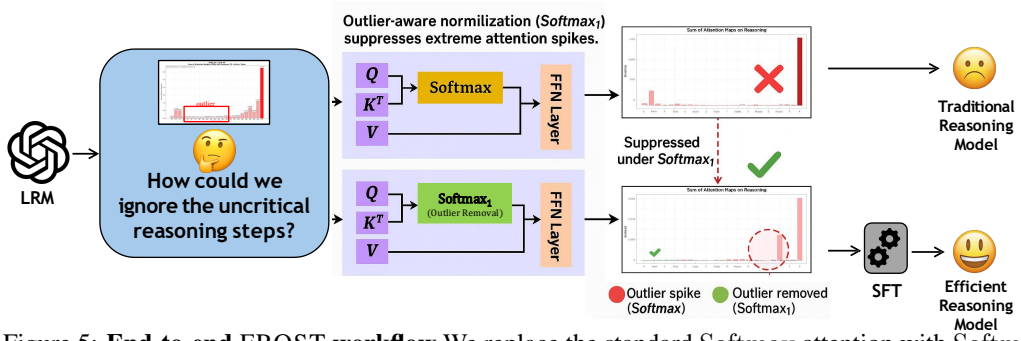

Figure 5: **End-to-end** FROST **workflow** We replace the standard $\mathrm{Softmax}$ attention with $\mathrm{Softmax}_1$ to trim low-impact reasoning traces. The model is then fine-tuned with SFT to adapt the new attention pattern. As a result, the model produces shorter reasoning paths while retaining the most important steps for the final answer.

To improve reasoning efficiency, we train LRMs with supervised fine-tuning (SFT) under an attention outlier-removal mechanism, as illustrated in Figure 5.

During SFT, we train on data for mathematical reasoning that includes step-by-step solutions and final answers. We replace vanilla $\mathrm{Softmax}$ with $\mathrm{Softmax}_1$ (eq. (1)) so that low-attention tails are contracted and critical traces receive relatively higher emphasis. Unlike prior outlier-removal approaches that require training from scratch (Hu et al., 2024) or multi-stage continual adaptation (Luo et al., 2025b), our approach adapts existing checkpoints in a lightweight way. We optimize with cross-entropy loss and apply LoRA (Hu et al., 2021) to reduce training cost. This combination yields a practical path to efficient reasoning: minimal parameter updates, shorter traces, and preserved reasoning quality.

5 Theoretical Analysis

In this section, we provide a concise theoretical analysis showing that $\mathrm{Softmax}_1$ can suppress reasoning outliers at the sentence level in LRMs. Our goal is to formalize deployment-time suppression in efficient reasoning consistent with the empirical trend in Figure 4.

Setup. Let a token sequence be partitioned into sentences $\{S_i\}_{i=1}^m$. For a query $q \in \mathbb{R}^d$ and keys $\{k_t\} \subset \mathbb{R}^d$, define token compatibilities $z_t = \mathrm{Softmax}_1(\frac{\langle q, k_t \rangle}{\sqrt{d}})v_t$, where t indexes tokens in sentence S_i and $v_t \in \mathbb{R}^d$ is the corresponding token value. Let $\phi : \mathbb{R}^{|S_i|} \to \mathbb{R}$ be a *monotone* pooling operator (e.g., sum/mean/logsumexp/max). Define sentence scores $s_i = \phi\big(\{z_t\}_{t \in S_i}\big)$ and $s = (s_1, \ldots, s_m) \in \mathbb{R}^m$. Define the probability simplex $\Delta^{m-1} = \Big\{ \alpha \in \mathbb{R}^m \,\Big|\, \alpha_i \geq 0,\ \sum_{i=1}^m \alpha_i = 1 \Big\}$.

Assumption 5.1 ($\mathrm{Softmax}_1$ operator)**.** There exists a $\mathrm{Softmax}_1$ mapping $\sigma_1 : \mathbb{R}^m \to \Delta^{m-1}$ such that:

1. **Order preservation:** If $x_i \geq x_j$ then $\sigma_1(x)_i \geq \sigma_1(x)_j$.
2. **Shift invariance:** $\sigma_1(x + c\mathbf{1}) = \sigma_1(x)$ for all $c \in \mathbb{R}$.
3. **Tail contraction:** There exists $\kappa \in (0, 1)$ such that for all $x \in \mathbb{R}^m$, $\frac{\|\sigma_1(x)\|_\infty}{\mathrm{median}(\sigma_1(x))} \leq \kappa \frac{\|x\|_\infty}{\mathrm{median}(x)}$.
4. **Smoothness and positivity:** σ_1 is continuously differentiable on $\mathbb{R}^m$ and $\sigma_1(x)_i > 0$ for all finite x.

We write the sentence-level attention as $\alpha = \sigma_1(s) \in \Delta^{m-1}$ and the layer output as $y = \sum_{i=1}^m \alpha_i v_i$, which α_i and v_i are attention probabilities and token values corresponding to sentence s_i. Assume $\|v_i\|_\infty \leq B_v$ and that all linear maps used below have finite operator norms B_v, which is a constant.

Lemma 5.1 (Monotone pooling preserves sentence dominance)**.** If ϕ is monotone coordinatewise, then for any i, j, $\big(\forall t \in S_i, \exists t' \in S_j : z_t \geq z_{t'}\big) \implies s_i \geq s_j$. Consequently, by Assumption 5.1(P1), $\alpha_i = \sigma_1(s)_i \geq \sigma_1(s)_j = \alpha_j$.

Proof. See Section C.1 for a detailed proof. □

Theorem 5.1 ($\mathrm{Softmax}_1$ suppresses sentence-level attention outliers)**.** Let $s = (s_1, \ldots, s_m)$ be the sentence scores built via a monotone pooling ϕ. If s is heavy-tailed (e.g., $\|s\|_\infty / \mathrm{median}(s) \gg 1$), then for $\alpha = \sigma_1(s)$

$$\frac{\|\alpha\|_\infty}{\mathrm{median}(\alpha)} \leq \kappa \cdot \frac{\|s\|_\infty}{\mathrm{median}(s)} \qquad \text{for some } \kappa \in (0, 1), \tag{2}$$

so the relative dominance of outliers contracts at the sentence level.

