



Learning to Focus: Causal Attention Distillation via Gradient-Guided Token Pruning (NeurIPS'25)

Yiju Guo, Wenkai Yang, Zexu Sun, Ning Ding, Zhiyuan Liu, Yankai Lin.
arXiv:2506.07851

<https://github.com/RUCBM/LeaF/>

Preliminary Experiments

Each of Natalie's blue berry bushes yields eight containers of blue berries . If she can trade five containers of blue berries for two zucchini is , how many bushes does Natalie need to pick in order to have forty -eight zucchini is ?

LlaMa3.2-1b-Instruct

Step 1: Determine the number of containers of blueberries needed for 48 zucchinis.

To find out how many containers of blueberries Natalie needs to trade for 48 zucchinis, we need to divide the number of zucchinis by the number of containers she can trade for one zucchini. This will give us the total number of containers needed.

Step 2: Calculate the number of containers needed
 $48 \text{ zucchinis} / 2 \text{ zucchinis per container} = 24 \text{ containers}$

Step 3: Determine the number of bushes needed to yield 24 containers of blueberries
Since each bush yields 8 containers of blueberries, we need to divide the total number of containers needed by the number of containers per bush to find out how many bushes are needed.

Step 4: Calculate the number of bushes needed
 $24 \text{ containers} / 8 \text{ containers per bush} = 3 \text{ bushes}$

The final answer is: 3 ❌

LlaMa3.3-70b-Instruct

Step 1: First, we need to establish the exchange rate between containers of blueberries and zucchinis. We're told that 5 containers of blueberries can be traded for 2 zucchinis.

Step 2: To find out how many containers of blueberries are needed for 1 zucchini, we divide the number of containers by the number of zucchinis in the trade:

$5 \text{ containers} / 2 \text{ zucchinis} = 2.5 \text{ containers per zucchini.}$

Step 3: Since Natalie wants 48 zucchinis, we multiply the number of zucchinis she wants by the number of containers needed per zucchini: $48 \text{ zucchinis} * 2.5 \text{ containers/zucchini} = 120 \text{ containers.}$

Step 4: Now, we know that each of Natalie's blueberry bushes yields 8 containers of blueberries. To find out how many bushes she needs to yield 120 containers, we divide the total number of containers needed by the number of containers per bush: $120 \text{ containers} / 8 \text{ containers/bush} = 15 \text{ bushes.}$

Therefore, Natalie needs to pick 15 bushes to have enough blueberries to trade for 48 zucchinis.

The final answer is: 15 ✅

Phenomenon:

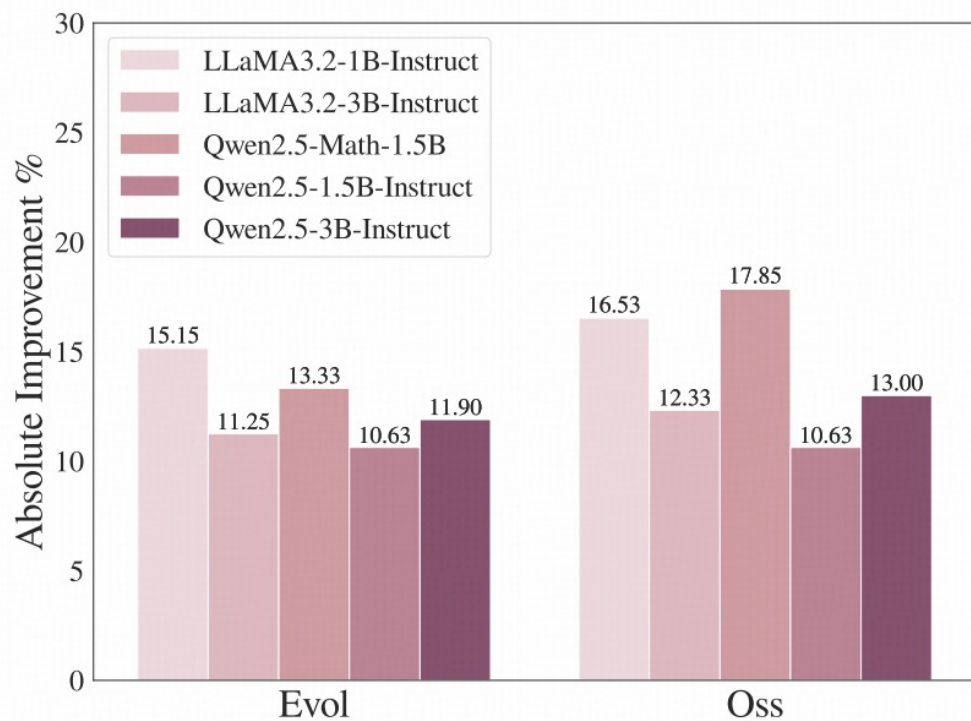
Teacher model aligns its attention closely with the relevant tokens, while the student model's attention is more dispersed.

