

NEAT: Neuron-Based Early Exit for Large Reasoning Models

Kang Liu¹, Yongkang Liu², Xiaocui Yang¹, Peidong Wang¹, Wen Zhang¹,
Shi Feng^{1*}, Yifei Zhang¹, Daling Wang^{1*}

¹School of Computer Science and Engineering, Northeastern University,
Shenyang 110819, China

²School of Computer and Communication Engineering, Northeastern University,
Qinhuangdao 066004, China

lk_stu_neu@163.com, {fengshi, wangdaling}@cse.neu.edu.cn

Abstract

Large Reasoning Models (LRMs) often suffer from *overthinking*, a phenomenon in which redundant reasoning steps are generated after a correct solution has already been reached. Existing early reasoning exit methods primarily rely on output-level heuristics or trained probing models to skip redundant reasoning steps, thereby mitigating overthinking. However, these approaches typically require additional rollout computation or externally labeled datasets. In this paper, we propose **NEAT**, a **N**euron-based **E**arly **r**easoning **e**xit framework that monitors neuron-level activation dynamics to enable training-free early exits, without introducing additional test-time computation. NEAT identifies exit-associated neurons and tracks their activation patterns during reasoning to dynamically trigger early exit or suppress reflection, thereby reducing unnecessary reasoning while preserving solution quality. Experiments on four reasoning benchmarks across six models with different scales and architectures show that, for each model, NEAT achieves an average token reduction of 22% to 28% when averaged over the four benchmarks, while maintaining accuracy.¹

1 Introduction

Large Reasoning Models (LRMs), such as OpenAI o1 (OpenAI, 2024) and DeepSeek-R1 (Guo et al., 2025), have achieved substantial performance gains on complex reasoning tasks, including mathematical problem solving and competitive programming—by scaling sequential test-time computation (Snell et al., 2024). However, LRMs often perform more reasoning steps than necessary, continuing generation even after correct answers have been reached. This behavior, referred to as *overthinking* (Sui et al., 2025), leads to excessive computational consumption. Beyond increased inference

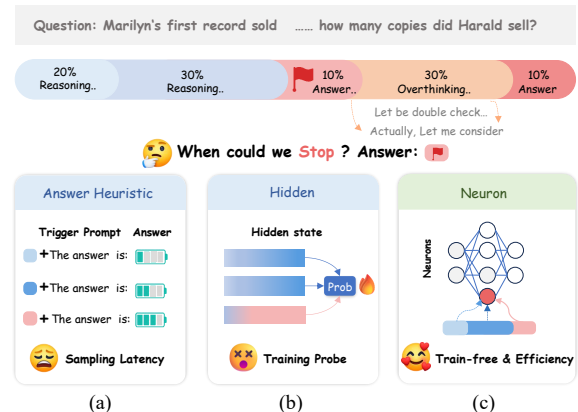


Figure 1: Comparison of different early reasoning exit methods. (a) Output-based Answer Heuristic methods (b) Probe-based method. (c) Our method NEAT.

latency, overthinking also raises the risk of hallucination and error accumulation in the generation tail (Wu et al., 2025; Yu and Ananiadou, 2024a). Several prior works attempt to mitigate this issue by training models to reason more efficiently through supervised fine-tuning (SFT) (Zhao et al., 2025a) or reinforcement learning (RL) (Zhang et al., 2025b; Lou et al., 2025). In contrast, a more direct and cost-effective alternative is to identify the moment when reasoning has effectively completed and exit the reasoning process early, thereby skipping redundant steps.

A widely explored line of work estimates reasoning completion by evaluating the convergence of intermediate outputs, such as answer consistency (Fu et al., 2024; Huang et al., 2025) or confidence-based criteria (Yang et al., 2025; Liu and Wang, 2025). While effective, these approaches typically require repeated rollouts, resulting in increased parallel test-time computation and inference latency, as illustrated in Figure 1(a). Another line of work explores information encoded in latent hidden states during reasoning to provide early exit signals and avoid additional sampling latency. Although

*Corresponding authors

¹https://github.com/KpKqwq/Early_Exist_Circuit

promising, existing methods (Zhang et al., 2025a; Liu and Wang, 2025; Eisenstadt et al., 2025) generally rely on externally supervised probes trained on hidden states, as shown in Figure 1(b).

In this paper, we further explore the potential of internal latent states from a more fine-grained, neuron-level perspective to provide efficient early-exit signals, and propose **NEAT** (Neuron-based Early reASONing exiT), a training-free framework that determines whether to exit or continue reasoning by leveraging neuron-level activation dynamics, without requiring external supervision or parallel inference-time computation. Specifically, our method first identifies a sparse set of *exit-associated neurons* whose activation patterns are tightly coupled with natural reasoning termination. These neurons are selected based on their causal contribution to termination-related token predictions and exhibit activation values whose temporal mass is concentrated toward the end of the reasoning process. During inference, NEAT monitors the real-time activation dynamics of these neurons to assess whether the model has internally converged. To mitigate the risk of premature termination, we incorporate a dynamic intervention strategy that leverages the strength of observed neuron activation patterns as indicators of reasoning completion. When the signal is strong, the model performs early exit, whereas when the signal is relatively weak, it suppresses reflection tokens and allows reasoning to continue. Experiments across multiple reasoning benchmarks and model architectures show that NEAT reduces redundant token generation while maintaining final-answer accuracy.

Our contributions are summarized as follows:

- We demonstrate that neuron-level activation dynamics provide reliable internal signals for early exit, offering a fine-grained alternative to output-level answer-convergence methods.
- We propose **NEAT**, a training-free early reasoning-exit framework that monitors neuron-level activation dynamics and performs two inference-time interventions, early exit and reflection suppression, to control the progression of reasoning, without requiring external supervision or parallel rollouts.
- Experiments across various model architectures and reasoning benchmarks demonstrate that NEAT reduces the average token count by 22.0% to 28.2% while maintaining accuracy.

