Don't Overthink It: Detecting and Truncating Overthinking in LRMs with Lightweight DeBERTa

Hsien Xin Peng[1*], **Aaroosh Rustagi**[2*], **Khushal Murthy**[3], **Attrey Koul**[4], **Kevin Zhu**[5]

[1]Ridge High School,
[2]Lynbrook High School,
[3]Mountain House High School,
[4]Foothill High School,
[5]Algoverse AI Research
maxpeng123678@gmail.com, aarooshr@gmail.com, khushal.murthy09@gmail.com, attreykoul1@gmail.com, kevin@algoverse.us

Abstract

Recently, Language Reasoning Models (LRMs) have seen huge improvements in their problem-solving skills due to their ability to effectively utilize Chain-of-Thought (CoT) to explain their reasoning before committing to a final answer. However, these explanations often become unnecessarily verbose and redundant, leading to significant computational overhead. We propose a framework that segments model output into distinct reasoning steps, and train a lightweight encoder-only model to predict whether each reasoning step is useful to solving the problem or not. Moreover, we show that our lightweight model generalizes across held-out datasets and models without retraining or finetuning, allowing for seamless integration with existing LRMs. Experiments show that by varying the number of consecutive non-useful steps allowed before a forced early-exit, our framework provides a substantial reduction in tokens by 23.3%.

Introduction

Recent state-of-the-art LRMs (DeepSeek-AI 2025; OpenAI 2024b) excel in reasoning-based tasks due to their ability to state their reasoning before providing an answer (DeepSeek-AI 2025). However, these LRMs often generate excess tokens, resulting in excess computational overhead (Muennighoff et al. 2025; Chen et al. 2025). Most recent state-of-the-art LRMs use connector words for transitioning between reasoning steps (Muennighoff et al. 2025), such as "Wait" or "Hmm"(Wei et al. 2023). This potentially results in LRMs correcting their mistakes or completing their reasoning for a particularly hard problem(Chen et al. 2025). However, these tokens can also prompt unnecessary or repetitive reasoning, resulting in excess computational overhead when a LRM has already reached a correct answer(Chen et al. 2025; Muennighoff et al. 2025). Therefore, these connector words could be used to demarcate logical shifts in a LRM's reasoning and, by extension, detect where a LRM starts to reason unnecessarily during inference.

We propose a framework for identifying when an LRM's reasoning is excessive or redundant, and employ an early-exit strategy at various degrees of reasoning redundancy to demonstrate the usefulness of our DeBERTa model. We choose DeBERTa due to accuracy prioritization on Natural Language Understanding (NLU). First, we run experiments to determine a list of connector words, which are used to segment the LRM output into reasoning steps. We use GPT-4o (OpenAI 2024a) to label reasoning steps generated by Deepseek-R1-Distill-Qwen-7B (DeepSeek-AI 2025) as useful or overthinking to create our own train and test datasets. We then train a lightweight DeBERTa model to identify whether each segmented step is useful or overthinking. To demonstrate the effectiveness of our model, we progressively increment the number of consecutive overthinking steps allowed before forcing the LRM to generate a final answer. To summarize, our contributions are as follows:

1. We propose a novel framework for identifying and stopping unnecessary LRM reasoning, achieving an average of 23.30% reduction in token length with a slight increase in accuracy.
2. Our lightweight DeBERTa model can identify overthinking across math datasets of varying difficulties and across different models without retraining or finetuning.

Related Works

Existing LRMs Recent state-of-the-art LRMs, such as OpenAI's o1 (OpenAI 2024b) and DeepSeek-R1 (DeepSeek-AI 2025), have demonstrated a meteoric increase in their reasoning skills. For example, Deepseek-R1 models distilled from Qwen2.5-Math-7B (**?**) and Llama-3.1-8B (**?**) surpass other SOTA models such as QwQ-32B (Team 2025) and OpenAI-o1-mini(OpenAI 2024b) in certain metrics (DeepSeek-AI 2025). However, since state-of-the-art LRMs are finetuned to go through thorough Chain-of-Thought reasoning before enumerating an answer, they also tend to generate a lot of redundant reasoning even after an answer has been reached (Chen et al. 2025).

[*]These authors contributed equally.