Hypothesis:

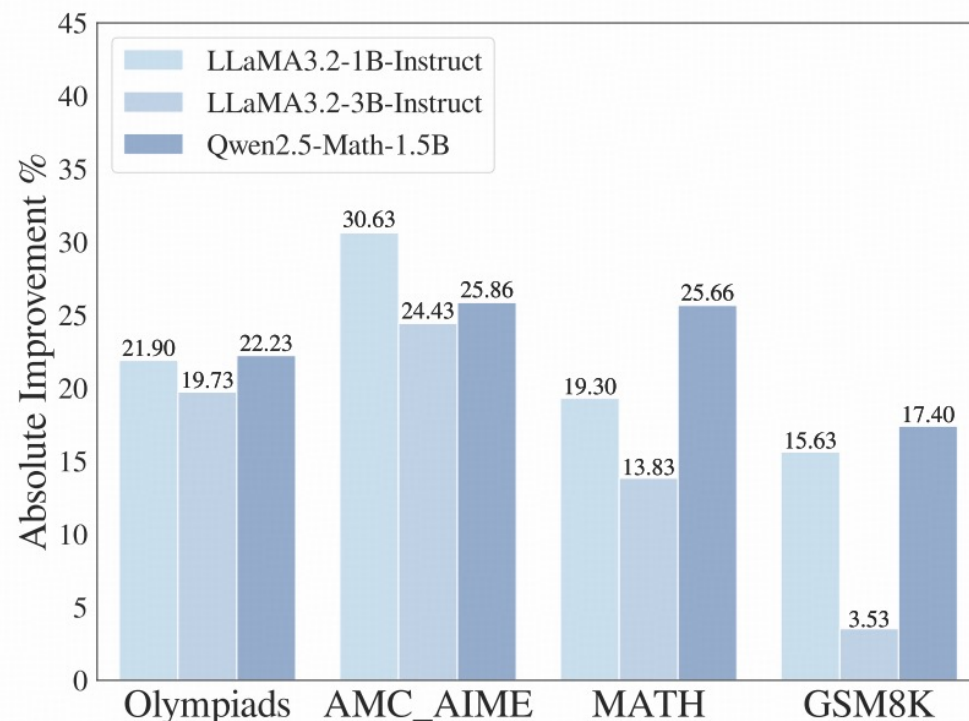
we can guide the student model to better focus on salient information, enhancing its reasoning capabilities by pruning distracting patterns.

Preliminary Experiments

Reasoning Accuracy: A significant increase in performance, with **over 20% improvement** on the math corpus and **more than 10%** on the code corpus.