2 Related Work

Early Reasoning Exit. Early reasoning exit methods aim to precisely identify the point at which a reasoning model has gathered sufficient information to produce a correct answer, allowing the model to skip redundant steps and thus mitigate overthinking during inference. Most existing studies primarily rely on output-based heuristic methods. For example, Dynasor (Fu et al., 2024) periodically probes intermediate answers at fixed token intervals and triggers an early exit when multiple consecutive outputs are consistent. However, probing answer convergence at fixed steps incurs additional overhead and lacks flexibility. Alternatively, DEER (Yang et al., 2025) alleviates this issue by triggering exits only at reflection keywords (e.g., Wait), guided by answer confidence. However, direct exiting may hurt performance. Thus, CGRS (Huang et al., 2025) proposes to only suppress reflection triggers when the model is confident in its current response. Another line of work leverages internal hidden states to provide early-exit signals without additional sampling overhead. For example, TPV (Eisenstadt et al., 2025) trains probes to encode internal reasoning progress and manipulates these probe signals during inference to skip redundant reasoning steps. Similarly, Zhang et al. (2025a) trains probes on hidden states to predict answer correctness. While such methods avoid additional computation during reasoning, they require extra supervised data to train probes and still exhibit a notable performance gap compared to mature output-based approaches (Yang et al., 2025).

Neurons in LLMs. Neurons are the most fundamental computational units in LLMs. Prior work has identified individual neurons that are closely associated with diverse model capabilities and behaviors (Tang et al., 2024; Shi et al., 2024; Chen et al., 2024). For reasoning tasks, studies such as Yu and Ananiadou (2024a); Rai and Yao (2024) show that only a small subset of neurons is selectively activated, and that these neurons play an important role in enabling arithmetic reasoning. Moreover, Yu et al. (2025) finds that failures in multi-step reasoning can often be attributed to neurons being activated at incorrect positions during the reasoning process, rather than to a lack of relevant knowledge. Beyond deepening the analysis and understanding of model behavior, identifying such neurons has also enabled several practical applications (Gurgurov et al., 2025; Yu and Ananiadou, 2025). For

example, Tang et al. (2025) improves the quality of chain-of-thought reasoning by directly stimulating neurons identified as critical for reasoning. EELo-CoT (Zhao et al., 2025b) further designs an activation control module that efficiently elicits long chain-of-thought behavior for base models. In this work, we further explore neuron-level signals as a real-time control mechanism for early reasoning exit, thereby enabling more efficient inference.

3 Methodology

We propose **NEAT**, an early reasoning exit framework by monitoring model’s internal neuron activation dynamics. As illustrated in Figure 2, our approach mainly consists of two stages: (1) **Exit Neurons Identification** (§3.1), which identifies a sparse set of neurons whose activations are strongly associated with reasoning exit (2) **Inference-Time Intervention** (§3.2), which monitors these neurons during decoding and intervenes in the reasoning process in real time based on their activation patterns.

3.1 Exit Neurons Identification

Large Reasoning Models (LRMs) generate a reasoning trace $\mathbf{R} = [r^{(1)}, r^{(2)}, \dots, r^{(T)}]$ before producing the final answer, where each r_t represents a reasoning step. The completion of reasoning is signaled by a special termination token w_{end} (e.g., `</think>`), which marks the transition from reasoning to answer generation. Our goal is to identify neurons whose activations provide early internal signals of reasoning termination before the explicit generation of w_{end} . We achieve this by attributing neurons to termination token prediction and selecting those that exhibit consistent activation toward the end of the reasoning trace.

Neuron Attribution. In the inference pass in decoder-only LLMs, for a given input sequence, each layer output \mathbf{h}_i^l (layer l , token position i) is a sum of the previous layer’s output \mathbf{h}_i^{l-1} , the attention output \mathbf{A}_i^l , and the FFN output \mathbf{F}_i^l :

$$\mathbf{h}_i^l = \mathbf{h}_i^{l-1} + \mathbf{A}_i^l + \mathbf{F}_i^l. \quad (1)$$

The FFN output \mathbf{F}_i^l is calculated by a non-linear σ on two MLPs $W_{fc1}^l \in \mathbb{R}^{N \times d}$ and $W_{fc2}^l \in \mathbb{R}^{d \times N}$:

$$\mathbf{F}_i^l = W_{fc2}^l \sigma(W_{fc1}^l (\mathbf{h}_i^{l-1} + \mathbf{A}_i^l)). \quad (2)$$

Following (Geva et al., 2021), the FFN layer output \mathbf{F}_i^l can be represented as a weighted sum over

neuron subvalues:

$$\mathbf{F}_i^l = \sum_{k=1}^N c_{i,k}^l \cdot fc2_k^l, \quad (3)$$

where $fc2_k^l$ denotes the k -th column of W_{fc2}^l , and $c_{i,k}^l$ is the activation of the k -th neuron:

$$c_{i,k}^l = \sigma(fc1_k^l \cdot (\mathbf{h}_i^{l-1} + \mathbf{A}_i^l)), \quad (4)$$

where $fc1_k^l$ denotes the k -th row of W_{fc1}^l .

To quantify the importance of each neuron for generating the termination token (e.g., `</think>`), we adopt the log probability increase method of (Yu and Ananiadou, 2024b). For a neuron in the l -th FFN layer, denoted as v^l , its importance score is defined as the increase in log probability of the target token when v^l is added to the residual stream $\mathbf{A}^l + \mathbf{h}^{l-1}$, compared to the baseline without v^l :

$$\text{Imp}(v^l) = \log p(w_{\text{end}} | v^l + \mathbf{A}^l + \mathbf{h}^{l-1}) - \log p(w_{\text{end}} | \mathbf{A}^l + \mathbf{h}^{l-1}). \quad (5)$$

This approach efficiently identifies neurons whose activations most strongly influence the model’s prediction at the termination position. We retain the top- k neurons with the highest positive importance scores as candidates.

