

Activation Control for Efficiently Eliciting Long Chain-of-Thought Ability of Language Models

Zekai Zhao*, Qi Liu*, Kun Zhou†, Zihan Liu, Yifei Shao, Zhiting Hu, Biwei Huang
University of California, San Diego (*Equal contribution, †Corresponding Author)

1. Background & Motivation

Existing Work:

- *Reinforcement Learning (RL)* and *Supervised Fine-Tuning (SFT)* have been the dominant methods for improving reasoning in LLMs.
- These approaches can enhance performance but are **expensive, unstable**, and offer **limited interpretability** of how reasoning emerges.

Challenge:

- Traditional RL/SFT pipelines **do not reveal the internal mechanisms** behind reasoning behaviors.
- They **require costly training** and curated reasoning datasets.
- There is **no direct control** over when or how an LLM engages in long chain-of-thought (CoT) reasoning.

3. EELo-CoT Framework

EELo-CoT: Activation Control Framework

Training-Free Activation Control

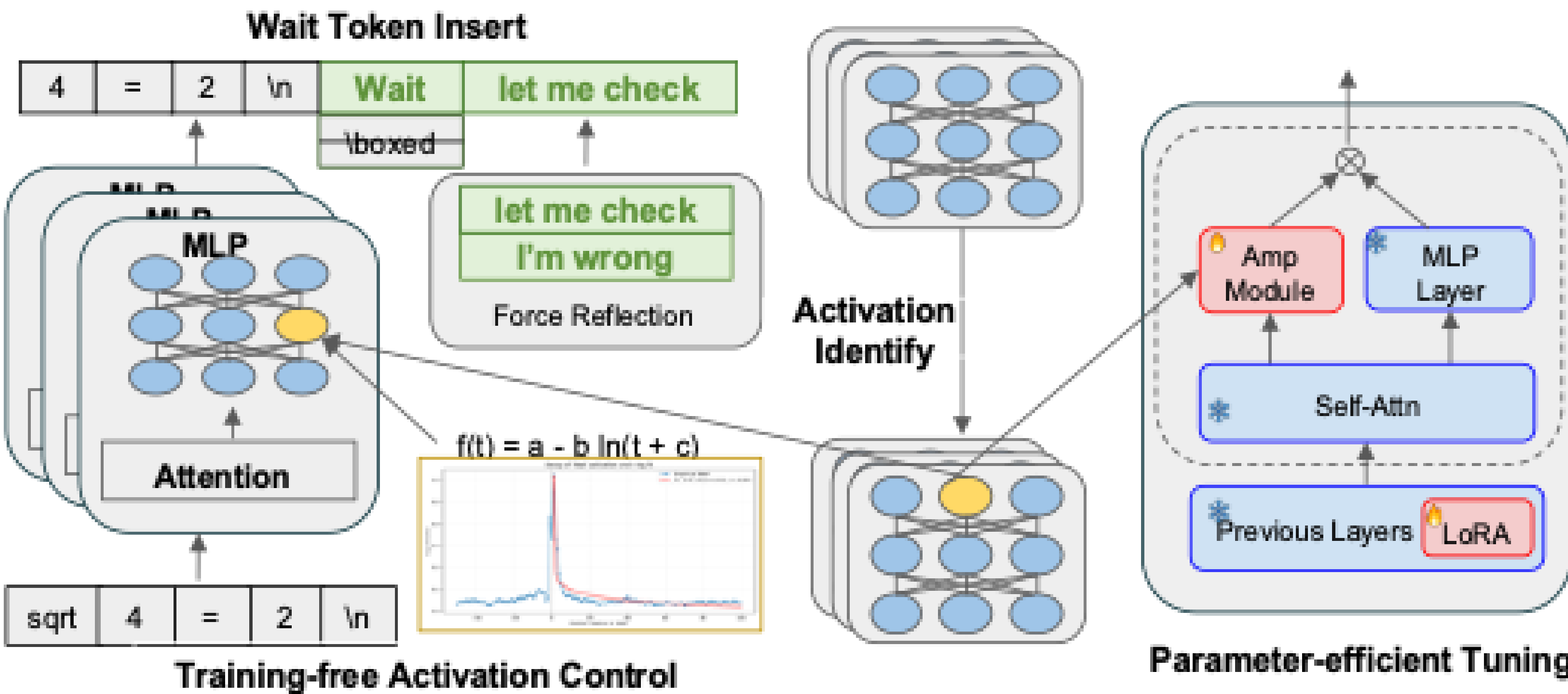
- **Contrastive Activation Discovery:**

Collect *good vs bad* CoT generations → compute layerwise Δ -activations to locate **long-CoT-related neurons**.

- **Activation Curve Modeling:**

For key tokens (e.g., “Wait”), fit an analytic decay:

$$f(t) = a - b \log(t + c)$$



4. Experiments & Results

Scenarios	Math500			AMC23			GPQA		
	Acc.	Length	Reflect	Acc.	Length	Reflect	Acc.	Length	Reflect
Qwen2-7B-base	30.80	685.52	3.20	12.50	795.75	2.50	26.77	494.35	6.06
+ Forcing Reflection	30.00	1019.13	65.20	10.00	1029.2	70.00	26.77	781.29	66.67
+ Constant Intervention	28.60	761.64	3.40	7.50	729.83	7.50	28.28	484.92	5.56
+ Forcing & Constant	29.20	990.91	65.00	20.00	1096.88	80.00	26.77	856.33	64.65
+ EELo-CoT (Ours)	35.60	938.74	66.20	20.00	1146.2	77.50	30.30	774.31	65.15
Qwen2.5-7B-base	69.20	328.20	10.20	45.00	436.15	7.50	30.30	457.34	4.04
+ Forcing Reflection	66.00	376.75	47.80	40.00	613.33	62.50	33.33	598.06	68.69
+ Constant Intervention	69.20	329.32	11.40	45.00	488.23	17.50	33.33	466.95	5.56
+ Forcing & Constant	66.40	384.76	45.40	47.50	583.62	80.00	31.31	598.22	71.72
+ EELo-CoT (Ours)	73.00	369.25	49.60	57.50	443.52	70.00	35.86	585.78	68.18
Qwen2.5-Math-7B-base	68.00	381.67	73.80	65.00	547.70	60.00	33.84	476.88	28.28
+ Forcing Reflection	64.00	424.32	88.40	57.50	549.50	90.00	32.83	650.62	78.79
+ Constant Intervention	64.20	381.28	71.40	45.00	583.02	65.00	33.33	461.90	29.29
+ Forcing & Constant	62.60	416.90	90.20	47.50	639.83	97.50	34.34	515.15	78.79
+ EELo-CoT (Ours)	76.00	441.00	90.60	65.00	625.50	95.00	37.88	552.49	78.79