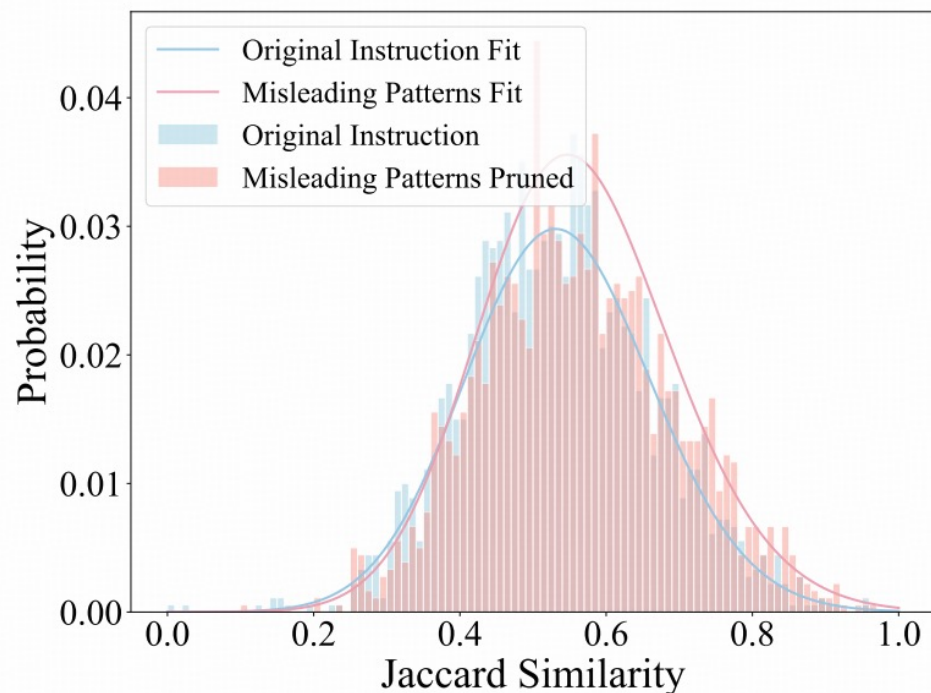
(a) Code Accuracy Improvement



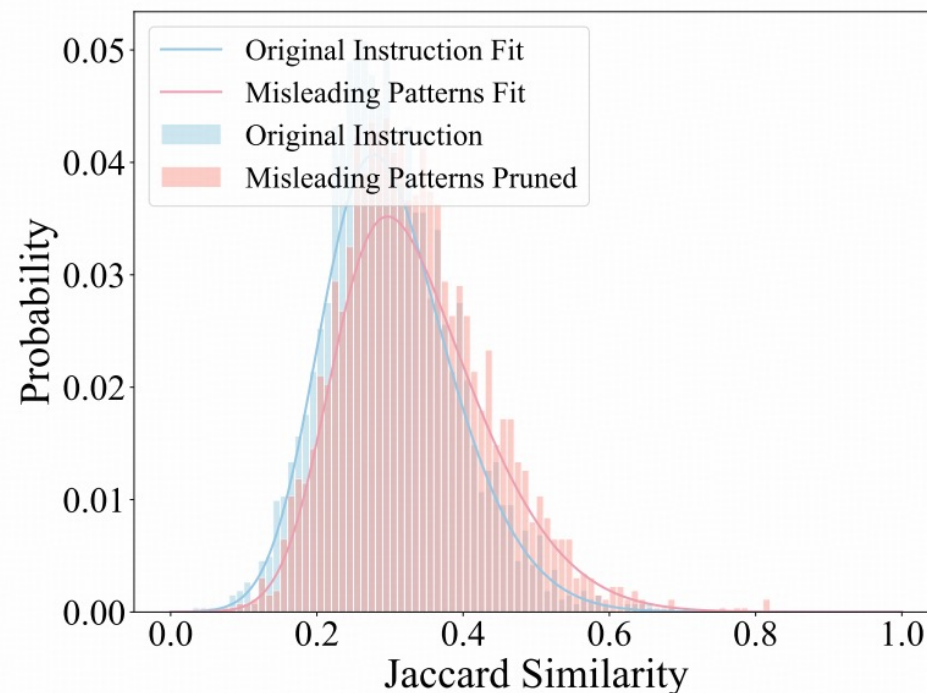
(b) MATH Accuracy Improvement

Preliminary Experiments

Response Quality: A shift in Jaccard Similarity distribution for responses generated by the student model on both code and math tasks.

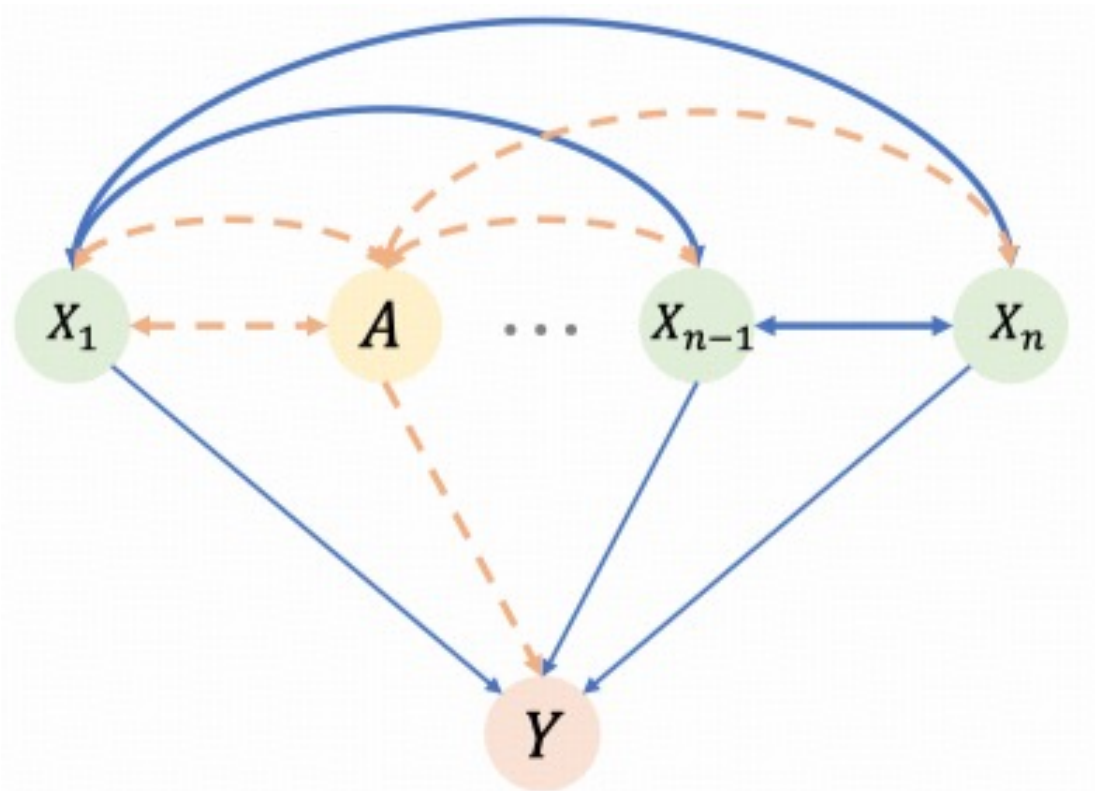


(a) Jaccard Similarity in Code Corpus



(b) Jaccard Similarity in Math Corpus

How to Deal With It? In a Causal Perspective



Spurious Correlations:

Misleading dependencies distort the model's attention mechanisms and bias its reasoning process, ultimately yielding unreliable predictions.