Temporal Filtering. High attribution at a single decoding step is insufficient to characterize neurons that reliably signal reasoning completion. Thus, we furthermore analyze the activation trajectory $\mathbf{c} = [c^{(1)}, \dots, c^{(T)}]$ of each candidate across the reasoning trace. We compute two metrics: the **Relative Center of Mass (CoM)**, which measures where activation concentrates along the trace:

$$\mu_{\text{com}} = \frac{1}{T} \sum_{t=1}^T t \cdot \frac{c^{(t)}}{\|\mathbf{c}\|_1}, \quad (6)$$

and the **Activation Entropy**, which measures temporal dispersion:

$$H(\mathbf{c}) = - \sum_t p_t \log p_t, \quad p_t = \frac{c^{(t)}}{\|\mathbf{c}\|_1}. \quad (7)$$

Neurons with high **CoM** and low **Entropy** exhibit focused, late-stage activation and are selected as exit-associated neurons. We aggregate these metrics over a calibration set \mathcal{D}_{cal} and define the final neuron set as:

$$\mathcal{S}^* = \{v \mid \Omega(v) \geq \tau_{\text{cons}} \wedge \bar{\mu}_{\text{com}}(v) \geq \tau_{\text{com}}\}, \quad (8)$$

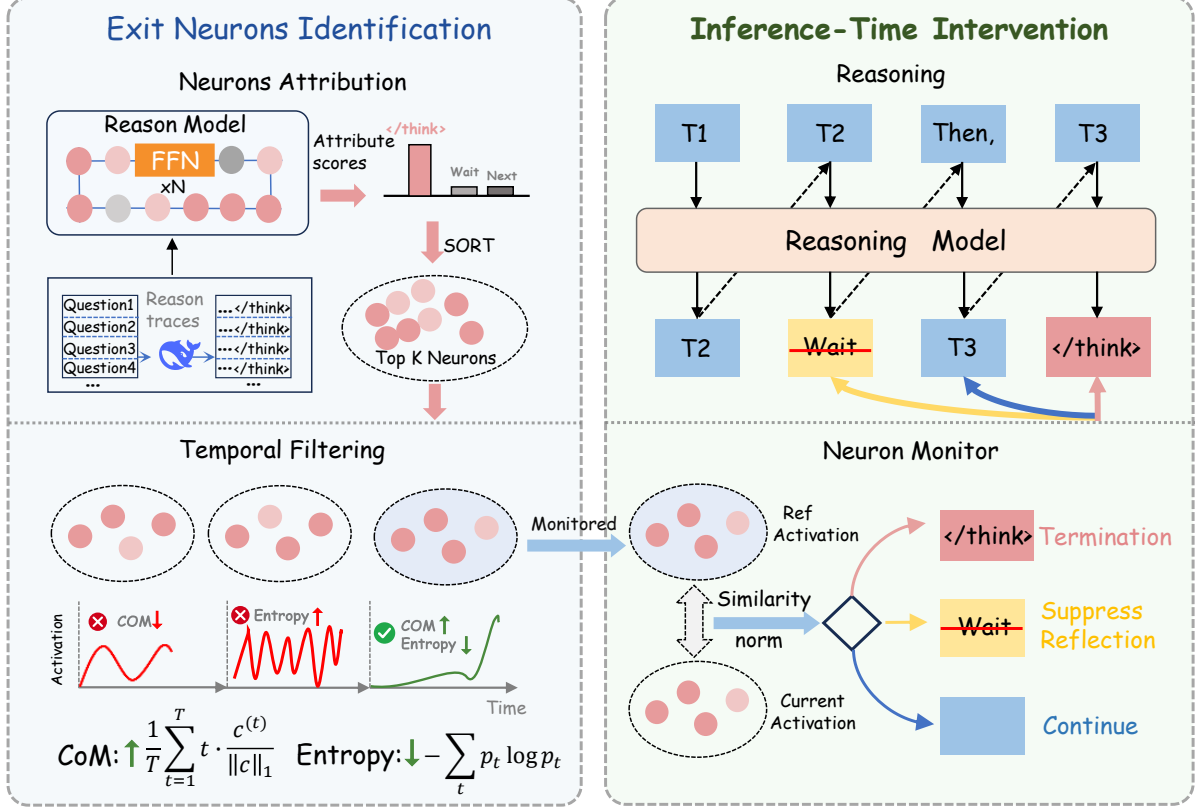


Figure 2: Overview of the proposed NEAT framework. **Left:** During calibration, we identify exit-associated neurons by computing attribution scores at the termination timestep and filtering based on temporal activation patterns. **Right:** During inference, we monitor the activation dynamics of the identified neurons and apply dynamic intervention when the pattern matches the reference regime.

where $\Omega(v)$ is the frequency of neuron v appearing among top- k candidates across samples, $\bar{\mu}_{\text{com}}(v)$ is the average CoM of neuron over \mathcal{D}_{cal} , τ_{cons} is the consistency threshold for filtering neurons that appear frequently, and τ_{com} is the CoM threshold for selecting neurons with late-stage activation patterns.

3.2 Inference-Time Intervention

During inference, the identified neuron set \mathcal{S}^* is held fixed. At each decoding step t , we extract the activation vector $\mathbf{a}_t \in \mathbb{R}^{|\mathcal{S}^*|}$ corresponding to these neurons and compare it against a reference pattern to determine whether to exit, suppress, or continue reasoning, as illustrated in the right panel of Figure 2.

Activation Pattern Matching To assess whether the model has reached an exit-associated state, we compute two alignment measures between the current activation \mathbf{a}_t and a reference pattern $\boldsymbol{\mu}_{\text{ref}}$, which is obtained by averaging the activations of \mathcal{S}^* at the termination timestep over the calibration

set \mathcal{D}_{cal} :

$$\rho_t = \text{CosSim}(\mathbf{a}_t, \boldsymbol{\mu}_{\text{ref}}), \quad \phi_t = \frac{\|\mathbf{a}_t\|_2}{\|\boldsymbol{\mu}_{\text{ref}}\|_2}. \quad (9)$$

Here, ρ_t captures the directional similarity between activation patterns, while ϕ_t measures the relative activation magnitude. Together, they indicate how closely the current state resembles the exit regime.

Dynamic Intervention Strategy Based on these alignment signals, NEAT applies one of three actions at each step:

- **Termination.** If $\rho_t > \tau_{\text{sim}}$ and $\phi_t > \tau_{\text{mag}}$, the activation pattern strongly matches the reference regime, indicating that reasoning has effectively completed. We halt generation and append the termination token $\langle \text{/think} \rangle$.
- **Suppress Reflection.** If $\tau_{\text{sup}} < \rho_t \leq \tau_{\text{sim}}$ and $\phi_t > \tau_{\text{mag}}$ but the termination criterion is not met, we suppress a predefined set of reflection tokens by setting their logits to $-\infty$, reducing unnecessary continuation without forcing immediate exit.

