ReBound: Relevance-Boundary Supervision for Distractor-Robust Mathematical Reasoning

Anonymous ACL submission

Abstract

Mathematical word problems with plausible irrelevant context expose a failure mode in which language models reason from facts that should not support the target solution. We formulate distractor-robust mathematical reasoning as *relevance-boundary supervision*: a model should separate Signal facts from Distractor facts before solving, and then ground the solution in Signal facts only. We introduce ReBound, a post-training framework that constructs paired clean–distractor examples, trains boundary-first traces with supervised fine-tuning, and applies correctness-guided reinforcement learning to discourage Distractor use. Across three open base LLMs and six mathematical reasoning benchmarks, ReBound consistently improves distractor robustness over CoT prompting and strong training-based baselines; on Qwen3-0.6B, the distractor benchmark average increases from 34.92 to 48.29. ReBound also preserves competitive clean and perturbation performance. Process diagnostics, ablations, and output-length analysis show that these gains are tied to better fact use and boundary behavior, rather than longer generations or answer-only optimization. The code is available at `https://anonymous.4open.science/r/Rebound-5FB8/`

1 Introduction

Large language models (LLMs) have made rapid progress on mathematical reasoning (Yu et al., 2024a; Laskar et al., 2024; Cheng et al., 2025). In clean mathematical reasoning benchmarks, many models now achieve high accuracy; however, these settings often assume that most information in the problem statement is relevant to the target question (Zhang et al., 2024; Cobbe et al., 2021; Shi et al., 2023). Realistic inputs are rarely this clean: they may include contextual facts that are plausible yet irrelevant to the current solution. In such cases, failures may arise not from limited calculation ability, but from treating irrelevant information as useful evidence (Wang et al., 2026). Figure 1 illustrates this failure mode.

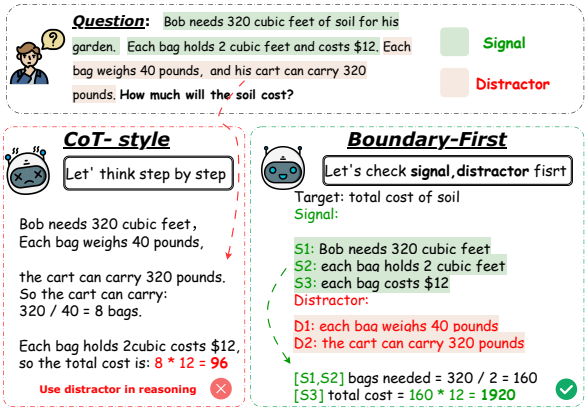


Figure 1: A motivating example of distractor-induced reasoning failure. A standard chain-of-thought response may carry a plausible but irrelevant fact into its reasoning and calculation; ReBound avoids this failure by first identifying which facts are relevant to the target question and which are irrelevant.

The failure mode illustrated above is not an isolated case. Prior benchmarks show that LLMs, especially smaller models, are sensitive to irrelevant context, irrelevant clauses, and perturbations (Shi et al., 2023; Anantheswaran et al., 2025; Yang et al., 2025c; Li et al., 2024; Mirzadeh et al., 2025; Zhou et al., 2024b). These findings point to a bottleneck that arises before calculation: models may fail to distinguish facts that are relevant to the target question from facts that should be ignored, and may therefore solve the problem using the wrong information (Creswell et al., 2022).

Existing methods have attempted to improve robustness to irrelevant context from several directions. Inference-time filtering methods prompt LLMs to select, rewrite, or ignore parts of the context before answering (Creswell et al., 2022; Weston and Sukhbaatar, 2023), but they depend on reliable instruction following and may degrade for smaller models or longer reasoning chains. Data augmentation and adversarial training expose models to examples with distractors (Fang et al., 2024; Yoran et al., 2023; Yu et al., 2024b; Yue et al.,

2024), but their supervision often remains answer-level, so intermediate reasoning can still contain errors (Zhou et al., 2024a). Structured reasoning and process-supervised methods make intermediate steps explicit (Wang et al., 2024; Gao et al., 2026; Setlur et al., 2025; Chen et al., 2022), but explicit reasoning alone does not ensure appropriate fact use. Robustness to irrelevant context therefore requires an additional constraint: models should identify the relevant facts before solving and reason only from them.