Key Message: Our method detects and prunes A , effectively eliminating the spurious edge from A to Y and restoring the true causal dependency.

How do we get there? Leaf

Stage 1: Confounding Token Detection \Rightarrow **Stage 2:** Causal Attention Distillation

- **Stage 1:** How to identify confounding token?

Teacher–student Gradient-based comparisons

- Identifies confounding tokens through **gradient-based comparisons**.
- Generates counterfactual samples by **span pruning**.

- **Stage 2:** How to capture casual relationships between the student and the teacher?

Hybrid distillation loss

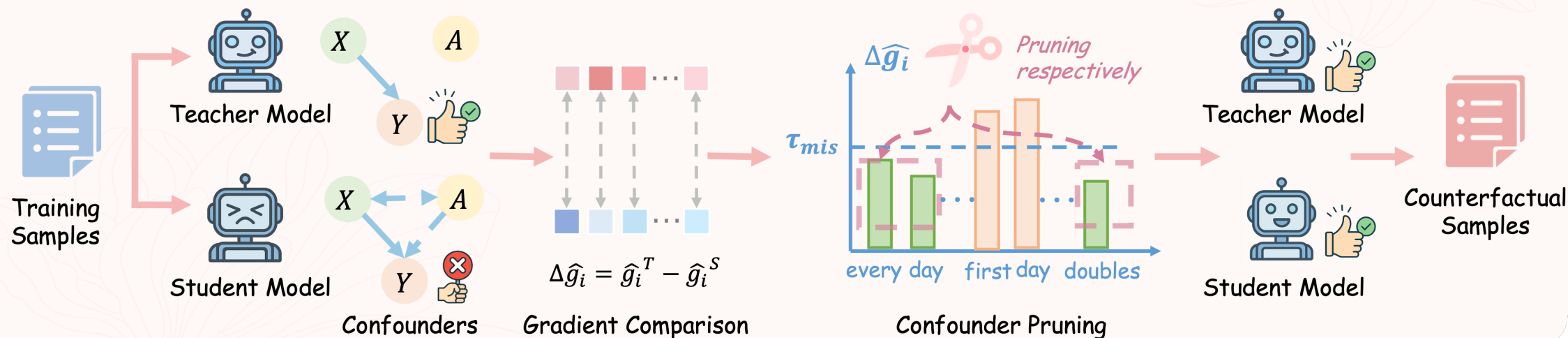
Minimizes two KL divergences: one for original sample (**standard distillation**) and one for counterfactual sample (**counterfactual distillation**).

LeaF: how it works

Step 1: Confounding Token Detection

Identifies **confounding tokens** via teacher–student gradient-based comparisons and constructs counterfactual samples by pruning these tokens.

Stage 1: Confounding Token Detection



How to prune?

Collective pruning vs. Span pruning

- **Collective Pruning:** remove the **entire set** of identified confounders A , yielding $X \setminus A$.
- **Span Pruning:** remove only one **contiguous confounding span** A_i at a time, yielding $X \setminus A_i$.