- **Continue.** Otherwise, decoding proceeds normally without intervention.

This hierarchical strategy allows NEAT to aggressively terminate when confidence is high, gently guide the model away from redundant patterns when confidence is moderate, and avoid premature exit when the signal is weak.

4 Experiments

4.1 Experimental Setup

Benchmarks and Metrics. We conduct experiments on a set of mathematical and scientific reasoning benchmarks. For mathematical reasoning, we consider three datasets: MATH500 (Lightman et al., 2023), a collection of 500 multi-step problems spanning algebra, geometry, and probability; AMC23 (AI-MO, 2024), which contains 40 problems from the 2023 American Mathematics Competitions; and AIME24 (MAA Committees, 2025), consisting of 30 challenging problems from the 2024 American Invitational Mathematics Examination. In addition, we evaluate scientific reasoning performance on GPQA Diamond (Rein et al., 2024) (GPQA-D), which includes 198 graduate-level multiple-choice questions across biology, chemistry, and physics. We report three evaluation metrics: *Accuracy* (**Acc**), the average number of generated tokens (**#Tok**), and the *Length Reduction Rate* (**LR**).

Baselines. We compare our method with the following baselines: (i) **Vanilla**, which performs standard decoding without any intervention; **NoThinking** (Ma et al., 2025), which skips the reasoning process entirely. (ii) Prompt-guided methods, including **TALE** (Han et al., 2025), which prompts the model to solve problems within token budgets. (iii) Output-based methods, including **Dynasor** (Fu et al., 2024), which periodically requests intermediate answers at fixed token intervals and exits early if multiple consecutive answers match; **DEER** (Yang et al., 2025), which dynamically truncates chain-of-thought generation by detecting high-confidence intermediate answers; and **CGRS** (Huang et al., 2025), which adjusts the ratio of reflection tokens according to answer confidence.

Implementation Details. For calibration, we randomly sample a subset from the training split of MATH (Hendrycks et al., 2021) and generate reasoning traces using greedy decoding. Attribution

scores (Eq. 5) are computed for FFN neurons at the termination timestep. Temporal Filtering and cross-sample consistency filtering are then applied, resulting in a compact neuron set \mathcal{S}^* , with the exact size depending on the model. During inference, the intervention is triggered based on similarity and magnitude criteria. Each experiment is repeated three times and the average results are reported. We use the vLLM framework (Kwon et al., 2023) for inference acceleration. All decoding is performed with temperature 0.6 and top- p 0.95. Hyperparameter settings are provided in Appendix A.5.

4.2 Main Results

Table 1 compares NEAT with representative baselines across four reasoning benchmarks and four models spanning different architectures and scales. Overall, NEAT consistently achieves around 22% to 28% average token reduction while maintaining comparable or slightly improved accuracy relative to Vanilla decoding, yielding a stable efficiency-accuracy trade-off. Additional results on smaller-scale models are provided in Appendix B.

In contrast, output-based early-exit methods that rely on intermediate answer confidence estimation exhibit substantial variance across models. For example, DEER achieves aggressive compression on stronger and better-calibrated models such as Qwen3, reducing generation length by 41.7% on average, but often at the cost of accuracy. On AIME24 with Qwen3-8B, DEER drops accuracy from 61.1% to 45.6%. Similar behavior is observed on Llama-based models, where both DEER and CGRS suffer notable accuracy degradation while achieving only marginal length reduction. On GPQA-D with Llama-8B, DEER and CGRS reduce accuracy by 7.5 and 9.7 points respectively, with length reduction below 6%, indicating premature or ineffective exits. These results suggest that output-based heuristic exit methods are highly sensitive to model calibration.

In comparison, NEAT more consistently preserves accuracy while maintaining non-trivial compression. Importantly, the difference is not only attributable to calibration sensitivity but also to the intervention strategy. Methods such as DEER and Dynasor employ direct and aggressive termination, immediately halting decoding once their criteria are met, which increases the risk of irreversible errors. By contrast, NEAT adopts a graded exit mechanism, preserving more opportunities for correcting erroneous reasoning paths before termina-