Proof. See Section C.2 for a detailed proof. □

Theorem 5.2 (Deployment-time suppression of low-attention sentences)**.** Let the output logits be $\ell = W_o y$ with $\|W_o\|_{\mathrm{op}} =: B_o$ and $\|v_i\| \leq B_v$. For a sentence i with $\alpha_i \leq \varepsilon$, its one-layer contribution to logits is bounded by

$$\|\Delta \ell_i\| = \|W_o(\alpha_i v_i)\| \leq B_o\, \varepsilon\, \|v_i\| \leq B_o\, B_v\, \varepsilon. \tag{3}$$

For L stacked layers with Jacobians $\{J_\ell\}_{\ell=1}^L$ and $\|J_\ell\|_{\mathrm{op}} \leq B_\ell$,

$$\|\Delta \ell_i^{(L)}\| \leq \varepsilon \Big(\prod_{\ell=1}^L B_\ell \Big) B_v\, B_o. \tag{4}$$

Since the $\mathrm{Softmax}_1$ map $\mathrm{sm} : \mathbb{R}^V \to \Delta^{V-1}$ is 1-Lipschitz in the $\ell_\infty \to \ell_1$ norm,

$$\big\|\mathrm{sm}(\ell + \Delta \ell_i^{(L)}) - \mathrm{sm}(\ell)\big\|_1 \leq \|\Delta \ell_i^{(L)}\| \leq B_o\, B_v \Big(\prod_{\ell=1}^L B_\ell \Big) \varepsilon. \tag{5}$$

Let $B := \max_{\ell \in [L]} B_\ell$ be the largest operator norm over L layers. Then

$$\Big\|\mathrm{sm}(\ell + \Delta \ell_i^{(L)}) - \mathrm{sm}(\ell)\Big\|_1 \leq B_o B_v B^L \varepsilon = O\big(B_o B_v B^L \varepsilon\big). \tag{6}$$

In practice B_o, B_v, B are approximately constant, so the bound reduces to $O(\varepsilon)$. Therefore, low-attention sentences are effectively skipped at inference.

Proof. See Section C.3 for a detailed proof. □

6 EXPERIMENTAL STUDIES

We conduct a comprehensive empirical evaluation of FROST for efficient reasoning on GPT-oss (Agarwal et al., 2025), Magistral-Small-1.1 (Rastogi et al., 2025), and Phi-4-Reasoning (Abdin et al., 2025). Each setting is run with three random seeds, and we report the average and standard deviation for each metric in our tables.

Table 1: **FROST vs. other Efficient Reasoning Models.** We compare FROST with four baseline methods on GSM8K, MATH500, AIME24, and Minerva. For each method, we report Pass@1 and generated token count (#Tk), with variance omitted because it remains consistently $\leq 2\%$. Best results are shown in **bold**, and second-best results are underlined. Overall, FROST gives the strongest accuracy–efficiency tradeoff and improves over the base model by 26.70% in accuracy while reducing token usage by 69.68%.

Type	Method	GSM8K		MATH500		AIME24		Minerva		$\Delta_{\text{Pass@1}}$	$\Delta_{\#\text{Tk}}$
		Pass@1	#Tk	Pass@1	#Tk	Pass@1	#Tk	Pass@1	#Tk		
Phi-4 -Reasoning	Base	0.9242	1017.70	0.5480	1721.95	0.0667	1017.70	0.2500	1898.86	0.000	0.00
	TALE	**0.9500**	1716.60	0.5800	1874.43	0.2900	2069.97	0.2627	2093.17	**+0.074**	+524.49
	DRP	0.8340	721.00	**0.6200**	2122.00	**0.3333**	6135.00	0.2701	1289.50	+0.067	+1152.69
	SelfBudgeter	0.9189	1507.14	0.5347	1195.18	0.1342	1372.83	0.2357	2618.23	+0.009	+259.30
	ThinkLess	0.9279	1421.90	0.5414	1101.21	0.1608	1405.40	0.2575	1708.70	+0.025	-4.75
	Ours	0.9311	**154.33**	0.5980	**344.37**	0.2667	**899.80**	**0.2716**	**401.19**	+0.070	**-964.13**
GPT-OSS-20B	Base	0.8704	1275.23	0.5400	1575.36	0.1333	**1003.57**	0.2574	1586.95	0.000	0.00
	TALE	0.8283	2664.41	0.5454	3878.87	0.2000	1354.67	0.2700	3262.47	+0.011	+1430.33
	DRP	0.7880	902.50	**0.6146**	4137.00	**0.2245**	4983.00	0.2715	1885.15	+0.024	+1616.64
	SelfBudgeter	0.8610	1850.00	0.5340	2285.00	0.1320	1256.00	0.2550	1298.00	-0.005	+312.47
	ThinkLess	0.8740	1785.00	0.5410	2206.00	0.1600	1205.00	0.2580	1220.00	+0.008	+244.22
	Ours	**0.8764**	**377.17**	0.5800	**680.89**	0.1667	1009.60	**0.2794**	**691.71**	**+0.025**	**-669.94**
Magistral -Small-1.1	Base	0.6075	2664.41	0.1480	1389.89	0.0000	537.13	0.0699	1288.04	0.000	0.00
	TALE	0.7146	1516.86	**0.3040**	723.91	0.0333	967.43	0.1544	748.18	**+0.095**	-480.77
	DRP	0.6500	902.50	0.2100	1680.33	0.0450	1350.77	0.1120	1604.22	+0.048	-85.41
	SelfBudgeter	0.6900	1850.00	0.2300	1520.00	0.0520	1256.00	0.1300	1298.00	+0.069	+11.13
	ThinkLess	0.7200	1785.00	0.2500	1405.00	0.0600	1205.00	0.1450	1220.00	+0.087	-66.12
	Ours	**0.7551**	**137.55**	**0.3040**	**98.20**	**0.0974**	**149.93**	**0.1551**	**109.23**	**+0.122**	**-1346.14**

Models. We use Phi-4-Reasoning (Abdin et al., 2025), Magistral-Small-1.1 (Rastogi et al., 2025), and GPT-oss (Agarwal et al., 2025) as backbone models. Concretely, we start from Phi-4-Reasoning*, Magistral-Small-1.1*, and GPT-oss-20B-finetune*, and fine-tune them with SFT under the FROST setting on math reasoning data.

Datasets. Similar to the setup in Zhao et al. (2025a), we use OpenR1 (Hugging Face, 2025) for training. For evaluation, we use four out-of-domain math benchmarks: GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2024), AIME24 (of America, 2024), and Minerva (Dyer & Gur-Ari, 2022). These datasets are commonly used to evaluate Efficient Reasoning in mathematical problems.

Metrics. To evaluate our proposed method, we report pass@1 as the accuracy metric and use the number of tokens in the model-generated response to measure token efficiency.

Baselines. We compare against five representative efficient-reasoning methods: **(1) TALE (Han et al., 2025):** a prompt-based token budgeting method to generate concise reasoning responses. **(2) DRP (Jiang et al., 2025b):** an SFT-based method that applies step-level pruning to produce concise, skill-aware reasoning traces. **(3) SelfBudgeter (Li et al., 2025):** an RL-based method that iteratively optimizes a token budget under budget and formats reward signals. **(4) ThinkLess (Fang et al., 2025):** an RL-based method that detects critical thinking points and skips low-value steps. It introduces a reward function that balances accuracy with token usage, while maintaining performance. We follow the same hyperparameter settings as specified in the original papers for fair comparison of each efficient reasoning method.

