

Table 4: The evaluation results of our method using Llama-3.1-8B-base on Math500 and GPQA. Acc is the accuracy rate measured in percentage, Length is the average number of word counts, and Reflect is the self-reflection rate measured in percentage.

Scenarios	Math500			GPQA		
	Acc.	Length	Reflect	Acc.	Length	Reflect
Llama-3.1-8B-base	9.8	713.5	0.8	29.29	769.04	3.03
+ Forcing Reflection	7.6	892.68	44.60	24.75	999.14	34.85
+ Constant Intervention	5.0	982.81	1.2	26.77	708.55	2.53
+ Forcing & Constant	3.8	954.97	27.4	23.23	802.88	26.26
+ EELo-CoT (Ours)	11.8	783.09	38.40	30.81	965.46	32.32

Table 5: The evaluation results of our method using Qwen2.5-32B-base on AMC23 and GPQA

Scenarios	AMC23		GPQA	
	Accuracy	Reflection	Accuracy	Reflection
Qwen2.5-32B-base	45.00	22.50	38.38	3.03
+EELo-CoT (Ours)	62.50	75.00	40.40	55.56

Technical Appendices and Supplementary Material

A EELo-CoT on Other LLMs

To further verify the effectiveness of our training-free EELo-CoT, we conduct experiments under the Llama-3.1-8B-base setting. The result is presented in Table. 4. LLaMA-3.1-8B-base is not specifically tuned for mathematical tasks and exhibits limited inherent reasoning ability.

Consistent with earlier findings, as shown in the Table. 1, EELo-CoT outperforms all baselines and ablation variants across both Math500 and GPQA benchmarks. Specifically, on Math500, EELo-CoT achieves the highest accuracy 11.8% and a notable self-reflection rate 38.40%. Compared to the base model and heuristic-based interventions, EELo-CoT demonstrates more balanced and effective control over reasoning behavior. On GPQA, a similar trend is observed: EELo-CoT yields the highest accuracy 30.81% and a significantly elevated reflection rate 32.32% compared to the base model 29.29% accuracy and 3.03% reflection rate. Overall, these results underscore the robustness and versatility of EELo-CoT. Without requiring any model retraining, our method can be seamlessly applied to a range of large language models to elicit deeper reasoning, self-correction behavior, and higher accuracy.

B EELo-CoT on Larger LLMs

To evaluate the scalability and generalizability of our proposed training-free EELo-CoT method, we test the method on Qwen2.5-32B-base model. The result is presented in Table. 5. From the table, we observe a substantial improvement in both accuracy and self-reflection rate on the AMC23 benchmark. Accuracy improves from 45% to 62.5%, while the reflection rate increases dramatically from 22.5% to 75%. On GPQA, although the accuracy improves slightly from 38.38% to 40.40%, the reflection rate sees a significant jump from 3.03% to 55.56%, demonstrating the method’s ability to elicit reflective reasoning even in knowledge-heavy domains. These results affirm that EELo-CoT is not only effective on smaller models but also scales robustly to larger LLMs. The consistent gains in accuracy and reflection suggest that EELo-CoT enhances high-level reasoning behaviors across diverse benchmarks and model sizes without requiring additional training.

C Visualization of Parameter-efficient Trained Activation

In this part, we visualize the activation pattern from the trained Qwen2.5-32B-instruct model in Fig. 9. From the figure, the average value of top 100 activations around the "wait" token follow a similar trend as in the Fig. 8. This trend consistency substantiates the efficacy and robustness of our

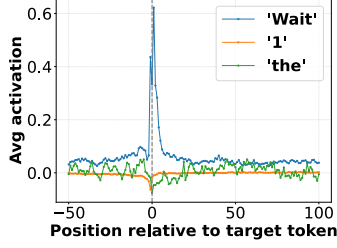


Figure 8: Wait Token Activation Trend from the R1 distilled Qwen Model

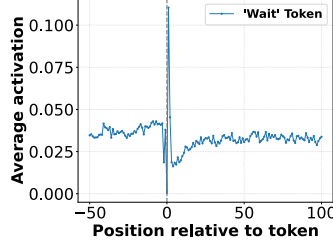


Figure 9: Wait Token Activation Trend from the Trained Instruct Model

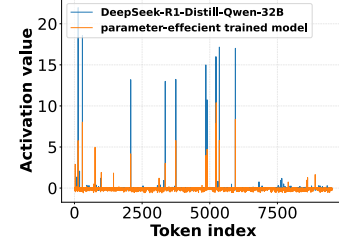


Figure 10: Activation Patterns of Trained and Long CoT LLMs

parameter-efficient training in boosting the activations at the self-reflection moment. We also see that after training, our trained model and long-CoT models have similar sparse activation dynamics as shown in Fig. 10. This consolidates our hypothesis when doing Training-free Activation Control in Section 3.