Method	MATH500			AMC23			AIME24			GPQA-D			AVG	
	Acc↑	#Tok↓	LR↑	Acc↑	#Tok↓	LR↑	Acc↑	#Tok↓	LR↑	Acc↑	#Tok↓	LR↑	Acc↑	LR↑
<i>DeepSeek-R1-Distill-Qwen-7B</i>														
Vanilla	92.2	3743	–	87.5	5861	–	52.2	10662	–	53.0	7301	–	71.2	–
<u>NoThinking</u>	80.9	1173	65.4%	75.8	2499	57.4%	32.2	6680	37.3%	37.9	1312	81.8%	56.7	61.4%
TALE	89.1	2657	29.0%	86.7	5107	12.9%	48.9	9727	8.8%	36.2	4938	31.3%	65.2	20.7%
Dynasor	81.8	2070	44.7%	84.2	5201	11.3%	47.8	8334	21.8%	22.2	561	92.3%	59.0	42.5%
DEER	89.6	2272	39.3%	88.3	4670	20.3%	47.8	9288	12.9%	52.5	6691	8.4%	69.6	20.2%
CGRS	92.0	2844	24.0%	88.3	3406	41.9%	52.2	7597	28.8%	53.0	5826	20.2%	71.3	28.7%
NEAT	92.2	2914	22.1%	88.3	4485	23.5%	53.3	8321	22.0%	53.0	5798	20.6%	71.7	22.0%
<i>DeepSeek-R1-Distill-Llama-8B</i>														
Vanilla	85.7	4087	–	84.2	6374	–	44.9	10585	–	49.3	7672	–	66.0	–
<u>NoThinking</u>	83.3	2405	41.2%	82.5	5796	9.1%	40.0	11242	-6.2%	36.2	6638	13.5%	60.5	14.4%
TALE	84.0	3541	13.4%	85.0	5915	7.2%	40.0	11141	-5.3%	46.1	5573	27.4%	63.8	10.7%
Dynasor	85.0	3585	12.3%	84.2	5681	10.9%	37.7	10368	2.1%	31.8	3095	59.7%	59.7	21.2%
DEER	82.3	2722	33.4%	80.8	5480	14.0%	42.2	9778	7.6%	41.8	7434	3.1%	61.8	14.5%
CGRS	84.7	3254	20.4%	86.7	4899	23.1%	47.8	9536	9.9%	39.6	7221	5.9%	64.7	14.8%
NEAT	84.7	2934	28.2%	85.6	4945	22.4%	45.5	8596	18.8%	48.5	5841	23.9%	66.1	23.3%
<i>Qwen3-8B</i>														
Vanilla	94.6	5140	–	89.4	7876	–	61.1	11924	–	57.7	9104	–	75.7	–
<u>NoThinking</u>	87.1	1239	75.7%	72.5	2426	69.2%	30.0	5967	50.0%	54.2	1546	83.0%	61.0	70.0%
TALE	92.3	3885	24.4%	88.3	6872	12.7%	68.9	10942	8.2%	59.1	7113	21.9%	77.2	16.8%
Dynasor	91.7	3841	25.3%	89.2	6457	18.0%	62.2	10174	14.7%	57.7	5965	34.5%	75.2	23.1%
DEER	88.7	1935	62.4%	79.2	3715	52.8%	45.6	7443	37.6%	59.3	7837	13.9%	68.2	41.7%
CGRS	93.3	3507	31.8%	89.2	5595	29.0%	61.1	8792	26.3%	59.8	6302	30.8%	75.8	29.5%
NEAT	95.0	3906	24.0%	88.3	5820	26.1%	60.0	9067	24.0%	59.3	7485	17.8%	75.6	23.0%
<i>Qwen3-14B</i>														
Vanilla	94.1	4551	–	93.3	7190	–	68.9	11316	–	64.0	7411	–	80.1	–
<u>NoThinking</u>	87.0	853	81.3%	77.5	1616	77.5%	27.8	3689	67.4%	56.9	1268	82.9%	62.3	77.3%
TALE	93.7	3389	25.5%	92.5	5951	17.2%	71.1	10860	4.0%	63.8	6091	17.8%	80.3	16.1%
Dynasor	84.4	3667	19.4%	90.0	6030	16.1%	65.6	9775	13.6%	64.3	5775	22.1%	76.1	17.8%
DEER	91.7	1956	57.0%	90.8	4079	43.3%	56.7	6755	40.3%	58.2	6338	14.5%	74.3	38.8%
CGRS	94.5	3235	28.9%	93.3	5076	29.4%	70.0	8662	23.5%	65.2	5953	19.7%	80.8	25.4%
NEAT	94.6	2992	34.3%	93.3	5070	29.5%	70.0	9426	16.7%	64.1	5027	32.2%	<u>80.5</u>	28.2%

Table 1: Comparison across models of different scales and multiple methods on four benchmarks. Acc (%) denotes accuracy, #Tok denotes average response length in tokens, and LR denotes length reduction ratio. Higher Acc and LR, and lower #Tok indicate better performance. Best average results are in **bold**, second best are underlined.

tion. Overall, these results suggest that internal neuron activation dynamics provide a more reliable proxy for early reasoning exit than surface-level signal.

4.3 Inference Latency.

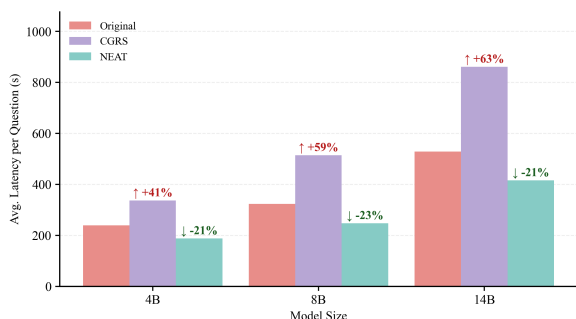


Figure 3: Inference latency on AIME2024 across Qwen3-series models of different scales.

Figure 3 compares inference latency on AIME2024 across Qwen3-series models of different scales. Although output-based heuristic meth-

ods aim to reduce generation length, they can still introduce substantial runtime overhead in practice. As shown in Figure 3, CGRS consistently incurs higher latency than Vanilla decoding, with increases ranging from 41% to 63% across model sizes, despite reducing the number of generated tokens. This additional cost mainly stems from the extra computation required for confidence estimation and decision-making during decoding, which offsets the benefit of shorter outputs. In contrast, NEAT achieves consistent latency reductions of approximately 21% to 23% across all model scales. Since our approach relies solely on monitoring internal neuron activations and does not require extra forward passes, auxiliary scoring modules, or parallel sampling, the reduction in generation length can be more directly translated into wall-clock speedups. These results demonstrate that internal activation-based intervention provides a more computationally efficient and practically scalable alternative to output-based early-exit methods.

Method	Accuracy (\uparrow)	#Tok (\downarrow)
Vanilla	92.2	3743
NEAT	92.2	2914
<i>Suppression-only</i>		
$\tau_{\text{sup}} = 0.6$	93.4	3485
$\tau_{\text{sup}} = 0.4^*$	93.2	3058
$\tau_{\text{sup}} = 0.2$	89.6	2763
<i>Exit-only</i>		
$\tau_{\text{sim}} = 0.8$	92.8	3604
$\tau_{\text{sim}} = 0.6^*$	92.6	3232
$\tau_{\text{sim}} = 0.4$	87.2	2287

Table 2: Ablation of inference-time intervention components for NEAT on MATH500 using DeepSeek-R1-Distill-Qwen-7B. * denotes the default configuration used in our main experiments.