*https://huggingface.co/microsoft/Phi-4-reasoning
*https://huggingface.co/mistralai/Magistral-Small-2507
*https://huggingface.co/openai/gpt-oss-20b

Results. According to Table 1, FROST offers the best overall balance between accuracy and token efficiency among SOTA efficient reasoning methods. Across the three base models—GPT-OSS-20B, Magistral-Small-1.1, and Phi-4-Reasoning—it yields an average 26.70% gain in accuracy while reducing token usage by 69.68%. Although TALE attains the best accuracy on a few individual tasks, it does so with substantially longer responses. This pattern is consistent with our hypothesis that both overly long and overly compressed reasoning can hurt performance. By concentrating generation on high-attention, decision-relevant reasoning traces, FROST improves efficiency and reduces the risk of hallucinated or misleading intermediate content. At the same time, because some useful steps may still receive low attention, FROST does not always achieve the top accuracy on every benchmark.

6.1 Supplementary Experiments

We provide additional analysis to investigate our method's performance at different training stages and under different attention functions.

Table 2: **Ablation on Different Activation Functions.** We compare different activations functions within the same training pipeline as FROST, using Phi-4-Reasoning on GSM8K, MATH500, AIME24, and Minerva. For each variant, we report Pass@1 and generated token count (#Tk), with variance discarded because it is consistently $\leq 2\%$. Best results are shown in **bold**, and second-best results are underlined. Across most settings, $\mathrm{Softmax}_1$ yields the strongest overall performance, while $\mathrm{Entmax15}$ is typically the next best alternative.

Method	GSM8K		MATH500		AIME24		Minerva		$\overline{\text{Pass@1}}$	$\overline{\#\text{Tk}}$
	Pass@1	#Tk	Pass@1	#Tk	Pass@1	#Tk	Pass@1	#Tk		
Base	0.9242	1017.70	0.5480	1721.95	0.0667	1017.70	0.2500	1898.86	0.4472	1414.05
$\mathrm{Softmax}$	0.8317	1160.63	0.4880	1379.52	0.1333	1909.07	0.2390	1934.72	0.4230	1595.99
$\mathrm{Sparsemax}$	0.8188	160.99	0.5120	451.59	0.1667	948.60	0.2647	580.84	0.4406	535.26
$\mathrm{Entmax15}$	0.8984	163.75	0.5520	406.97	0.1667	**876.63**	**0.2831**	439.48	0.4751	471.71
$\mathrm{Softmax}_1$ (FROST)	**0.9311**	**154.33**	**0.5980**	**344.37**	**0.2667**	899.80	0.2716	**401.19**	**0.5169**	**449.92**

FROST with Different Activation Functions. To better understand the role of $\mathrm{Softmax}_1$, we compare FROST with three alternative activation functions: standard $\mathrm{Softmax}$, $\mathrm{Sparsemax}$ (Hu et al., 2023; Martins & Astudillo, 2016), and $\mathrm{Entmax15}$ (Wu et al., 2024; Correia et al., 2019). Here, $\mathrm{Entmax15}$ refers to the Tsallis α-entmax family, which lies between softmax and sparsemax and produces a sharper attention distribution. The comparison in Table 2, conducted on Phi-4-Reasoning over GSM8K, MATH500, AIME24, and Minerva, shows that $\mathrm{Softmax}_1$ provides the strongest overall balance between accuracy and token efficiency. On average, it improves Pass@1 by **15.65%** and reduces token usage by **68.18%** relative to the base model. Compared with $\mathrm{Sparsemax}$ and $\mathrm{Entmax15}$, FROST more effectively suppresses low-impact reasoning traces without discarding too many important ones. The main exception is Minerva, where $\mathrm{Entmax15}$ achieves slightly higher accuracy at a similar token budget. This suggests that the effect of sharper attention can vary across tasks, and is consistent with prior discussions of aggressive reasoning compression (Yang et al., 2025; Wang, 2024).

Table 3: **Reasoning Outlier-removal Performance.** We evaluate the Phi-4-Reasoning model's performance for removing reasoning outliers on AIME2024 dataset. We report two outlier-related metrics: the maximum infinity norm $\|\mathbf{x}\|_\infty$ and the average kurtosis of the activation tensors, together with average sentence entropy, which reflects how strongly the model concentrates on critical reasoning traces. Variance is discarded because it is consistently $\leq 2\%$. Best results are shown in **bold**, and second-best results are underlined. Across most metrics, FROST achieves the strongest outlier-removal performance and the highest sentence entropy, indicating more focused and efficient reasoning.

Method	Maximum Infinity Norm $\|\mathbf{x}\|_\infty \downarrow$	Average Kurtosis $\downarrow$	Average Sentence Entropy $\uparrow$	Pass@1 $\uparrow$	#Tk $\downarrow$
Base	35.31	241.72	2.71	0.0667	1017.70
$\mathrm{Softmax}$	34.53	189.36	2.79	0.1333	1909.07
$\mathrm{Sparsemax}$	34.06	152.18	2.93	0.1667	948.60
$\mathrm{Entmax15}$	30.39	43.72	2.92	0.1667	**876.63**
FROST	**29.67**	**21.54**	**3.07**	**0.2667**	899.80

Reasoning Outlier-removal Performance in FROST. We study whether FROST actually removes the kinds of attention patterns associated with inefficient reasoning. To do this, we use two activation-level measures: the *maximum infinity norm* $\|\mathbf{x}\|_\infty$ and the *average kurtosis* of activation tensors $\mathbf{x}$ across transformer layers. These two metrics together quantify the presence of outliers. Lower values on these metrics indicate weaker outlier behavior. We also report average sentence entropy, following Wang et al. (2025), as a signal of how strongly the model concentrates on critical reasoning content. Critical tokens tend to exhibit higher entropy than non-critical ones. As shown in Table 3, FROST improves all three indicators on AIME2024 with Phi-4-Reasoning. Compared with the base model, it reduces the maximum infinity norm by **15.97%**, reduces average kurtosis by **91.09%**, and increases average sentence entropy by **13.28%**. These results suggest that FROST both suppresses outlier-heavy activations and shifts attention toward more useful reasoning traces. Additionally, the table shows a clear pattern: larger outlier values tend to coincide with lower sentence entropy and weaker reasoning efficiency. The main exception is the comparison between $\mathrm{Sparsemax}$ and $\mathrm{Entmax15}$, where entropy remains similar even though the outlier statistics differ. This suggests that internal activation behavior and final output quality are related, but not perfectly coupled. One possible explanation is that both $\mathrm{Entmax15}$ and $\mathrm{Sparsemax}$ sharpen attention by suppressing not only weak scores but also some larger ones. While this reduces outlier-related statistics, it can also remove parts of genuinely useful reasoning traces, which lowers sentence entropy and can hurt Pass@1 performance. In other words, these activations may improve internal sharpness without consistently improving output quality. Because $\mathrm{Softmax}_1$ contracts the low-attention tail more selectively, it avoids this trade-off more effectively and better preserves important reasoning signals.

6.2 Generalizability of FROST

We evaluate whether efficiency gains transfer beyond the training domain and how FROST generalizes to out-of-domain reasoning tasks. Using Phi-4-Reasoning as the base model, we evaluate on LeetCode (Xia et al., 2025b), LiveCodeBench (Jain et al., 2025), and UGPhysical (Xu et al., 2025b), spanning coding and physics reasoning tasks. As shown in Table 4, FROST preserves and often improves out-of-domain performance while substantially reducing token usage. This indicates that the method does not overfit the training distribution and successfully filters out irrelevant reasoning traces. It also validates that FROST generalizes beyond any specific tasks used during fine-tuning. Because FROST only replaces the attention activation and updates a small set of LoRA parameters, overall reasoning capabilities remain stable.