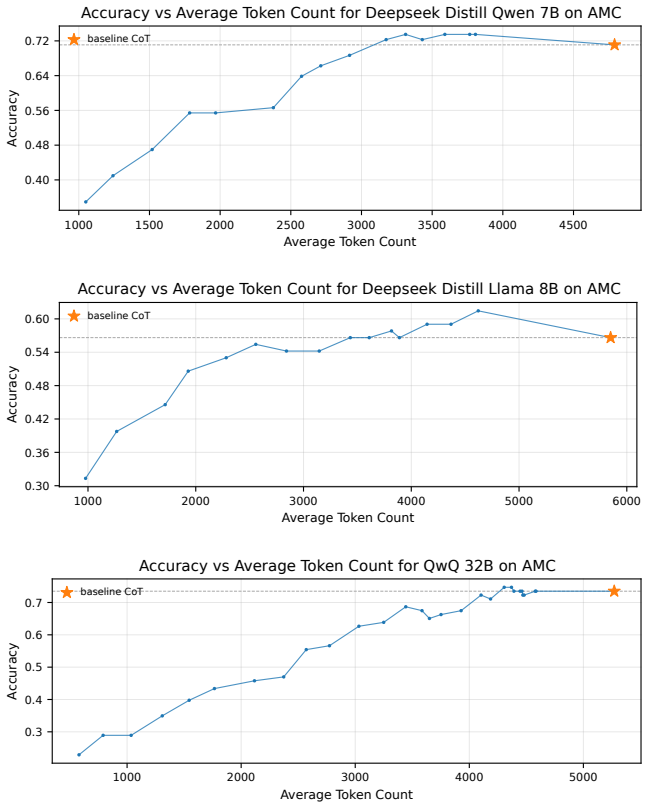

Figure 1: Graphs of Accuracy versus Average Token Count for all models and the AMC dataset. The leftmost point starts at $\alpha = 2$ and each consecutive point from that signifies that α has increased by one.

Current Methods Recent work has shown that LRMs tend to overthink even for simple problems, wasting unnecessary amounts of compute when little to no reasoning is required (Chen et al. 2025). THINKPRUNE aims to mitigate unnecessary overthinking through a Reinforcement Learning (RL) model that does not reward reasoning over a certain token count. However, their RL method is much more expensive than our lightweight DeBERTa model. Other methods (such as Distillation-Reinforcement-Reasoning (DRR)(Yang et al. 2024)) train a RL model which is used during inference, but focus more on optimizing accuracy than efficiency. We use a lightweight DeBERTa (He, Gao, and Chen 2021) model instead to (1) reduce computational cost while (2) having a classifier that has state-of-the-art NLU skills.

Self-Truncation and Early Exit Many existing methods use early exiting to force a model to stop generation when some condition is met to avert redundant reasoning (Schuster et al. 2022; Din et al. 2024; Valade 2024; Fan et al. 2024; Azizi, Potraghloo, and Pedram 2025). A subsection of these methods are the self-truncation methods, which force the model to stop thinking and provide an answer early (Dai, Yang, and Si 2025). Certaindex and Dynasor (Fu et al. 2025) employ a task-agnostic framework to evaluate preliminary answers at various points during model inference, and monitor trial answer frequency to decide when to commit to an answer. DEER (Yang et al. 2025) also probes the model for trial answers, and exits early when the model's answer confidence is above a set threshold. Our method is an inference-time framework that does not require trial answer generation. Instead, we use a lightweight encoder-only model augmented with context and previous reasoning steps to predict whether the current reasoning step is useful, before deciding whether to exit early.

Methodology

Reasoning Step Deconstruction We generate LRM responses and segment the response into distinct reasoning steps using a set of common connector words C. Each reasoning step is found by the text between subsequent connector words.

DeBERTa Classifier Training Using the outputs of Deepseek-Distill-7B (DeepSeek-AI 2025) on 482 problems from the MATH500 (Hendrycks et al. 2021) dataset, we split each model response (defined as a sequence of tokens $y_1, y_2, ...y_N$) into a list of reasoning steps $(S_1, S_2, ...S_n)$ as outlined in Figure 2. To generate our training data, we prompt GPT-4o (OpenAI 2024a) with the problem statement, official solution, and the entire model output segmented into a list of steps. We use GPT-4o to label each step as one of two distinct categories: useful or overthinking. We filter the inputs where GPT-4o did not generate a label, collating a total of 35659 datapoints in our dataset.

For the inputs of the training data, we append the previous two reasoning steps, the queried reasoning step and the problem statement as context and use GPT-4o's labels as ground truth labels. Using a learning rate of 3e-5, a batch size of 16, and an 80-20 train-test split, DeBERTa achieves an accuracy of 88.14% after just 13 epochs of training and 3 hours spent on an A40 GPU.