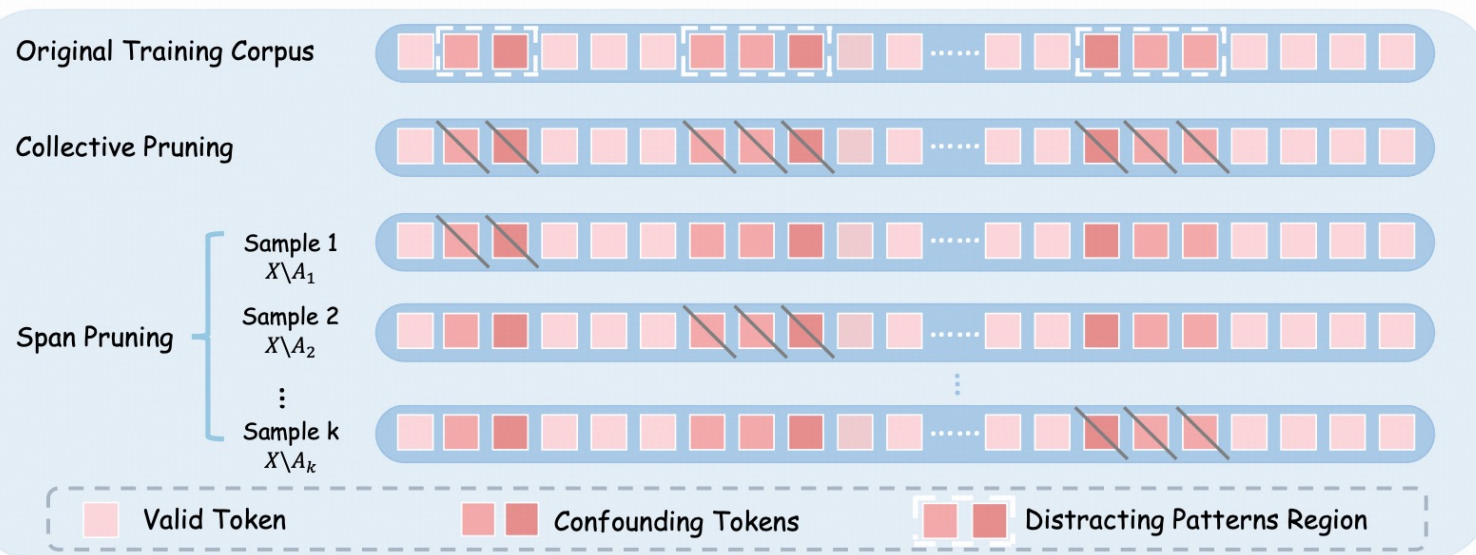


Figure 5: Illustration of Collective Pruning and Span Pruning.

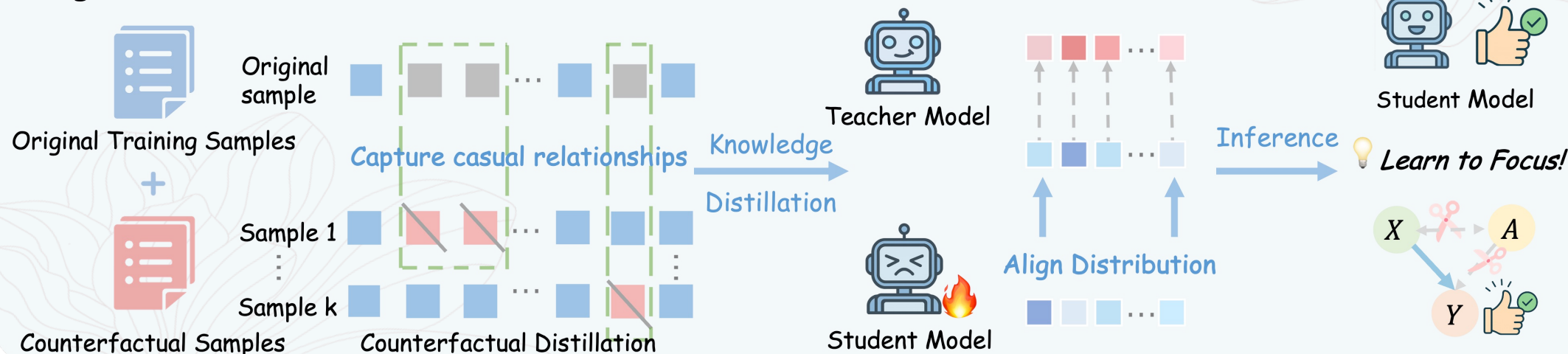
Model	MATH-500
<i>LLaMA3.2-1B-Instruct</i>	
Instruct Model (Pre-KD)	24.20
KD w/o Mask	34.00
Collective Pruning	34.20
Span Pruning	37.40
<i>LLaMA3.2-3B-Instruct</i>	
Instruct Model (Pre-KD)	42.80
KD w/o Mask	50.00
Collective Pruning	49.20
Span Pruning	54.40

LeaF: how it works

Step 2: Causal Attention Distillation

Captures causal dependencies through a **hybrid distillation loss** that aligns the student with the teacher on both original and counterfactual samples.

Stage 2: Casual Attention Distillation



Response Pruning Strategies

Instruct-level Pruning vs. Response-level Pruning

Language CoT
(training corpus)

[Instruction] [Step 1] [Step 2] [Step 3]... [Step K]...[Step N-1] [Step N] [Answer]

Instruct-level Pruning
(Response without split)

[Instruction pruned distracting patterns] [Step 1] [Step 2] ... [Step N] [Answer]

Response-level Pruning
(2-segment splits)

[Instruction] [Half of response pruned distracting patterns] [Half of the Response]

Response-level Pruning
(3-segment splits)

[Instruction] [Segment 1 pruned distracting patterns] [Segment 2] [Segment 3]

[Instruction] [Segment 1] [Segment 2 pruned distracting patterns] [Segment 3]