Table 4: **Ablation on Other Reasoning Tasks.** Using Phi-4-Reasoning as the base model, we test FROST on three out-of-domain tasks: LeetCode and LiveCodeBench for code reasoning, and UGPhysics for physics reasoning. We report Pass@1 and generated token count (#Tk), and variance consistently $\leq 2\%$. **Bold** indicates the best result and underline indicates the second-best result. FROST achieves the best overall performance, showing that it generalizes well beyond math reasoning task.

Method	Leetcode		LiveCodeBench		UGPhysics		$\overline{\text{Pass@1}}$	$\overline{\#\text{Tk}}$
	Pass@1	#Tk	Pass@1	#Tk	Pass@1	#Tk		
Base	0.3222	2755.13	0.3248	3154.80	0.3172	2603.00	0.3214	2837.64
Softmax	0.3778	2106.85	0.3538	2909.07	0.3011	2622.52	0.3442	2546.15
FROST	**0.3889**	**1163.06**	**0.3777**	**1967.56**	**0.3473**	**805.77**	**0.3713**	**1312.13**

7 Discussion and Conclusion

We introduce **FROST**, an attention-aware approach for making large reasoning models more efficient. Our approach detects and removes *reasoning outliers*, which generate low-value reasoning traces in LRMs, resulting in longer, slower responses without improving answer quality. FROST combines $\mathrm{Softmax}_1$-based outlier suppression with lightweight supervised fine-tuning, allowing the model to produce shorter and more accurate reasoning paths for math problems. Our analysis also shows that $\mathrm{Softmax}_1$ can suppress these outliers at the sentence level. In experiments, FROST improves the overall accuracy–efficiency tradeoff, with **+26.70%** average accuracy and **-69.68%** token usage. It also reduces the maximum infinity norm $\|\mathbf{x}\|_\infty$ by **15.97%** and average kurtosis by **91.09%**, further supporting its effectiveness at mitigating reasoning outliers.

Despite FROST showing strong performance in efficient reasoning, several limitations remain. First, our study primarily focuses on mathematical reasoning, so the method still needs broader evaluation across other reasoning domains, such as coding and science. Second, the current version of FROST relies on supervised fine-tuning and does not leverage GRPO-style optimization, which could yield further efficiency gains. In future work, we plan to test FROST on a wider range of reasoning tasks and explore GRPO-based extensions that build on the current outlier-removal framework.

ACKNOWLEDGMENTS

HL would like to thank Dennis Wu for insightful discussions on theoretical analysis. This research was supported in part by the computational resources and staff contributions of the RTRC High Performance Computing Cluster. Additional support was provided by the Quest High Performance Computing Facility at Northwestern University, which is jointly supported by the Office of the Provost, the Office for Research, and Northwestern University Information Technology. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

ETHICAL STATEMENT

This work investigates reasoning outliers in large reasoning models (LRMs) and proposes an outlier-removal technique to mitigate attention outliers. In line with the ICLR Code of Ethics[*] , we acknowledge that our method may inadvertently amplify biases present in training data, potentially leading to unfair outcomes for underrepresented groups. Prior studies have also noted that supervised fine-tuning (SFT) can induce shallow alignment and affect red-teaming protection, but this issue is outside the scope of our work. Our focus is on improving reasoning efficiency, and we believe this research does not raise serious ethical concerns.

REPRODUCIBILITY

All experiments are conducted with three random seeds, yielding stable results with standard deviations below 2%. We adopt a unified training setup using the AdamW optimizer with learning rate 1×10^{-5}, batch size 8, and batch size 256 for deployment. For low-rank adaptation, we set the LoRA rank to 8 and LoRA α to 16. Detailed hyperparameters are provided in Section E.2, and theoretical proofs are included in Section C.

REFERENCES

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

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

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, et al. R-kv: Redundancy-aware kv cache compression for reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*, 2025.

[*] https://iclr.cc/public/CodeOfEthics

Yew Ken Chia, Guizhen Chen, Weiwen Xu, Anh Tuan Luu, Soujanya Poria, and Lidong Bing. Reasoning paths optimization: Learning to reason and explore from diverse paths. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

Daewon Choi, Jimin Lee, Jihoon Tack, Woomin Song, Saket Dingliwal, Sai Muralidhar Jayanthi, Bhavana Ganesh, Jinwoo Shin, Aram Galstyan, and Sravan Babu Bodapati. Think clearly: Improving reasoning via redundant token pruning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 21437–21451, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7.

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

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2174–2184, Hong Kong, China, November 2019. Association for Computational Linguistics.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *The Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Hao Ding, Ziwei Fan, Ingo Guehring, Gaurav Gupta, Wooseok Ha, Jun Huan, Linbo Liu, Behrooz Omidvar-Tehrani, Shiqi Wang, and Hao Zhou. Reasoning and planning with large language models in code development. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6480–6490, 2024a.

Yangruibo Ding, Jinjun Peng, Marcus Min, Gail Kaiser, Junfeng Yang, and Baishakhi Ray. Semcoder: Training code language models with comprehensive semantics reasoning. *Advances in Neural Information Processing Systems*, 37:60275–60308, 2024b.

Ethan Dyer and Guy Gur-Ari. Minerva: Solving quantitative reasoning problems with language models. *June*, 30:2022, 2022.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. In *The Thirty-ninth Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Tianyu Fu, Yi Ge, Yichen You, Enshu Liu, Zhihang Yuan, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. R2r: Efficiently navigating divergent reasoning paths with small-large model token routing. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware LLM reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*,

pp. 24842–24855, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyan Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, Zhen Wang, and Zhiting Hu. LLM reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models. In *First Conference on Language Modeling*, 2024.

Haoyu He, Haozheng Luo, and Qi R Wang. St-moe-bert: A spatial-temporal mixture-of-experts framework for long-term cross-city mobility prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*, pp. 10–15, 2024.

Haoyu He, Haozheng Luo, Yan Chen, and Qi Wang. RHYTHM: Reasoning with hierarchical temporal tokenization for human mobility. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of LLMs via reinforcement learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations (ICLR)*, 2021.

Jerry Yao-Chieh Hu, Donglin Yang, Dennis Wu, Chenwei Xu, Bo-Yu Chen, and Han Liu. On sparse modern hopfield model. In *The Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Jerry Yao-Chieh Hu, Pei-Hsuan Chang, Robin Luo, Hong-Yu Chen, Weijian Li, Wei-Po Wang, and Han Liu. Outlier-efficient hopfield layers for large transformer-based models. In *The Forty-first International Conference on Machine Learning (ICML)*, 2024.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

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

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

Eric Hanchen Jiang, Haozheng Luo, Shengyuan Pang, Xiaomin Li, Zhenting Qi, Hengli Li, Cheng-Fu Yang, Zongyu Lin, Xinfeng Li, Hao Xu, et al. Learning to rank chain-of-thought: An energy-based approach with outcome supervision. *arXiv preprint arXiv:2505.14999*, 2025a.

