

731 A Appendix

732 A.1 Threshold sensitivity analysis

733 We perform a sensitivity analysis on the threshold used for misleading token masking, assessing the
 734 performance of LLaMA3.2-1B-Instruct and LLaMA3.2-3B-Instruct as student models. As illustrated
 735 in the Figure 9 and Figure 10, we present the average experimental results of these models across
 736 MathBench (GSM8K, MATH, OlympiadBench) at different threshold levels.

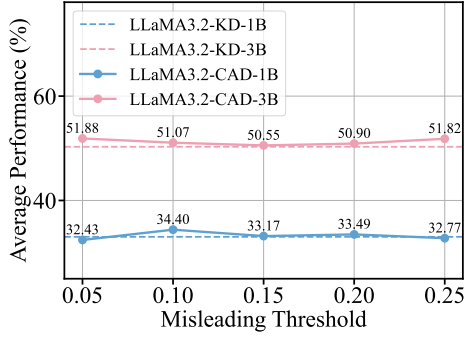


Figure 9: Instruct-level in MathBench.

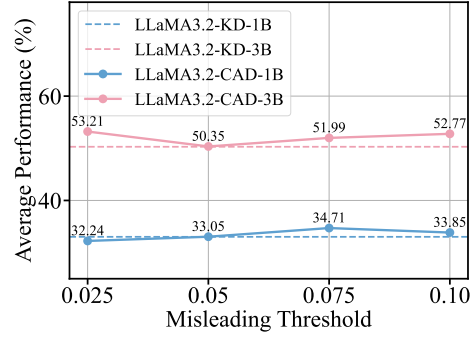


Figure 10: Response-level in MathBench.

737 **Results** We observe the following results: (1) Instruct-level: LLaMa3.2-CAD-1B performs best at
 738 a threshold of 0.10, while LLaMa3.2-CAD-3B performs best at 0.05. (2) Response-level: LLaMa3.2-
 739 CAD-1B performs best at 0.15, and LLaMa3.2-CAD-3B at 0.10. (3) For both instruction-level and
 740 response-level, LLaMa3.2-CAD-1B achieves optimal performance at a higher misleading token
 741 threshold than LLaMa3.2-CAD-3B. This suggests that smaller models, which are more susceptible to
 misleading tokens, benefit from higher thresholds that filter out disruptive tokens more effectively.

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

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 ✅

Figure 11: A case study comparing the performance of the student model (LLaMa3.2-1b-Instruct) and the teacher model (LLaMa3.3-70b-Instruct) on the MATH task.

A.2 Preliminary experiment

A.2.1 Gradient heatmap comparison

We compare the gradient heatmaps of the small model and the big model to analyze the differences in their attention and focus on critical tokens during the inference process.

Results Figure 11 illustrates that Llama3.3-70B-Instruct successfully captures a key contextual relation: "Five containers of blueberries can be traded for two zucchinis." Gradient heatmaps show that the teacher model aligns its attention closely with the relevant tokens, while the student model's attention is more dispersed. This observation motivates our hypothesis: **by pruning misleading patterns, we can guide the student model to better focus on salient information, enhancing its reasoning capabilities.** To evaluate this, we conduct pilot studies in two settings: (1) assessing accuracy gains on math and code benchmarks after pruning misleading patterns (Appendix A.2.2) and (2) evaluating improvements in response quality (Appendix A.2.3).

A.2.2 Performance gains from token pruning

For the LLaMA series, we use LLaMA3.2-1B/3B-Instruct as the student models and LLaMA3.3-70B-Instruct as the teacher model. For the Qwen series, we use Qwen2.5-Math-1.5B as the student model and Qwen2.5-72B-Instruct as the teacher model for preliminary experiments.

For mathematical problem solving, we select a filtered subset from the Numina-CoT dataset [22], where the teacher models generate correct inferences, consisting of 12K examples (3K each from GSM8K, Olympiads, AMC_AIME, and MATH). We then select the subset where the student models produce incorrect results. We compute each sample's gradient difference between the student and teacher models to identify potential misleading patterns. Finally, these misleading patterns are removed, and the student models are re-evaluated to assess the resulting improvement in accuracy. The results are presented in Figure 1(b).

For code generation, we construct a filtered subset of 12K examples from the AceCode dataset [44], where teacher models generate correct inferences. This subset comprises 6K samples from the Evol subset and 6K from the Oss subset. We apply the same misleading pattern identification and masking procedure as used in the mathematical domain. Subsequently, we re-evaluate student models to assess the impact of this intervention on code generation accuracy. The results are presented in Figure 1(a).