4.4 Ablation Study

Effect of Number of Monitored Neurons. Figure 4 shows the impact of the number of monitored neurons in \mathcal{S}^* . Sensitivity to this choice varies across models. For Qwen3-8B (Figure 4(a)), monitoring too few neurons noticeably degrades accuracy, while accuracy quickly recovers and stabilizes as more neurons are included, with diminishing returns beyond a small threshold. In contrast, DeepSeek-R1-Distill-Qwen-7B (Figure 4(b)) shows much lower sensitivity, maintaining stable accuracy across a wide range of neuron counts. This robustness arises from the combination of neuron attribution and temporal filtering. Neurons are ranked by attribution scores (Eq. 5) and further required to exhibit concentrated late-stage activation, so additional neurons beyond the most relevant ones tend to have weaker or noisier signals and provide limited benefit. Token length varies only moderately with different k , indicating that NEAT largely preserves efficiency as long as a sufficient subset of exit-associated neurons is monitored.

Intervention Components. Table 2 studies inference-time intervention components by separately ablating suppression and exit mechanisms of NEAT on MATH500 with DeepSeek-R1-Distill-Qwen-7B. In the *Suppression-only* setting, the suppression threshold τ_{sup} controls intervention strength without forcing early termination. Moderate suppression improves accuracy while reducing token usage compared to Vanilla decoding, whereas overly strong suppression significantly harms accuracy, suggesting that excessive interference can disrupt valid reasoning even without hard exits. In the *Exit-only* setting, the similarity threshold τ_{sim}

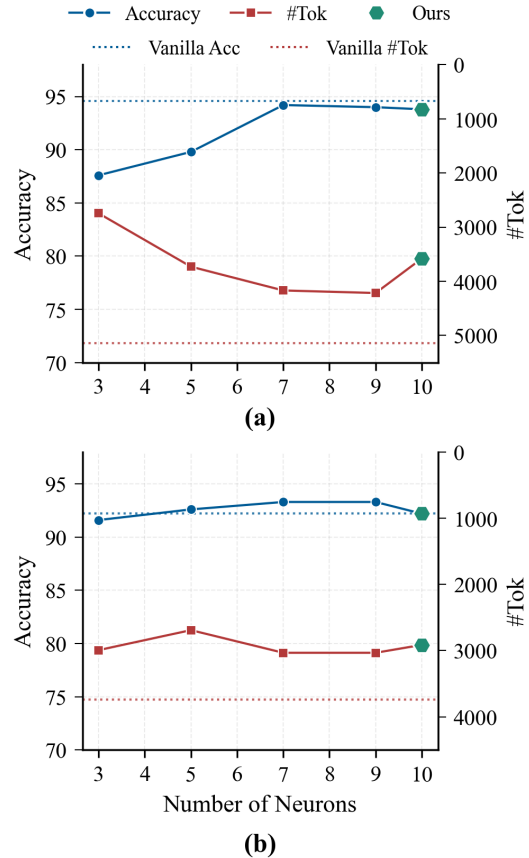


Figure 4: Sensitivity analysis with respect to the number of monitored neurons in \mathcal{S}^* . Blue solid lines denote accuracy (left y-axis), and red dashed lines denote average response length (#Tok, right y-axis), where the right axis is inverted. Horizontal dotted lines indicate the Vanilla baseline. (a) Qwen3-8B. (b) DeepSeek-R1-Distill-Qwen-7B.

determines early termination frequency. Lower thresholds yield greater token savings but cause notable accuracy degradation due to premature exits, while higher thresholds better preserve accuracy but offer limited efficiency gains. Overall, suppression provides smoother and more robust control, while exit decisions are more sensitive to threshold selection. Combining both mechanisms enables NEAT to achieve a better balance between accuracy and inference efficiency, motivating the default configuration used in our main experiments.

4.5 Further Analysis

Comparison of Exit Signals. Table 3 compares different early-exit signals on MATH500, including surface-level logits, hidden states, and neuron-level activations. Implementation details are provided in Appendix A.3.

Among surface-level signals, answer-based con-

Method	Accuracy (\uparrow)	#Tok (\downarrow)
Vanilla	92.2	3743
NoThinking	80.9	1173
<i>Logit-based</i>		
Wait (Wang et al., 2025)	87.8	2829
</think> (Liu and Wang, 2025)	88.8	2373
Answer (Yang et al., 2025)	92.0	2844
<i>Hidden-state-based</i>		
Similarity	80.6	1301
Probe (Zhang et al., 2025a)	82.0	1855
<i>Neuron-based</i>		
Random	77.8	1562
Top Activated	88.8	2458
Ours	92.2	2914

Table 3: Comparison of different exit signals on MATH500 using DeepSeek-R1-Distill-Qwen-7B.

confidence outperforms simple logit heuristics (e.g., </think> or hesitation tokens), but still lags behind Vanilla decoding in accuracy and often incurs additional test-time overhead. Moreover, directly suppressing explicit reasoning degrades performance, suggesting that reflective reasoning remains important for difficult problems.

Hidden-state-based signals are less reliable. Although hidden representations contain confidence-related information, similarity- and probing-based methods achieve large length reductions at the cost of substantial accuracy loss, indicating that they often trigger premature exits.

In contrast, neuron-level signals provide a better balance between accuracy and efficiency. Random or top-activated neurons fail to serve as reliable exit indicators, implying that termination cues are sparse and not solely determined by activation magnitude. By identifying exit-associated neurons, NEAT achieves near-Vanilla accuracy while reducing generation length by over 20%, with significantly lower inference overhead than answer-based signals.

Case Study. Figure 5 presents a representative example illustrating how NEAT avoids redundant reasoning after a correct solution has been reached. In this case, the model correctly determines that the two direction vectors are perpendicular, yielding an angle of 90° . After the correct answer is obtained, the activation patterns of exit-associated neurons progressively align with the reference regime, which triggers suppression and an early exit from further decoding. As a result, NEAT terminates generation after 1,358 tokens while preserving the correct answer and stable reasoning

An example from MATH500

Problem: The set of points (x, y, z) that satisfy $2x = 3y = -3z$ is a line. The set of points (x, y, z) that satisfy $6x = -y = -4z$ is another line. Find the angle between these lines, in degrees.

Ground Truth: 90°

Response:

Okay, so I have this problem where I need to find the angle between two lines in 3D space...