Yuxuan Jiang, Dawei Li, and Frank Ferraro. Drp: Distilled reasoning pruning with skill-aware step decomposition for efficient large reasoning models. *arXiv preprint arXiv:2505.13975*, 2025b.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, PeiFeng Wang, silvio savarese, Caiming Xiong, and Shafiq Joty. A survey of frontiers in LLM reasoning: Inference scaling, learning to reason, and agentic systems. *Transactions on Machine Learning Research*. ISSN 2835-8856. Survey Certification.

Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. Overthinking: Slowdown attacks on reasoning llms. *arXiv preprint arXiv:2502.02542*, 2025.

Zheng Li, Qingxiu Dong, Jingyuan Ma, Di Zhang, Kai Jia, and Zhifang Sui. Selfbudgeter: Adaptive token allocation for efficient LLM reasoning, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

Yule Liu, Jingyi Zheng, Zhen Sun, Zifan Peng, Wenhan Dong, Zeyang Sha, Shiwen Cui, Weiqiang Wang, and Xinlei He. Thought manipulation: External thought can be efficient for large reasoning models. *arXiv preprint arXiv:2504.13626*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Haozheng Luo, Chenghao Qiu, Maojiang Su, Zhihan Zhou, Zoe Mehta, Guo Ye, Jerry Yao-Chieh Hu, and Han Liu. Fast and low-cost genomic foundation models via outlier removal. In *Forty-second International Conference on Machine Learning*, 2025b.

Haozheng Luo, Jiahao Yu, Wenxin Zhang, Jialong Li, Jerry Yao-Chieh Hu, Yan Chen, Binghui Wang, Xinyu Xing, and Han Liu. Decoupled alignment for robust plug-and-play adaptation. In *The Impact of Memorization on Trustworthy Foundation Models: ICML 2025 Workshop*, 2025c.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. In *Submitted to ACL Rolling Review - February 2025*, 2025. under review.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pp. 1614–1623. PMLR, 2016.

Mathematical Association of America. American invitational mathematics examination 2024, 2024. Official competition problems.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *The Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Zhenyu Pan, Haozheng Luo, Manling Li, and Han Liu. Conv-coa: Improving open-domain question answering in large language models via conversational chain-of-action. *arXiv preprint arXiv:2405.17822*, 2024.

Zhenyu Pan, Haozheng Luo, Manling Li, and Han Liu. Chain-of-action: Faithful and multimodal question answering through large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *The Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. Group robust preference optimization in reward-free rlhf. *Advances in Neural Information Processing Systems*, 37:37100–37137, 2024.

Abhinav Rastogi, Albert Q Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, et al. Magistral. *arXiv preprint arXiv:2506.10910*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Hongze Tan and Jianfei Pan. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Libo Wang. Reducing reasoning costs - the path of optimization for chain of thought via sparse attention mechanism. In *Muslims in ML Workshop co-located with NeurIPS 2024*, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *The Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Dennis Wu, Jerry Yao-Chieh Hu, Weijian Li, Bo-Yu Chen, and Han Liu. STanhop: Sparse tandem hopfield model for memory-enhanced time series prediction. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. TokenSkip: Controllable chain-of-thought compression in LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3351–3363, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-332-6.

Yunhui Xia, Wei Shen, Yan Wang, Jason Klein Liu, Huifeng Sun, Siyue Wu, Jian Hu, and Xiaolong Xu. Leetcodedataset: A temporal dataset for robust evaluation and efficient training of code llms. *arXiv preprint arXiv:2504.14655*, 2025b.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Xin Xu, Qiyun Xu, Tong Xiao, Tianhao Chen, Yuchen Yan, Jiaxin ZHANG, Shizhe Diao, Can Yang, and Yang Wang. UGPhysics: A comprehensive benchmark for undergraduate physics reasoning with large language models. In *Forty-second International Conference on Machine Learning*, 2025b.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Lijie Yang, Zhihao Zhang, Arti Jain, Shijie Cao, Baihong Yuan, Yiwei Chen, Zhihao Jia, and Ravi Netravali. Less is more: Training-free sparse attention with global locality for efficient reasoning. *arXiv preprint arXiv:2508.07101*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Jingyang Yi, Justin Wang, and Sida Li. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Haoran Zhao, Yuchen Yan, Yongliang Shen, Haolei Xu, Wenqi Zhang, Kaitao Song, Jian Shao, Weiming Lu, Jun Xiao, and Yueting Zhuang. Let LRMs break free from overthinking via self-braking tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Shieh. Understanding and enhancing safety mechanisms of LLMs via safety-specific neuron. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Zhihan Zhou, Robert Riley, Satria Kautsar, Weimin Wu, Rob Egan, Steven Hofmeyr, Shira Goldhaber-Gordon, Mutian Yu, Harrison Ho, Fengchen Liu, et al. Genomeocean: An efficient genome foundation model trained on large-scale metagenomic assemblies. *bioRxiv*, pp. 2025–01, 2025.

Supplementary Material

A Impact Statement **16**

B Additional Related Work **16**

C Proofs of Main Text **17**
C.1 Lemma 5.1 . . . 17
C.2 Theorem 5.1 . . . 17
C.3 Theorem 5.2 . . . 17

D An Example of LRM Reasoning Traces **17**

E Experiment System and Implement Settings **17**
E.1 Computational Resources . . . 17
E.2 Hyperparameters . . . 17

F Additional Experiments **18**
F.1 Training and Test Time Comparison . . . 19
F.2 Attention Distributions of Activation Functions . . . 19

G Influence the attention dynamics of $\mathrm{Softmax}_1$ during training and inference **19**

H Influence of $\mathrm{Softmax}_1$ Across Layers **20**

I Extended Attention Heatmaps Across Additional Layers and Heads **20**

J Human Expert Evaluation **20**

K Disclosure of LLM Usage **20**

A Impact Statement

We believe this methodology offers an opportunity to strengthen the core of large reasoning models by improving efficiency and enabling models to produce more critical reasoning traces. However, it may also amplify biases present in the training data, potentially leading to unfair or discriminatory outcomes for underrepresented groups.

B Additional Related Work

Efficient Alignment. In recent years, foundation models (Zhou et al., 2025; He et al., 2025; 2024; Wang et al., 2024; Touvron et al., 2023) have shown strong capabilities in solving multitask problems. To further improve their performance on specific tasks, alignment techniques are essential for refining model behavior. However, traditional approaches like RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2023) are computationally expensive. This highlights the urgent need for parameter-efficient fine-tuning methods that offer effective and economical alignment for foundation models. Several traditional methods demonstrate strong capabilities in aligning foundation models, including LoRA (Hu et al., 2021) and QLoRA (Dettmers et al., 2023). Building on this, Luo et al. (2025b) propose a LoRA variant that replaces the standard softmax layer with OutEffHop layers (Hu et al., 2024) to improve the efficiency of low-rank adaptation. However, all of these methods are heavily based on LoRA, and when adaptation is required for modules outside the attention architecture, the computational cost increases significantly. Zhao et al. (2025b); Luo et al. (2025c) propose novel alignment methods that focus on small subsets of neurons within foundation models. For example, Zhao et al. (2025b) identify key neurons with high influence on LLMs' jailbreak defense using latent representations, and fine-tune only these neurons using red-teaming datasets. Our method builds on fast low-rank adaptation techniques (Luo et al., 2025b), further improving adaptation efficiency, and integrates them into SFT training to optimize reasoning paths and produce efficient reasoning models.