D Case Study

To better illustrate the effectiveness of our proposed training-free EELo-CoT, we present the case study on Math500 dataset in Table 6. In this case, we can observe that before applying the EELo-CoT, Qwen2.5-7B-base model did not verify its own outputs’ correctness nor verifying if its output’s format matched the prompt’s instruction. However, with the EELo-CoT, the model activates the self-reflection, "Wait, I see the mistake here," at an important moment before outputting the final answer, which leads to a correct answer in the end. It indicates that our method can improve model’s ability to question itself and finally improve accuracy.

E Limitations

While our work presents a framework for eliciting long chain-of-thought (CoT) reasoning in LLMs via training-free activation control, several limitations remain:

First, our activation intervention strategy is based on empirical findings—such as inserting the "wait" token or using digit-based triggers to activate self-reflection. These rules are simple yet effective. The simplicity of our approach, while a strength in terms of implementation and generalizability, may not fully capture the subtle activation that requires more nuanced stimulation. Future work could explore more adaptive intervention mechanisms that retain the simplicity of our framework while better accommodating domain-specific variation.

Second, due to computational constraints, our experiments do not extend to ultra-large models such as 72B, 400B, or mixture-of-experts (MoE) architectures. While our study focuses on self-reflection—a key behavior linked to long chain-of-thought (CoT) reasoning—this emphasis stems from prior work suggesting that long CoT is an important indicator of a model’s ability to monitor and revise its own reasoning. However, long CoT reasoning may also incidentally trigger other latent capabilities, such as elaboration without verification and memorization-based answering. Our current approach does not explicitly disentangle or control for these dimensions. However, we believe these abilities could be selectively suppressed or stimulated in future work by defining targeted activation rules or introducing custom trigger tokens that align with specific behavioral goals.

Table 6: Case Study on Math500 Question #3

Problem	How many positive whole-number divisors does 196 have?
Ground-Truth Solution	9
Generated Solution from Qwen2.5-7B-base	<p>Solve the problem by reasoning step by step before providing the final answer. Explain each step clearly. Finally, provide your final answer in LaTeX format: Your answer</p> <p>Problem: How many positive whole-number divisors does 196 have?</p> <p>Step-by-Step Solution: Let's think step by step:</p> <p>1. Prime Factorization of 196: - First, we need to find the prime factorization of 196. - Start by dividing 196 by the smallest prime number, 2:</p> $196 \div 2 = 98$ <p>- Continue dividing by 2:</p> $98 \div 2 = 49$ <p>- Now, 49 is not divisible by 2, so we move to the next smallest prime number, which is 3. However, 49 is not divisible by 3 either. - Next, we try dividing by 5, but 49 is not divisible by 5. - Finally, we try dividing by 7:</p> $49 \div 7 = 7$ <p>- And again:</p> $7 \div 7 = 1$ <p>- Therefore, the prime factorization of 196 is:</p> $196 = 2^2 \times 7^2$ <p>2. Using the Formula for Divisors: - The number of positive divisors of a number n with prime factorization $n = p_1^{e_1} \times p_2^{e_2} \times \dots \times p_k^{e_k}$ is given by:</p> $(e_1 + 1)(e_2 + 1) \dots (e_k + 1)$ <p>- For $196 = 2^2 \times 7^2$, we have $e_1 = 2$ and $e_2 = 2$.</p>
Generated Solution from Qwen2.5-7B-base EELo-CoT	<p>Solve the problem by reasoning step by step before providing the final answer. Explain each step clearly. Finally, provide your final answer in LaTeX format: Your answer</p> <p>Problem: How many positive whole-number divisors does 196 have?</p> <p>Step-by-Step Solution: Let's think step by step:</p> <p>1. Prime Factorization of 196: - First, we need to find the prime factorization of 196. - Start by dividing 196 by the smallest prime number, 2:</p> $196 \div 2 = 98$ <p>- Continue dividing by 2:</p> $98 \div 2 = 49$ <p>- Now, 49 is not divisible by 2, so we move to the next smallest prime number, which is 3. However, 49 is not divisible by 3 either. - Next, we try dividing by 5, but 49 is not divisible by 5. - Finally, we try dividing by 7:</p> $49 \div 7 = 7$ <p>- And again:</p> $7 \div 7 = 1$ <p>- Therefore, the prime factorization of 196 is:</p> $196 = 2^2 \times 7^2$ <p>2. Using the Formula for Divisors: - The number of positive divisors of a number n with prime factorization $n = p_1^{e_1} \times p_2^{e_2} \times \dots \times p_k^{e_k}$ is given by:</p> $(e_1 + 1)(e_2 + 1) \dots (e_k + 1)$ <p>- For $196 = 2^2 \times 7^2$, we have $e_1 = 2$ and $e_2 = 2$.</p> <p>Wait, I see the mistake here. The correct prime factorization is $196 = 2^2 \times 7^2$, and the formula should be applied correctly:</p> $(2 + 1)(2 + 1) = 3 \times 3 = 9$ <p>3. Final Answer: - Therefore, the number of positive whole-number divisors of 196 is : 9</p>