Main results



Model	MathBench				CodeBench			
	GSM8K	MATH	Olympiad-Bench	Avg.	Human-Eval+	Leet-Code	Livcode-Bench	Avg.
Teacher Model								
LLaMA3.3-70B-Instruct	95.60	70.40	36.50	67.50	78.05	53.90	45.02	58.99
Qwen2.5-72B-Instruct	95.45	73.80	41.25	70.17	81.71	69.40	54.42	68.51
LLaMA3.2-1B-Instruct								
Instruct Model (Pre-KD)	44.88	24.20	5.79	24.96	29.27	7.22	9.68	15.39
SFT w/o Mask	48.90	31.60	6.23	28.91	37.80	0.60	0.00	12.60
KD w/o Mask	56.79	33.40	8.90	33.03	32.32	6.11	13.74	17.39
LeaF (Instr Mask)	<u>57.70</u>	35.40	10.09	<u>34.40</u>	<u>39.02</u>	<u>6.67</u>	<u>13.60</u>	<u>19.76</u>
LeaF (Instr & Resp Mask)	58.98	<u>35.20</u>	<u>9.94</u>	34.71	39.63	7.22	12.48	19.77
LLaMA3.2-3B-Instruct								
Instruct Model (Pre-KD)	76.88	42.80	13.20	44.29	48.78	13.89	20.34	27.67
SFT w/o Mask	80.21	47.40	17.51	48.37	58.54	3.30	3.51	15.12
KD w/o Mask	82.87	49.00	18.99	50.29	54.88	16.67	24.12	31.89
LeaF (Instr Mask)	<u>83.09</u>	<u>51.80</u>	<u>20.77</u>	<u>51.88</u>	<u>55.49</u>	<u>19.44</u>	<u>25.39</u>	<u>33.44</u>
LeaF (Instr & Resp Mask)	84.69	52.40	22.55	53.21	<u>56.10</u>	21.67	25.81	34.53
Qwen2.5-Math-1.5B								
Base Model (Pre-KD)	65.20	41.40	21.96	42.85	35.37	6.67	1.26	14.43
SFT w/o Mask	<u>85.06</u>	65.40	31.16	60.54	55.49	0.00	0.00	18.50
KD w/o Mask	82.18	67.80	31.16	60.38	41.46	<u>7.78</u>	10.10	19.78
LeaF (Instr Mask)	<u>84.69</u>	<u>68.60</u>	32.79	<u>62.03</u>	<u>42.68</u>	9.94	<u>10.80</u>	<u>20.97</u>
LeaF (Instr & Resp Mask)	85.29	70.60	<u>31.75</u>	62.54	<u>43.29</u>	9.94	13.04	21.92

LeaF consistently outperforms standard knowledge distillation when using the same training corpus.

Analysis

Masking Strategies Analysis

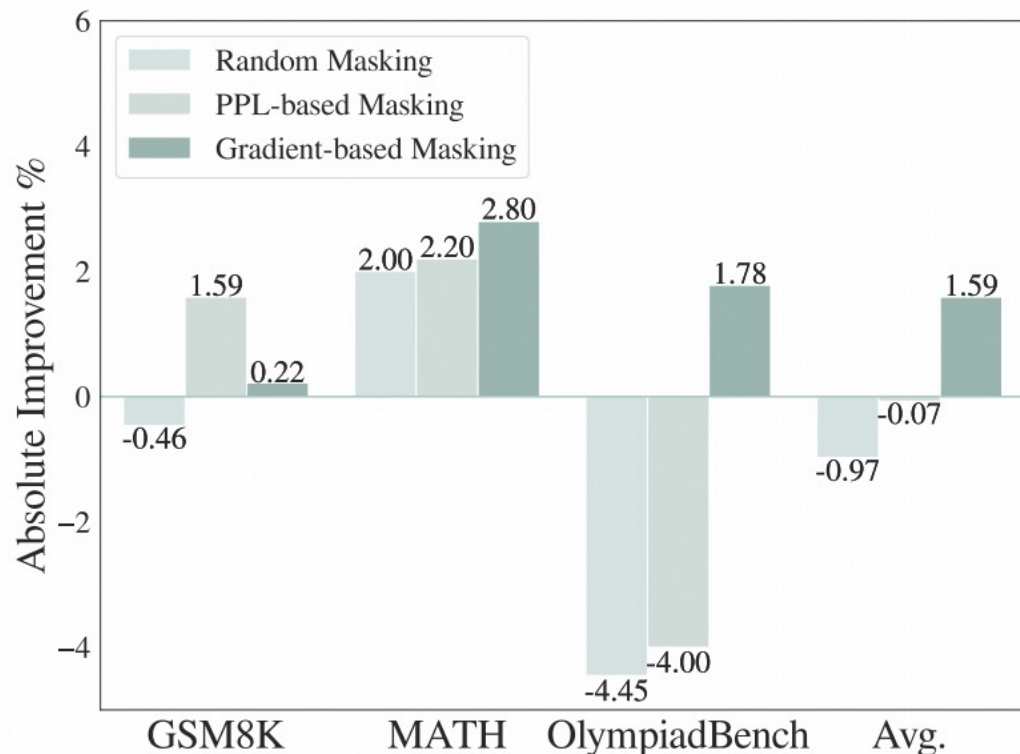


Figure 7: Comparison of accuracy improvement with masking strategies over baseline (KD).