C PROOFS OF MAIN TEXT

C.1 LEMMA 5.1

Proof of Lemma 5.1. Monotonicity means that if we increase any input coordinate to ϕ, its output does not decrease. Let $u = \{z_t\}_{t \in S_i}$ and $w = \{z_{t'}\}_{t' \in S_j}$. If for each coordinate of u there is a not-smaller coordinate in w replaced by the smaller value, then by repeatedly applying coordinatewise monotonicity we obtain $s_i = \phi(u) \;\geq\; \phi(w) = s_j$. Order preservation (P1) then yields $\alpha_i \geq \alpha_j$.

C.2 THEOREM 5.1

Proof of Theorem 5.1. By Lemma 5.1, sentence scores s reflect dominance induced by token compatibilities under ϕ. Applying Assumption 5.1(P3) directly to s yields (2). Assumption 5.1(P2) allows re-centering $s \leftarrow s - c\mathbf{1}$ without changing α; thus (2) is invariant to any global shift and depends only on relative separations.

C.3 THEOREM 5.2

Proof of Theorem 5.2. For (3), apply operator-norm submultiplicativity: $\|W_o(\alpha_i v_i)\| \leq \|W_o\|_{\mathrm{op}} \cdot \alpha_i \|v_i\| \leq B_o\, \varepsilon\, B_v$. To obtain (4), propagate the perturbation through L differentiable layers with Jacobians J_ℓ:

$$\|\Delta \ell_i^{(L)}\| \;\leq\; \Big(\prod_{\ell=1}^{L} \|J_\ell\|_{\mathrm{op}}\Big)\, \|W_o\|_{\mathrm{op}}\, \alpha_i \|v_i\| \;\leq\; \varepsilon \Big(\prod_{\ell=1}^{L} B_\ell\Big) B_o B_v.$$

Finally, since $\mathrm{Softmax}_1$ is 1-Lipschitz in $\ell_\infty \to \ell_1$, the change in probabilities is bounded by the logit change, yielding (5). Replacing $\prod_{\ell=1}^{L} B_\ell$ with B^L (by definition of B) gives the stated $O(B_o B_v B^L \varepsilon)$ rate. If B_o, B_v, B are $O(1)$, the rate simplifies to $O(\varepsilon)$.

D AN EXAMPLE OF LRM REASONING TRACES

In this section, we analyze the Phi-4-Reasoning response to the first question of AIME24, which is also illustrated in Figure 4. As shown in the color box in Section D, traces S1 and S2 are classified as uncritical. Although S2 includes partially critical content such as `"So the walking time (actual walking time) plus t minutes equals total time."`, its overall reasoning remains non-critical. Trace S3 represents a critical reasoning step, where the model identifies the two key equations in the problem. Subsequently, from S4 to S19, the model enters a self-verification phase, producing reasoning traces beginning with `wait` that reflect self-checking and correction. Starting from S20, the model resumes critical reasoning after the signal `"We'll produce final answer in a box."`, and by S24, it generates the final answer, concluding its reasoning process.

E EXPERIMENT SYSTEM AND IMPLEMENT SETTINGS

E.1 COMPUTATIONAL RESOURCES

We perform all experiments using two NVIDIA H100 GPUs with 80GB of memory and a 12-core INTEL(R) XEON(R) PLATINUM 8592 CPU operating at 1.90GHz. Our code is developed in PyTorch and utilizes the Hugging Face Transformer Library for experimental execution. For running the LLMs, we use the default system prompt provided by the official source and set the temperature to 0.6 to balance consistency and performance.

E.2 HYPERPARAMETERS

We present the hyperparameters used in the fine-tuning stage for each model. We use **AdamW** (Loshchilov & Hutter, 2019) as the optimizer. Most other hyperparameters are kept consistent across all models and datasets, including a batch size of 256 during deployment and 8 during training. In training, we also use gradient accumulation with 4 steps and set the weight decay to 0.01 for

An Example of LRM Reasoning Traces

[Question] Every morning Aya walks 9 km and then spends t minutes in a coffee shop. At speed s km/h, the total time is 4 hours (including the coffee stop). At speed $s + 2$ km/h, the total time is 2 hours 24 minutes. If she walks at $s + \frac{1}{2}$ km/h, find the total time in minutes (including the coffee stop).
[Uncritical Trace S1] <think>We are given: "Every morning Aya goes for a 9-kilometer-long walk and stops at a coffee shop afterwards. When she walks at a constant speed of s kilometers per hour, the walk takes her 4 hours, including t minutes spent in the coffee shop. When she walks $s + 2$ kilometers per hour, the walk takes her 2 hours and 24 minutes, including t minutes spent in the coffee shop. Suppose Aya walks at $s + \frac{1}{2}$ kilometers per hour. Find the number of minutes the walk takes her, including the t minutes spent in the coffee shop."
[Uncritical Trace S2] Let's denote: "Walking distance = 9 km. There's a coffee shop stop time t minutes, but note that the given total times includes the coffee shop stop time. So the walking time (actual walking time) plus t minutes equals total time. Let s be her walking speed (km/h) in first scenario. Then walking time (in hours) is $9/s$. And the total time is 4 hours."
[Critical Trace S3] But includes t minutes stop, so actual walking time is 4 hours minus t minutes (converted to hours, $t/60$ hours), so equation: $9/s = 4 - t/60$. (Equation 1) and $9/(s + 2) = 2.4 - t/60$ (Equation 2)
[Uncritical Trace S4] But wait, let's check: speed = $s + 2$, then walking time (in hours) is $9/(s + 2)$. Total time is 2 hours 24 minutes, which is $2 + 24/60 = 2.4$ hours.
[Uncritical Trace S5] But wait, let's check: The problem said: "Suppose Aya walks at s+1/2 kilometers per hour. Find the number of minutes the walk takes her, including the t minutes spent in the coffee shop."
[Uncritical Trace S6] Wait, 2 hours 24 minutes = 144 minutes.
[Uncritical Trace S7] Let me recalc: 3 hours 24 minutes = 2*60 + 24 = 120 + 24 = 144 minutes.
[Uncritical Trace S8] Let us double-check the reasoning process.
[Uncritical Trace S9-S17] .....
[Uncritical Trace S18] I'll now produce step-by-step explanation.
[Uncritical Trace S19] We'll produce final answer in a box.
[Critical Trace S20] The two equations are given as $9/s = 4 - t/60$ (Equation 1) and $9/(s + 2) = 2.4 - t/60$ (Equation 2).
[Critical Trace S21] Alternatively, we can solve for t in terms of s from equation 1: $t = 60*(4-9/s) = 240-540/s$, and plug in equation 2: $9/(s+2) = 2.4-(240-540/s)/60 = 2.4 - (4 - 9/s) = 2.4 - 4 + 9/s = (9/s) - 1.6$.
[Critical Trace S22] So equation becomes: $9/(s + 2) = (9/s) - 1.6$. Multiply both sides by $s(s + 2)$: $9s = (9(s + 2)) - 1.6s(s + 2)$.
[Critical Trace S23] Now, when Aya walks at speed $s + 1/2 = 2.5 + 0.5 = 3$ km/h, then walking time (in hours) is $9/3 = 3$ hours. But total time includes coffee shop stop time t minutes, which is 24 minutes = $24/60 = 0.4$ hours. So total time is $3 + 0.4 = 3.4$ hours, which is 3 hours 24 minutes.
[Critical Trace S24] I'll produce final answer: $\boxed{204}$. </think>