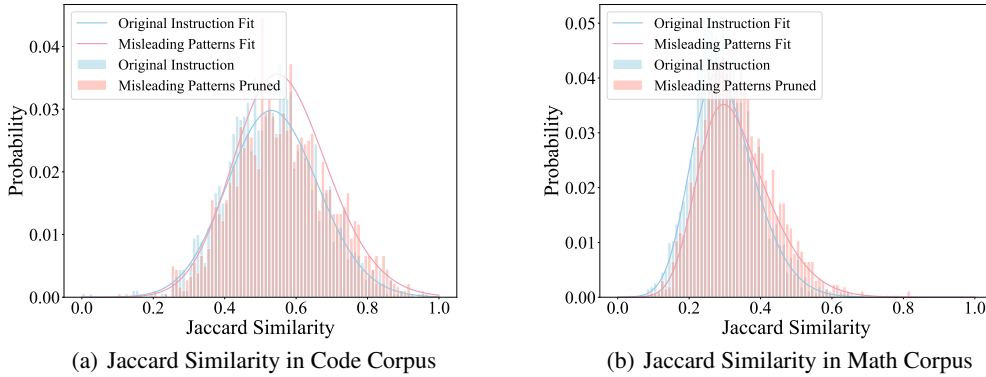


Figure 12: Jaccard similarity distribution between student model responses (original vs. instruction pruned misleading patterns) and ground-truth responses on math and code datasets.

A.2.3 Generation quality improvements from token pruning

We analyze the alignment between the student model's outputs and the teacher model's outputs under two conditions: (1) **Original Instruction**: The student model generates output based on the original instruction. (2) **Instruction pruned Misleading Patterns**: The student model generates output from an instruction where misleading tokens are masked. To quantify the similarity between the model

776 outputs, we use two widely adopted metrics: Jaccard Similarity [45]. This metric enables a practical
 777 evaluation of how well the student model’s output aligns with the contextual meaning conveyed by
 778 the teacher model.

779 **Results** Figure 12 show a shift in Jaccard Similarity distribution for responses generated by the
 780 student model on both code and math tasks after removing misleading tokens. This indicates that, by
 781 ignoring misleading patterns, the student model not only improves reasoning accuracy (Figure 1) but
 782 also generates responses more aligned with the teacher model, thereby enhancing output quality.

783 A.3 Collective Masking vs. Span Masking

784 We compare two masking strategies (Figure 13): **Collective Masking**, which simultaneously masks
 all identified misleading tokens, and **Span Masking**, which masks only contiguous spans of tokens.

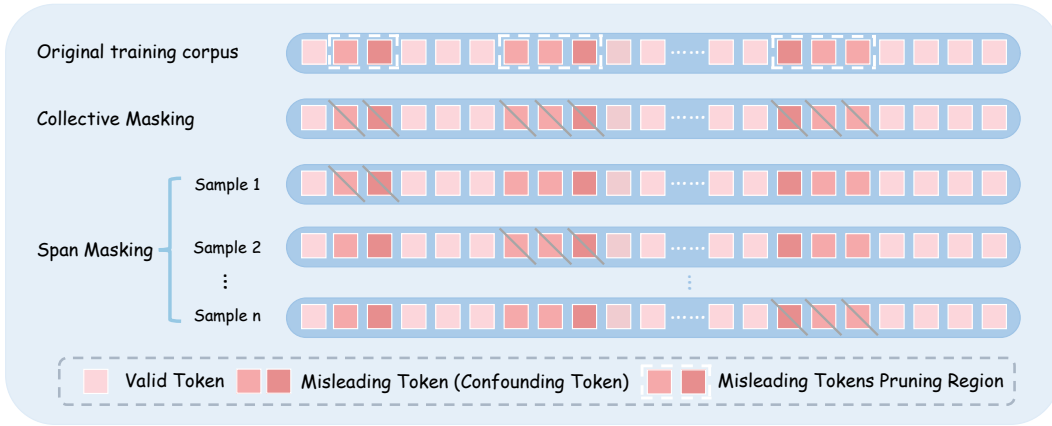


Figure 13: Illustration of Collective Masking and Span Masking.