$3 \cdot 2 = 6, 2 \cdot (-12) = -24, -6 \cdot (-3) = 18$

Adding them up: $6 - 24 + 18 = 0$ [Suppression: Sim:0.48]

That means the vectors are perpendicular, so the angle between them is 90 degrees.

I double-check my calculations because that seems a bit surprising. First... Yes, that's correct. So the angle between them is 90 degrees. [Exit: Sim:0.68]

Total 1358 tokens

Vanilla (Continue):

But wait [Reflection], the problem says to round the answer to the direction of the direction vector of the second line.

Hmm, I'm not sure what that means. ...

So, I think the answer is 90 degrees.

But just to be thorough, let me compute the angle using the formula. [Overthinking] ...So, the angle between the two lines is 90 degrees.

Total 2198 tokens

Figure 5: Case study illustrating overthinking in Vanilla decoding and how NEAT avoids redundant continuation.

outcome. By contrast, Vanilla decoding continues for an additional 840 tokens, repeatedly revisiting the same conclusion through reflective and confirmatory reasoning steps. Despite the absence of new information or changes to the final answer, the model engages in extended verification and re-computation, leading to substantial redundant continuation without improving correctness.

5 Conclusion

We study the problem of overthinking in large reasoning models and propose NEAT, a training-free neuron-based framework for early reasoning exit. By identifying exit-associated neurons and monitoring their activation dynamics during inference, NEAT performs early exit or suppresses reflection to reduce redundant reasoning steps. Experiments across multiple benchmarks and model architectures show that NEAT consistently reduces generation length while maintaining accuracy comparable to Vanilla decoding. Overall, our results demonstrate that neuron-level activation dynamics provide an effective and efficient signal for controlling reasoning progression without external supervision or parallel rollouts.

Limitations

Our approach assumes access to internal model activations, which may not be available in all deployment scenarios such as API-only settings, and thus limits applicability to models with full access. In addition, our experiments are conducted on models up to 14B parameters, and it remains an open question whether the same neuron identification and calibration procedure generalizes to substantially larger models. Finally, while activation monitoring introduces minimal overhead in our current implementation, integrating such mechanisms into highly optimized inference engines may require additional engineering effort. We leave these directions for future work.

Ethics Statement

This work focuses on improving inference efficiency in large reasoning models through internal activation analysis. It does not involve human subjects, personal data, or user-generated content. We do not anticipate direct negative societal impacts from the proposed method. As with other techniques that improve model efficiency, responsible deployment should follow existing best practices for large language models.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62272092, No.62172086), National Science Foundation for Young Scientists of China (No. 62502081), and the Fundamental Research Funds for the Central Universities under Grants (N2523011, N25XQD004).

References

- AI-MO. 2024. AMC 2023. <https://huggingface.co/datasets/AI-MO/aimo-validation-amc>. Accessed: 2025-07-26.
- Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17817–17825.
- Roy Eisenstadt, Itamar Zimerman, and Lior Wolf. 2025. Overclocking llm reasoning: Monitoring and controlling thinking path lengths in llms. *arXiv preprint arXiv:2506.07240*.
- Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Aurick Qiao, and Hao Zhang. 2024.

Efficiently serving llm reasoning programs with certaintindex. *arXiv e-prints*, pages arXiv–2412.

- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Daniil Gurgurov, Katharina Trnley, Yusser Al Ghussin, Tanja Baeumel, Josef van Genabith, and Simon Ostermann. 2025. Language arithmetics: Towards systematic language neuron identification and manipulation. *arXiv preprint arXiv:2507.22608*.
- Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. 2025. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. *arXiv preprint arXiv:2508.05337*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*.
- Xin Liu and Lu Wang. 2025. Answer convergence as a signal for early stopping in reasoning. *arXiv preprint arXiv:2506.02536*.
- Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. 2025. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*.

- Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. 2025. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*.
- MAA Committees. 2025. AIME Problems and Solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions. Accessed: 2025-07-26.
- OpenAI. 2024. Introducing openai o1. <https://openai.com/o1/>. Accessed: December 5, 2024.
- Daking Rai and Ziyu Yao. 2024. An investigation of neuron activation as a unified lens to explain chain-of-thought eliciting arithmetic reasoning of llms. *arXiv preprint arXiv:2406.12288*.
- 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*.
- Dan Shi, Renren Jin, Tianhao Shen, Weilong Dong, Xinwei Wu, and Deyi Xiong. 2024. Ircan: Mitigating knowledge conflicts in llm generation via identifying and reweighting context-aware neurons. *Advances in Neural Information Processing Systems*, 37:4997–5024.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.
- Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.
- Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Wayne Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715.
- Yiru Tang, Kun Zhou, Yingqian Min, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, and Shijin Wang. 2025. Enhancing chain-of-thought reasoning via neuron activation differential analysis. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16162–16170.
- Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. 2025. Wait, we don't need to "wait"! removing thinking tokens improves reasoning efficiency. *arXiv preprint arXiv:2506.08343*.
- Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.
- Zeping Yu and Sophia Ananiadou. 2024a. Interpreting arithmetic mechanism in large language models through comparative neuron analysis. *arXiv preprint arXiv:2409.14144*.
- Zeping Yu and Sophia Ananiadou. 2024b. Neuron-level knowledge attribution in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3267–3280.
- Zeping Yu and Sophia Ananiadou. 2025. Locate-then-merge: Neuron-level parameter fusion for mitigating catastrophic forgetting in multimodal llms. *arXiv preprint arXiv:2505.16703*.
- Zeping Yu, Yonatan Belinkov, and Sophia Ananiadou. 2025. Back attention: Understanding and enhancing multi-hop reasoning in large language models. *arXiv preprint arXiv:2502.10835*.
- Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025a. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*.
- Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025b. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*.
- Haoran Zhao, Yuchen Yan, Yongliang Shen, Haolei Xu, Wenqi Zhang, Kaitao Song, Jian Shao, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2025a. Let llms break free from overthinking via self-braking tuning. *arXiv preprint arXiv:2505.14604*.
- Zekai Zhao, Qi Liu, Kun Zhou, Zihan Liu, Yifei Shao, Zhiting Hu, and Biwei Huang. 2025b. Activation control for efficiently eliciting long chain-of-thought ability of language models. *arXiv preprint arXiv:2505.17697*.