all training runs. A learning rate of $1e^{-5}$ is used for all models during fine-tuning. For low-rank adaptation, we use a LoRA rank of 8 and LoRA alpha set to 16. In FROST, we set the maximum training steps to 5,000. All supervised fine-tuning and GRPO training are conducted using mixed precision with bfloat16. In deployment, we set the temperature to 0.6 for all models with top-p sampling at 0.9. For evaluation, we use a maximum generation length of 4096 across all models, except TALE.

F Additional Experiments

In this section, we present additional experiments demonstrating that FROST surpasses current state-of-the-art efficient reasoning methods.

F.1 Training and Test Time Comparison

We conduct experiments to measure the training and inference time of each baseline and compare their computational costs with FROST. For evaluation, test time is measured on the AIME tasks with the GPT-OSS-20B model, while training time is reported using the respective datasets specified in each baseline's original paper. All experiments are conducted on the same computational resources, as described in Section E.1.

Table 5: **Comparison of Training and Test Time Costs Across Methods.** We conduct experiments to measure the training and test time of each method. For test-time evaluation, we use the AIME dataset with the GPT-OSS-20B model. Best results are shown in **bold**, and second-best are underlined.

Method	TALE	DRP	ThinkLess	FROST
Training Time (m)	-	353	1186	**204**
Test Time (m)	56	18.5	4.2	**3**

As shown in Table 5, FROST achieves the fastest training time among all methods, while also minimizing computation cost and inference time during deployment. This demonstrates that our approach not only accelerates training but also reduces deployment overhead.

F.2 Attention Distributions of Activation Functions

We conduct an additional experiment to analyze the attention distribution of GPT-OSS-20B on a sample from the GSM8K dataset. As shown in Figure 6, FROST effectively removes a large number of low-attention sentences while retaining significant ones. In contrast, the vanilla model produces many sentences with low attention weights, and $\mathrm{Sparsemax}$ and $\mathrm{Entmax15}$ retain only one to two sentences, often aggressively discarding important reasoning traces. This visualization provides an explanation consistent with the performance results reported in Table 2.

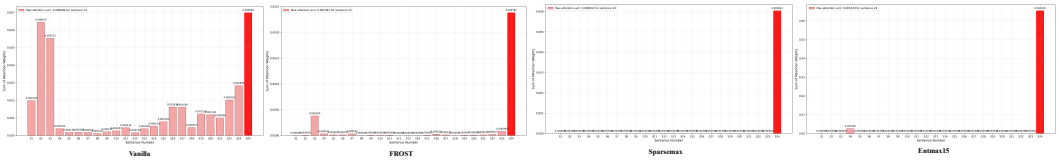

Figure 6: **Attention Distribution of Each Activation Function.**

G Influence the Attention Dynamics of $\mathrm{Softmax}_1$ during Training and Inference

We observe that incorporating $\mathrm{Softmax}_1$ significantly influences both training and inference attention dynamics across transformer layers. During supervised fine-tuning (SFT), $\mathrm{Softmax}_1$ enforces tail contraction by suppressing low-attention activations, which stabilizes gradients and reduces the variance of updates propagated through residual connections. This effect leads to faster convergence of LoRA adapters, as the low-rank parameter subspace more efficiently aligns with critical attention directions, improving overall adaptation coverage within fewer training steps. This observation is consistent with Luo et al. (2025b); Hu et al. (2024). Across layers, $\mathrm{Softmax}_1$ reshapes the attention landscape—shallow layers become more selective in contextual grounding, while deeper layers exhibit higher entropy concentration around critical reasoning traces. During inference, this sharpening propagates forward, effectively filtering redundant reasoning sentences while maintaining coherence. Together, these behaviors demonstrate that $\mathrm{Softmax}_1$ not only enhances efficient reasoning but also accelerates LoRA-SFT optimization by improving the representational focus of each attention head.

H INFLUENCE OF $\mathrm{Softmax}_1$ ACROSS LAYERS

We analyze the effect of $\mathrm{Softmax}_1$ across transformer layers by visualizing the attention distributions of head 15 for both vanilla Softmax and $\mathrm{Softmax}_1$. As shown in Figures 7 and 8, $\mathrm{Softmax}_1$ consistently suppresses attention outliers, leading to smoother and more stable activations across the network. In lower layers, $\mathrm{Softmax}_1$ contracts heavy tails and mitigates rare extreme peaks, enhancing local feature mixing with higher-entropy and reduced kurtosis distributions. In higher layers, it suppresses residual long-range spikes and sharpens focus on semantically relevant tokens, yielding sparser yet more stable attention and clearer causal information flow.

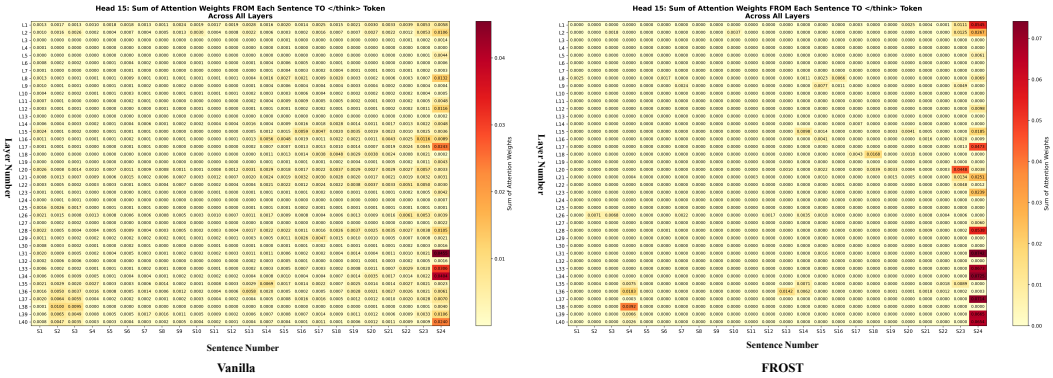

Figure 7: **Theoretical Analysis of Reasoning Outlier Removal in All Layers**

I EXTENDED ATTENTION HEATMAPS ACROSS ADDITIONAL LAYERS AND HEADS

In this section, we present extended attention heatmaps covering additional layers and heads. Specifically, we analyze **Layers 0, 5, 15, 25, 30, 35, and 39** and **Heads 0, 5, 10, 15, 20, 25, 30, 35, and 39** to provide a more comprehensive view of attention evolution across the network. The corresponding observations are illustrated in Figure 9.