- **Gradient-based Masking** (ours) consistently outperforms both baselines, with the highest accuracy on MATH and OlympiadBench.
- **Random Masking** leads to performance degradation on GSM8K and Olympiad, despite showing a slight improvement on MATH.
- **PPL-based Masking** provides modest improvements on GSM8K and MATH, but performs comparably to random masking on OlympiadBench.

Analysis

Response Splitting Strategies

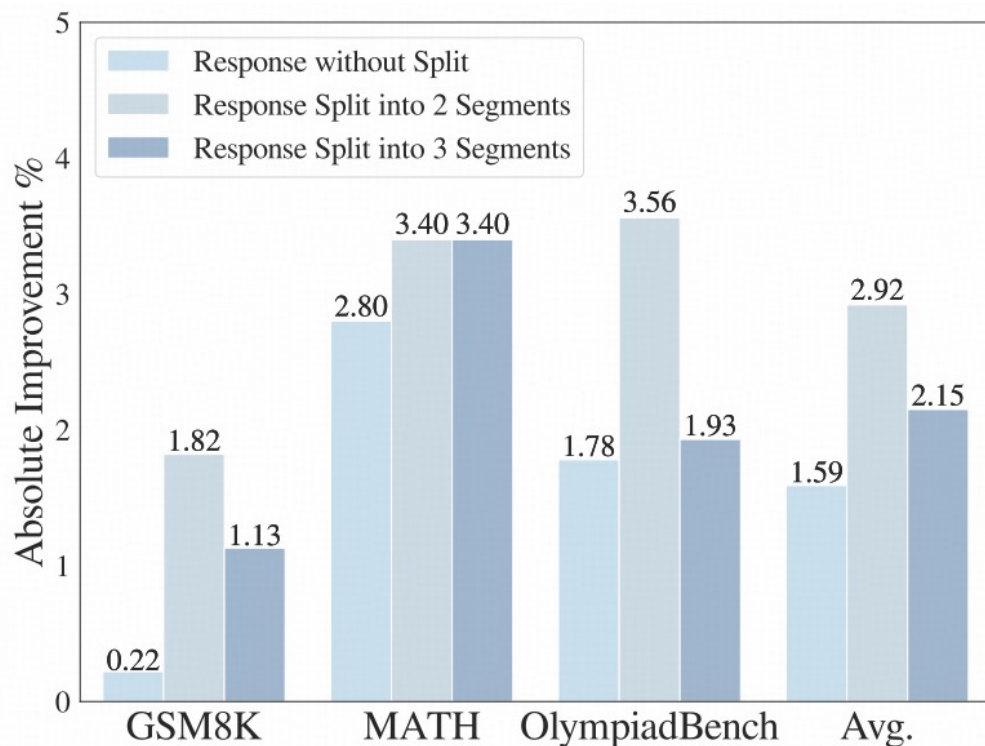


Figure 8: Comparison of accuracy improvement with splitting strategies over baseline (KD).

- Response-level pruning (both 2-segment and 3-segment splits) significantly outperforms instruct-level pruning.
- The performance of 3-segment splits is comparable to that of 2-segment splits, suggesting that further segmentation at the response level yields diminishing returns.

Analysis

Threshold Sensitivity Analysis

- For both instruction-level and response-level, LLaMa3.2-LeaF-1B achieves optimal performance at a higher misleading token threshold than LLaMa3.2-LeaF-3B.

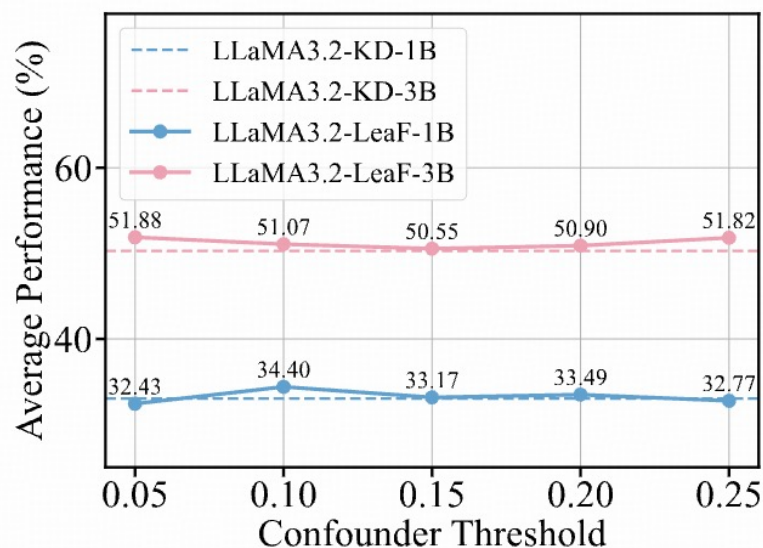


Figure 9: Instruct-level in MathBench.