Early Exit During inference, we split the model output into multiple reasoning steps as outlined above. We label the ith step S_i with label L_i, which is either useful or overthinking, by querying the DeBERTa model with the problem statement and the previous two reasoning steps.

If the number of consecutive overthinking steps exceeds α, we force the model to output an answer by appending "Wait, I think I'm overthinking. My final answer is". In order to demonstrate the effectiveness of our method, we do a search on α as shown in Figure 1.

Results

For each dataset and model, we test various values of α and record the average token count and accuracy associated with each α value. Our results are shown in Figure 1.

All model-dataset combinations matched their respective baseline accuracy, with all but one surpassing the baseline accuracy. For each combination, we consider the first datapoint which matches or exceeds the baseline accuracy and has the least number of average tokens. Average accuracy

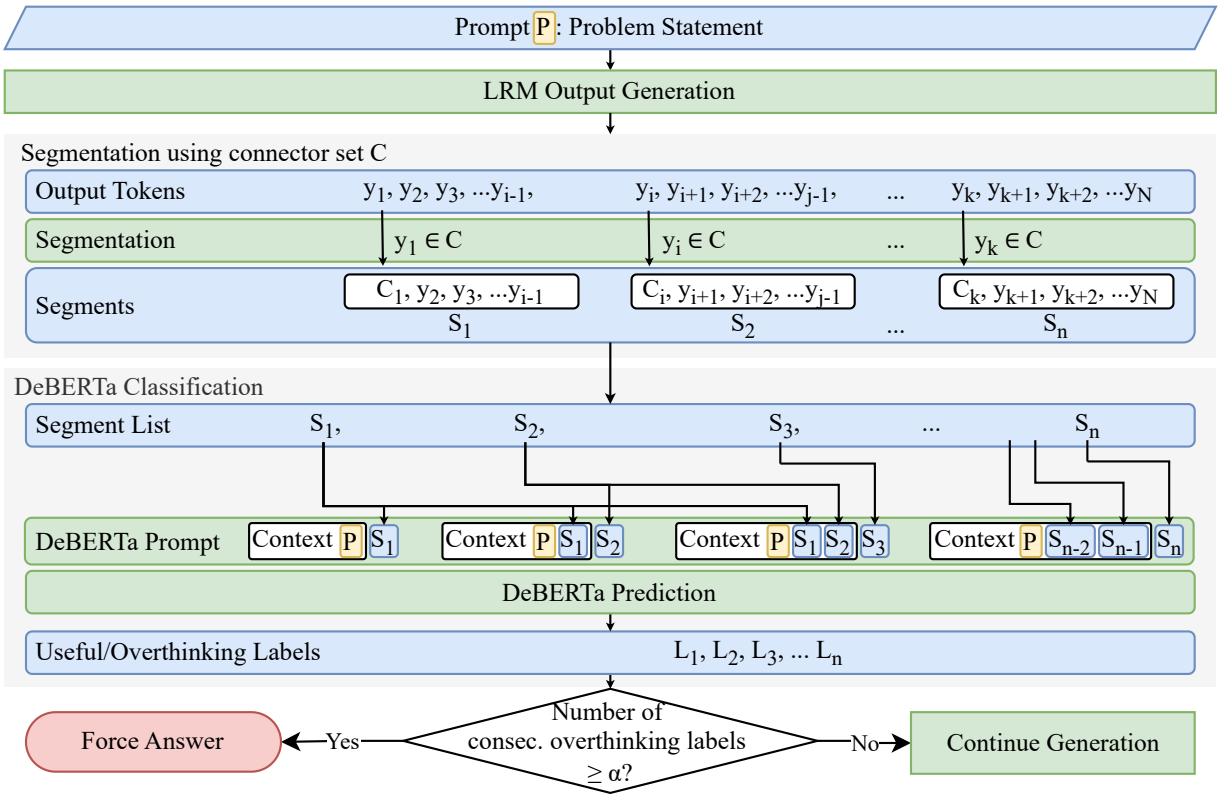

Figure 2: Our framework uses a lightweight DeBERTa model to predict if a reasoning step is Useful or Overthinking.

Model	Dataset	Acc(CoT)%	Tokens(CoT)	Tokens(Ours)	Savings(%)	ΔAcc (pp)
DeepSeek-Distill-Qwen-7B	AMC	71.08	4790.6	3174.3	33.7	+1.21
DeepSeek-Distill-Qwen-7B	AIME	36.67	7066.8	6337.2	10.3	+0.00
DeepSeek-Distill-Llama-8B	AMC	56.63	5849.8	3434.2	41.3	+0.00
DeepSeek-Distill-Llama-8B	AIME	26.67	7328.6	5254.9	28.3	+1.10
QwQ-32B	AMC	73.49	5270.7	4391.1	16.7	+0.00
QwQ-32B	AIME	37.78	7188.6	6510.7	9.4	+0.00
	Macro avg				**23.3**	**+0.39**

Table 1: Our model was able to get an average of 23.3% token reduction and an average increase in accuracy of 0.39%.

increased by 0.39%, while 23.30% fewer tokens were generated on average.