785

786 **Results** Table 2 presents results for LLaMA3.2-1B/3B-Instruct models trained on Math corpus
 787 (86K) and evaluated on MATH-500 under two masking schemes. (1) We find that span masking
 788 substantially outperforms both collective masking and native knowledge distillation. (2) Collective
 789 Masking not only fails to yield improvements but even degrades performance on the LLaMA3.2-3B-
 790 Instruct model. We suspect that masking all misleading tokens at once disrupts the training data’s
 791 semantic coherence, undermining the model’s learning. Consequently, we adopt the **Span Masking**
strategy for all subsequent experiments.

Table 2: Comparison of two masking strategies: Collective Masking vs. Span Masking on Math-500 for LLaMA3.2-1B/3B-Instruct. The best and second-best results are marked in **bold** and underlined, respectively.

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

792

793 A.4 Detailed Training Settings

The complete training hyper-parameters in knowledge distillation are put in Table 3.

Table 3: Training hyper-parameters in Knowledge Distillation.

Model	Hyper-parameter	Value
LLaMA3.2-1B-Instruct	LR	1×10^{-5}
	LR Scheduler	cosine
	Batch Size	64
	Epochs	3
	Maximum Sequence Length	4096
	Warmup Steps	5
	Distill Loss Type	KL
	Validation Set Size (Math)	1035
	Validation Set Size (Code)	2000
LLaMA3.2-3B-Instruct	LR	1×10^{-5}
	LR Scheduler	cosine
	Batch Size	64
	Epochs	3
	Maximum Sequence Length	3000
	Warmup Steps	5
	Distill Loss Type	KL
	Validation Set Size (Math)	1035
	Validation Set Size (Code)	2000
Qwen2.5-Math-1.5B	LR	1×10^{-5}
	LR Scheduler	cosine
	Batch Size	32
	Epochs	3
	Maximum Sequence Length	4096
	Warmup Steps	5
	Distill Loss Type	KL
	Validation Set Size (Math)	1200
	Validation Set Size (Code)	2000

794

795 A.5 Detailed evaluation settings

796 For mathematical problem solving, we evaluate on GSM8K, MATH, and OlympiadBench using the
797 Step-DPO framework [20], with modifications to address its data extraction inaccuracies. For code
798 generation, we report pass@1 on HumanEval(+) [4] and LeetCode [7], using EvalPlus [28], and
799 pass@10 on LiveCodeBench [18], using the Skythought-Evals framework[21].

800 A.6 Evaluation benchmarks

801 GSM8K [6] comprises 8500 grade-school-level word problems, each requiring 2–8 steps of basic
802 arithmetic. Its natural-language diversity and multi-step structure make it a standard measure for
803 chain-of-thought prompting.

804 MATH [14] contains 12500 competition-style problems grouped into seven topics (Prealgebra,
805 Algebra, Number Theory, Counting & Probability, Geometry, Intermediate Algebra, Precalculus).
806 Every question is accompanied by a detailed solution.

807 OlympiadBench [13] was originally a bilingual, multimodal collection of 8476 Olympiad-level math
808 and physics problems. We filter out proof-based and image-based questions to obtain 674 pure-text
809 tasks, enabling focused evaluation of advanced symbolic reasoning.

810 HumanEval+ [28] extends the original HumanEval with additional Python programming tasks and
811 augmented unit tests, targeting functional correctness across diverse code patterns.

812 LeetCode [7] samples real-world algorithmic challenges from the LeetCode platform—arrays, trees,
813 dynamic programming, etc.—to assess models’ ability to generate correct and efficient solutions.
814 LiveCodeBench [18] provides a large-scale suite of real-world coding tasks with comprehensive unit
815 tests and human preference annotations, allowing evaluation of both functional accuracy and coding
816 style.

817 A.7 Open-source instruct models

818 The download links for the four open-source models are provided below:

- 819 • Llama-3.2-1B-Instruct: <https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct>
- 820 • Llama-3.2-3B-Instruct: <https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct>
- 821 • Llama-3.3-70B-Instruct: <https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct>
- 822 • Qwen2.5-Math-1.5B: <https://huggingface.co/Qwen/Qwen2.5-Math-1.5B>
- 823 • Qwen2.5-72B-Instruct: <https://huggingface.co/Qwen/Qwen2.5-72B-Instruct>

824 A.8 Limitations

825 Our work has the following limitations: (1) **Dependence on an advanced teacher model:** Our
826 method relies on a teacher model to identify misleading tokens. Exploring self-improvement that
827 enables models to refine their attention to critical tokens and boost reasoning can be an interesting and
828 important future work. (2) **Limited scalability to long-text tasks:** Due to the inherent limitations
829 of the student model, we have only validated our approach on math and code tasks, leaving its
830 applicability to long-text and other domains for future investigation.