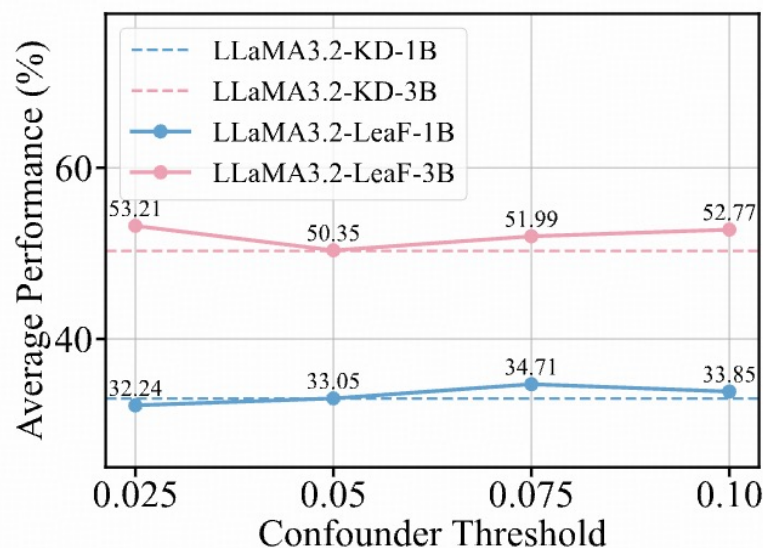


Figure 10: Response-level in MathBench.

Case Study

Two fair , 6 -sided dice are thrown . What is the probability that the product of the two numbers is a multiple of 5 ? Express your answer as a common fraction .

LlaMa3.2-3b-Instruct (Distilled by KD)

Step 1: When two dice are thrown, each die has 6 possible outcomes. Therefore, the total number of outcomes for two dice is $6 * 6 = 36$.

Step 2: To have a product that is a multiple of 5, at least one of the dice must roll a 5, or one die must roll a 4 and the other

.....

or one die must roll a 6 and the other a 4,
or one die must roll a 6 and the other a 3,
or one die must roll a 4 and the other a 6,

.....

Step 3: Count the favorable outcomes. Let's count the favorable outcomes based on the conditions identified:

Rolling a 5 on one die: $6 * 1 = 6$

(since the other die can be any number)

Rolling a 4 and a 5: $1 * 1 = 1$ (since the other die must be a 5)

Rolling a 4 and a 2: $1 * 1 = 1$ (since the other die must be a 2)

.....

Rolling a 3 and a 6: $1 * 1 = 1$ (since the other die must be a 6)

Rolling a 2 and a 6: $1 * 1 = 1$ (since the other die must be a 6)

.....

The final answer is:

$$\frac{2}{3}$$



LlaMa3.2-3b-Instruct (Distilled by LeaF)

Step 1: Determine the total possible outcomes when two dice are thrown. When two dice are thrown, each die can land on one of six faces, so the total number of outcomes is $6 * 6 = 36$.

Step 2: Identify the combinations that result in a product that is a multiple of 5. To have a product that is a multiple of 5, at least one of the dice **must land on a 5**. The combinations that result in **a product that is a multiple of 5 are: (1,5), (2,5), (3,5), (4,5), (5,1), (5,2), (5,3), (5,4), (5,5), (5,6), (6,5)**.

Step 3: Count the number of favorable outcomes.

From Step 2, we identified **11 combinations that result in a product that is a multiple of 5**.

Step 4: Calculate the probability of getting a product that is a multiple of 5.

The probability is calculated by dividing the number of favorable outcomes by the total number of possible outcomes. So, the probability = $11 / 36$.

The final answer is:

$$\frac{11}{36}$$



LeaF allows the model to focus more on the critical information, such as “multiple”, “5”.



Summary

- **LeaF = a simple, effective and efficient causal attention method**
 - Robust performance across math and code benchmarks.
 - The interpretability case study shows that LeaF allows the model to focus more on the critical information.
- **Limitations:**
 - Dependence on an **advanced teacher model**.
 - Limited scalability to long-text tasks.
 - Limited by the fixed threshold.

A decorative border consisting of a repeating pattern of stylized flowers and leaves, rendered in a light red or pink line-art style, framing the central text.

Thank You!

Yiju Guo

Email: yijuguo@ruc.edu.cn