A More Implementation Details

A.1 Reflection Token List

Table 4 presents the complete list of reflection tokens used for soft suppression. We include both variants with and without a leading space.

Token	Variant
Wait	" Wait"
Hmm	" Hmm"
But	" But"
Alternatively	" Alternatively"
However	" However"

Table 4: Reflection tokens used for soft suppression.

A.2 Latency Experiment Details

For each model, we conduct latency experiments on a single NVIDIA A6000 GPU (48GB) using the vLLM framework (Kwon et al., 2023). GPU memory utilization is set to 0.9, and the maximum number of generated tokens is set to 16,384. We report the average inference time per question in seconds.

A.3 Exit Signal Implementation Details

Output-based Methods. **Answer** (Huang et al., 2025): We use the answer-based early-exit method proposed in CGRS (Huang et al., 2025). **Wait** (Wang et al., 2025): We suppress reflection tokens (e.g., Wait) during generation. **</think>** (Liu and Wang, 2025): We increase the logits of the termination token to encourage earlier stopping.

Hidden-state-based Methods. **Similarity:** We compute the cosine similarity between the current last-layer hidden state and a reference state, which is obtained by averaging hidden states over calibration samples. The exit threshold is set to 0.4. **Probe** (Zhang et al., 2025a): We use a pretrained linear probe on hidden states to predict reasoning correctness. The exit threshold is set to 0.85.

Neuron-based Methods. **Random:** We randomly select neurons at the termination position, using the same number of neurons and layer distribution as NEAT. The exit threshold is set to 0.6. **Top-Activated:** We select neurons with the highest activation values at the termination position, using the same number of neurons and layer distribution as NEAT. The exit threshold is set to 0.6.

Prompt Template (MATH / AMC / AIME)

```
{problem}

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

Figure 6: Prompt template used for MATH500, AMC23, and AIME24.

Prompt Template (GPQA-D)

```
{problem}

Please reason step by step and output only the choice letter within \boxed{}
```

Figure 7: Prompt template used for GPQA-D.

A.4 Prompt Template

Figure 6 shows the prompt used for MATH / AMC / AIME. Figure 7 shows the prompt we use for dataset GPQA-D.

A.5 Hyperparameter Settings

We evaluate our method on six models spanning different scales and architectures: DeepSeek-R1-Distill-Qwen-1.5B/7B, DeepSeek-R1-Distill-Llama-8B, and Qwen3-4B/8B/14B. The main hyperparameters we use are shown in Table 5. The termination token w_{end} is set to `</think>` for DeepSeek-R1-Distill-Qwen-1.5B/7B, and `**` for all other models. We keep all other parameters the same across all models.

Param.	Value
$ \mathcal{D}_{\text{cal}} $	20
k	300
τ_{com}	0.6
τ_{cons}	0.6
τ_{sim}	0.6
τ_{sup}	0.4
τ_{mag}	0.2
w_{end}	{</think>, **}

Table 5: Hyperparameters used in our experiments.

B Results on Smaller Models

Table 6 presents results on smaller-scale models with 1.5B and 4B parameters. Compared to larger

Method	MATH500			AMC23			AIME24			GPQA-D			AVG	
	Acc↑	#Tok↓	LR↑	Acc↑	#Tok↓	LR↑	Acc↑	#Tok↓	LR↑	Acc↑	#Tok↓	LR↑	Acc↑	LR↑
<i>DeepSeek-R1-Distill-Qwen-1.5B</i>														
Vanilla	84.8	4687	–	72.5	6789	–	36.6	10577	–	36.6	7742	–	57.6	–
NoThinking	71.9	1759	62.5%	63.3	3725	45.1%	18.9	7831	26.0%	27.6	2202	71.6%	45.4	51.3%
TALE	82.9	4387	6.4%	70.0	7486	10.3%	31.0	11701	10.6%	32.3	6886	11.1%	54.1	0.9%
Dynasor	79.3	2678	42.9%	70.8	5661	16.6%	25.6	9745	7.9%	34.5	2998	61.3%	52.6	32.2%
DEER	79.6	2665	43.2%	67.5	5097	25.0%	28.3	9564	9.6%	34.8	6900	10.9%	52.6	22.2%
CGRS	75.5	2174	53.6%	62.5	4180	38.4%	26.7	8082	23.6%	30.0	3257	57.9%	48.7	43.4%
NEAT	82.7	3223	31.2%	72.5	4580	32.5%	31.1	8956	15.3%	35.4	6787	12.3%	55.0	22.8%
<i>Qwen3-4B</i>														
Vanilla	92.7	4796	–	91.7	7449	–	60.0	11449	–	54.0	8112	–	74.6	–
NoThinking	84.9	988	79.4%	70.0	1710	77.0%	24.4	4504	60.7%	47.5	1471	82.1%	56.7	74.8%
TALE	89.1	2657	44.6%	86.7	5107	31.4%	48.9	9727	15.0%	36.2	4938	39.1%	65.2	32.5%
Dynasor	90.1	3877	19.2%	86.7	6233	16.3%	54.3	9912	13.4%	50.5	4398	45.8%	70.4	23.7%
DEER	87.0	1854	61.3%	80.0	3231	56.6%	50.0	6873	40.0%	54.9	7033	13.3%	68.0	42.8%
CGRS	91.3	2704	43.6%	86.7	4351	41.6%	56.7	7893	31.1%	55.2	5229	35.5%	72.5	37.9%
NEAT	94.0	3415	28.8%	89.1	5219	29.9%	57.4	9174	19.9%	55.5	6625	18.3%	74.0	24.2%

Table 6: Comparison of methods on smaller-scale models: DeepSeek-R1-Distill-Qwen-1.5B and Qwen3-4B.

models, smaller models are generally more sensitive to premature termination, making early-exit strategies more challenging. Despite this, NEAT consistently preserves accuracy close to the Vanilla baseline while achieving meaningful length reduction across benchmarks. In contrast, output-based methods often obtain higher compression at the cost of substantial accuracy degradation, particularly on the 1.5B model. These results indicate that neuron-level exit signals remain robust across model scales and enable a stable efficiency-accuracy trade-off even on smaller models.