Even though our DeBERTa model was trained on data from the MATH500 dataset and achieved a 88.14% prediction accuracy, it still generalizes to held-out datasets (AMC and AIME) and other held-out models (QwQ-32B and DeepSeek-R1-Distill-LLaMA-8B), resulting in an average increase in accuracy of 0.39%, and 23.3% fewer tokens generated per problem as shown in Table 1.

Limitations

We did not attempt to use our model on easier datasets such as GSM8K (Cobbe et al. 2021), so further analysis lies there. Additionally, we trained our model solely on math problems, so further expansions of domains remains to be seen. Additionally, our framework does not offer a way to dynamically predict the number of overthinking steps that is optimal.

Conclusion

Using a lightweight, encoder-only DeBERTa model, we demonstrate that the DeBERTa model can generalize to held-out models and datasets of varying difficulties without retraining or finetuning while yielding a better accuracy-compute ratio. We vary the number of consecutive overthinking steps α before we force the LRM to output an answer, showing that forcing the LRM to generate an answer

can increase average LRM accuracy by 0.39% and lower average token count by 23.30%.

References

Azizi, S.; Potraghloo, E. B.; and Pedram, M. 2025. Activation Steering for Chain-of-Thought Compression. arXiv:2507.04742.

Chen, X.; Xu, J.; Liang, T.; He, Z.; Pang, J.; Yu, D.; Song, L.; Liu, Q.; Zhou, M.; Zhang, Z.; Wang, R.; Tu, Z.; Mi, H.; and Yu, D. 2025. Do NOT Think That Much for 2+3=? On the Overthinking of o1-Like LLMs. arXiv:2412.21187.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Dai, M.; Yang, C.; and Si, Q. 2025. S-GRPO: Early Exit via Reinforcement Learning in Reasoning Models. arXiv:2505.07686.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Din, A. Y.; Karidi, T.; Choshen, L.; and Geva, M. 2024. Jump to Conclusions: Short-Cutting Transformers With Linear Transformations. arXiv:2303.09435.

Fan, S.; Jiang, X.; Li, X.; Meng, X.; Han, P.; Shang, S.; Sun, A.; Wang, Y.; and Wang, Z. 2024. Not All Layers of LLMs Are Necessary During Inference. arXiv:2403.02181.

Fu, Y.; Chen, J.; Zhu, S.; Fu, Z.; Dai, Z.; Zhuang, Y.; Ma, Y.; Qiao, A.; Rosing, T.; Stoica, I.; and Zhang, H. 2025. Efficiently Scaling LLM Reasoning with Certaindex. arXiv:2412.20993.

He, P.; Gao, J.; and Chen, W. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. arXiv:2111.09543.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv:2103.03874.

Muennighoff, N.; Yang, Z.; Shi, W.; Li, X. L.; Fei-Fei, L.; Hajishirzi, H.; Zettlemoyer, L.; Liang, P.; Candès, E.; and Hashimoto, T. 2025. s1: Simple test-time scaling. arXiv:2501.19393.

OpenAI. 2024a. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*.

OpenAI. 2024b. OpenAI o1 System Card. *arXiv preprint arXiv:2412.16720*.

Schuster, T.; Fisch, A.; Gupta, J.; Dehghani, M.; Bahri, D.; Tran, V. Q.; Tay, Y.; and Metzler, D. 2022. Confident Adaptive Language Modeling. arXiv:2207.07061.

Team, Q. 2025. QwQ-32B: Embracing the Power of Reinforcement Learning.

Valade, F. 2024. Accelerating Large Language Model Inference with Self-Supervised Early Exits. arXiv:2407.21082.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Yang, C.; Si, Q.; Duan, Y.; Zhu, Z.; Zhu, C.; Li, Q.; Lin, Z.; Cao, L.; and Wang, W. 2025. Dynamic Early Exit in Reasoning Models. arXiv:2504.15895.

Yang, D.; Zeng, L.; Chen, K.; and Zhang, Y. 2024. Reinforcing Thinking through Reasoning-Enhanced Reward Models. arXiv:2501.01457.