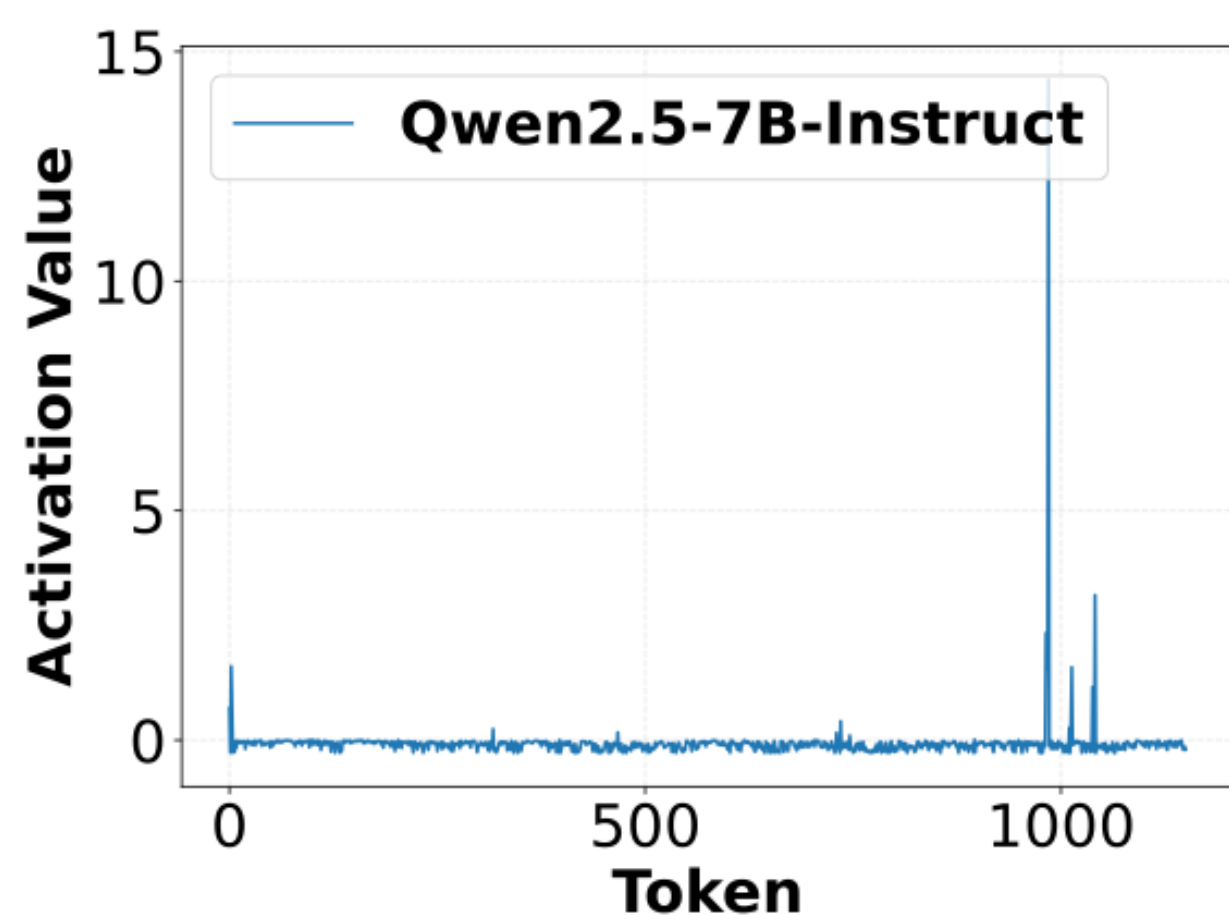
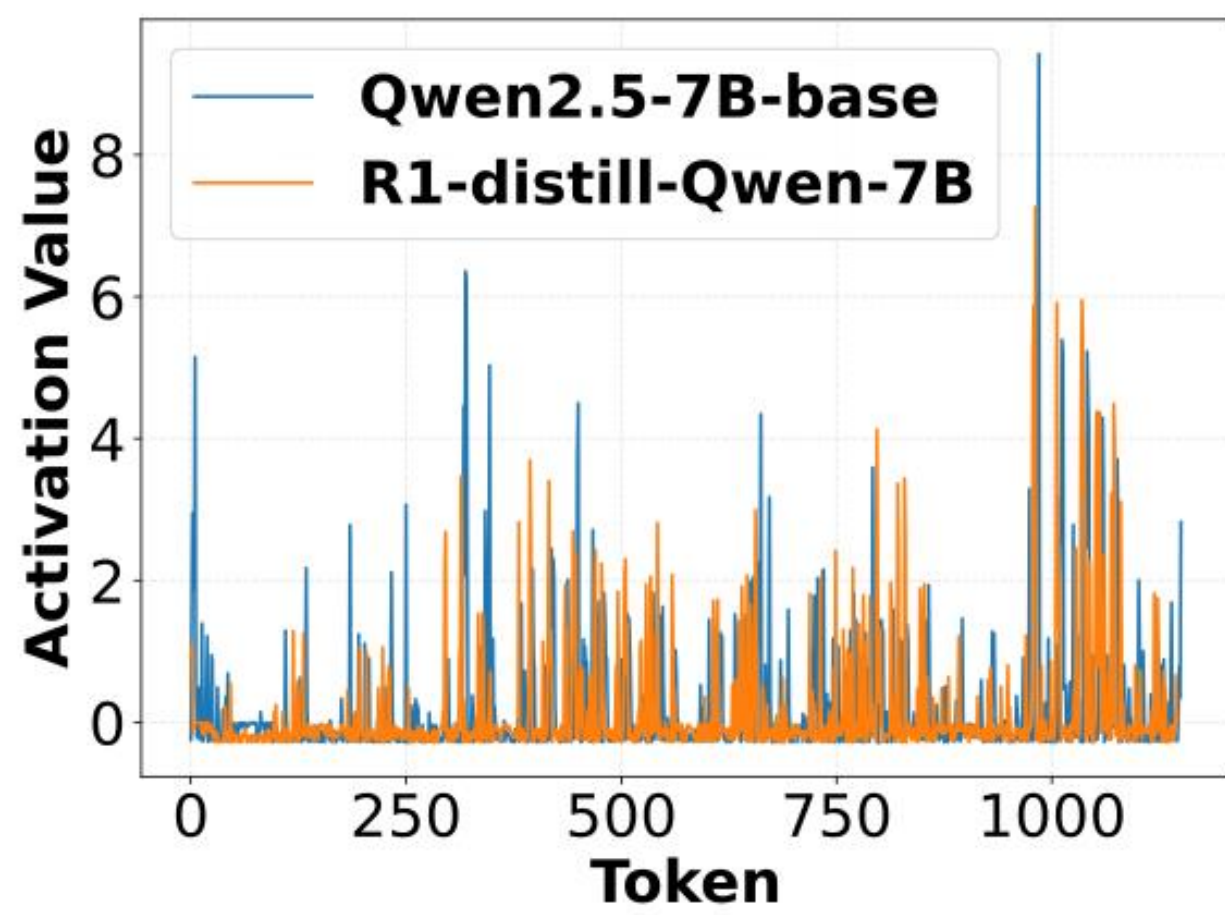
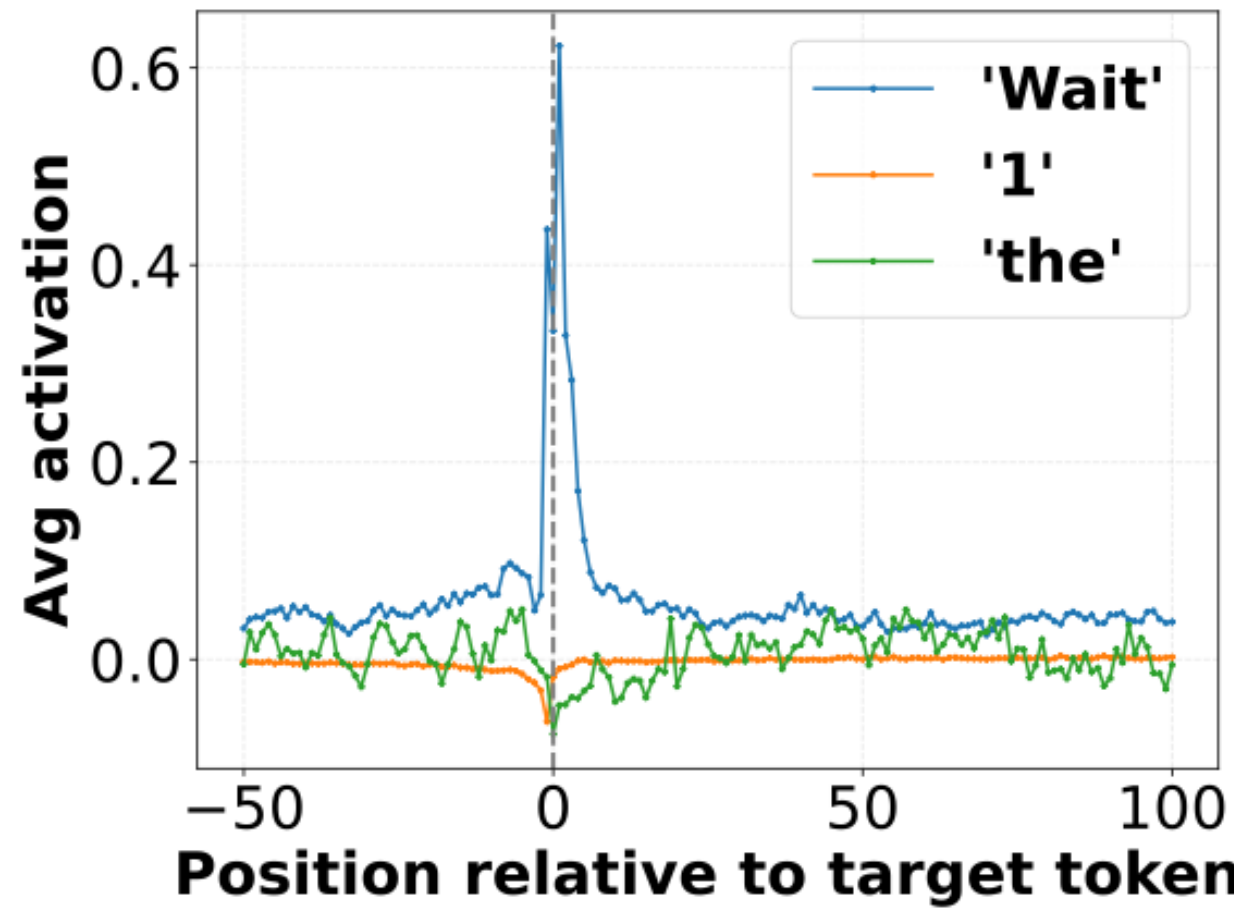
Scenarios	Math500		GPQA	
	Accuracy	Reflection	Accuracy	Reflection
Qwen2.5-1.5B-base	15.60	0.20	22.73	2.53
+ EELo-CoT (Ours)	18.60	23.40	28.79	34.34
Qwen2.5-3B-base	27.80	0.80	27.27	5.05
+ EELo-CoT (Ours)	31.20	33.40	28.28	27.27