J HUMAN EXPERT EVALUATION

We invite three computer science students specializing in reasoning models to annotate reasoning traces generated by the original and FROST-trained models. We then compare the traces pruned by FROST and evaluate their criticality based on relevance and contribution to the final answer. Averaging across all evaluators, FROST achieves **92%** accuracy in correctly removing non-critical reasoning traces. Only **8%** of reasoning traces are incorrectly removed, which significantly degrades final-answer accuracy. These mistakenly pruned traces are typically long and contain repeated information that supports self-verification and error correction. However, they also provide critical content—such as key equations—in the end of trace. This observation suggests a potential explanation for why FROST achieves the second-best Pass@1 score in the Phi-4-Reasoning experiment shown in Table 1.

K DISCLOSURE OF LLM USAGE

In our paper and project, we use large language models (LLMs) to help revise the text for greater conciseness and precision.

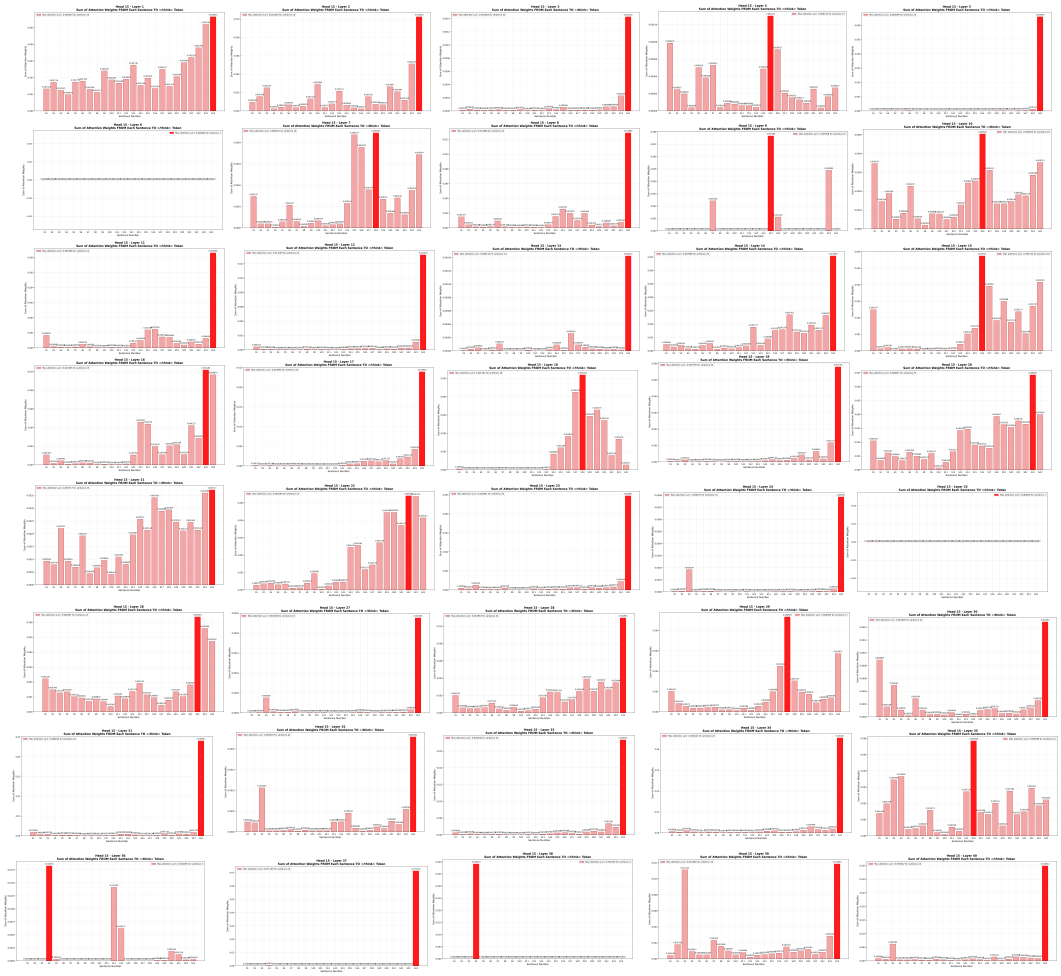

Figure 8: **Attention Distribution of** $\mathrm{Softmax}_1$ **Across All Layers**

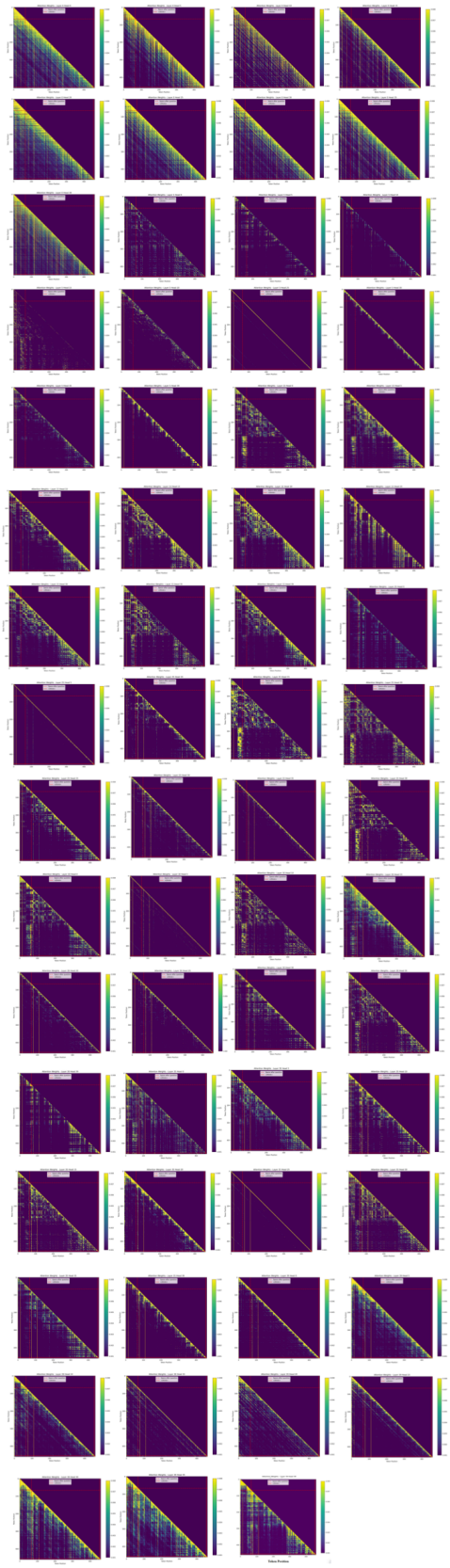

Figure 9: **Extended Attention Heatmaps Across Additional Layers and Heads**