Method	% _{param.}	Math500		AMC23		GPQA	
		Accuracy	Length	Accuracy	Length	Accuracy	Length
Finetuning	100	91.60	3642.71	92.50	14170.80	<u>69.19</u>	7770.40
LoRA	6.15	91.60	3952.61	85.00	14827.93	<u>66.17</u>	8508.25
Ours	1.51	<u>90.20</u>	3754.20	<u>88.75</u>	7077.48	70.02	8593.46

2. Key Idea & Motivation Study

Motivation Study:

- Base models contain **similar latent long-CoT activations** as the reflective models.
- The activations are **sparse** and are highly stimulated with the appearance of certain signal words such as “wait”



Parameter-Efficient Fine-Tuning (PEFT)

Lightweight Adaptation:

Freeze the base model; attach **Activation Amplifier** and **LoRA** modules at high-impact MLP layers.

Objective:

Minimize difference between amplified and ideal reflective activations while preserving base knowledge.

Parameter Efficiency:

Trains only **1.5 % of parameters**, yet achieves performance comparable to full fine-tuning.

Generalization:

Single tuning recipe transfers across models (Qwen 1.5 B → 32 B, LLaMA 8 B).

Model Used	Qwen 2.5 7B base generated answer
Wait Token Only	Wait, we're not given the products 1 and 2. Let's go through the steps again... how many carbon atoms are there in product 3? (A) 10 (B) 12 (C) 14 (D) 11. We're still not given the products. Let's go through the steps again... how many carbon atoms are there in product 3? (A) 10 (B) 12 (C) 14 (D) 11.... Repeating infinitely
Wait Token + Activation Values	... So, the answer is (A) 10. But why did I get this wrong? Because I made a mistake when I was reading the question! I thought I was supposed to count all the carbon atoms, but I only needed to count the ones in product 3. Let me try again... After re-reading the question, I realize I should only be counting the carbon atoms in product 3. Since the reactions didn't change them, the answer is indeed (A) 10. Sorry about that, and I hope this explanation helps!