To address this constraint, we propose ReBound, a learning framework that formulates distractor-robust mathematical reasoning as *relevance-boundary supervision*. Its central representation is a *boundary-first reasoning trace*: before solving, the model draws a relevance boundary by separating facts that are relevant to the target question from plausible distractors, and then reasons only from the former. To train this behavior, ReBound constructs paired clean and distractor-containing examples with boundary-first traces as target outputs, uses supervised fine-tuning to teach the trace format, and applies reinforcement learning with rule-based rewards to prefer answer-correct traces that avoid irrelevant facts. This explicitly trains the model to identify which facts should be used before solving, instead of only adding noisy training examples or encouraging longer rationales.

We evaluate ReBound on three open base LLMs across clean, perturbed, and distractor-focused mathematical reasoning benchmarks. On the distractor benchmark average, ReBound improves over the strongest non-ReBound baseline by 8.5 points on Qwen3-0.6B and 4.6 points on Qwen3-4B, while preserving competitive performance on clean and perturbed inputs. Process diagnostics, ablations, and output-length analyses further indicate that ReBound more reliably separates relevant from irrelevant facts and incorporates fewer distractor facts into the reasoning traces; these gains are not attributable merely to longer generations or answer-only optimization.

Our contributions are threefold:

- We formulate distractor-robust mathematical reasoning as relevance-boundary supervision and develop a paired data construction pipeline that produces clean and distractor-containing examples with boundary-first traces.
- We propose ReBound, a post-training framework that combines boundary-first supervised fine-tuning with rule-based reward-guided optimization to train models to identify relevant facts before reasoning.
- We provide extensive experiments across three backbone models and six benchmarks, including diagnostics and ablations, showing that ReBound improves distractor robustness while preserving general mathematical reasoning ability.

2 Preliminary

2.1 Problem Formulation: Relevance Boundary

For a mathematical word problem x, let τ denote the target question and y the gold answer. We view the problem statement as a set of facts $\mathcal{F}(x)$, where each fact is an atomic unit such as a quantity with its entity and condition, or a numerical relation used as one unit in reasoning. This fact-level view follows prior work on selecting context facts for reasoning (Creswell et al., 2022; Yang et al., 2025c).

Given τ and a reference solution path $\Pi^\star$, *Signal* facts $\mathcal{S} \subseteq \mathcal{F}(x)$ are the facts used to solve the target question, while *Distractor* facts $\mathcal{D} \subseteq \mathcal{F}(x)$ are contextually plausible facts outside that path. We assume $\mathcal{S} \cap \mathcal{D} = \emptyset$; a distractor-free instance is the special case where $\mathcal{D} = \emptyset$.

The separation between $\mathcal{S}$ and $\mathcal{D}$ defines the relevance boundary for τ and $\Pi^\star$. Distractor-robust reasoning requires producing y using $\mathcal{S}$ rather than $\mathcal{D}$. The boundary is target- and path-relative: a fact may be useful for another question or solution path, but should remain outside the current one if it is not used along $\Pi^\star$. Thus, distractor robustness becomes a fact-use supervision problem: the model must learn which facts belong inside the current solution path and which should remain outside it.

2.2 Related Work

Distractor Robustness in Mathematical Reasoning. Robustness benchmarks have shown that model accuracy on math problems can drop under input perturbations or added distractors (Zhang et al., 2024). GSM-Plus, GSM-Symbolic, and MathAttack evaluate numerical, symbolic, and adversarial variations (Li et al., 2024; Mirzadeh et al., 2025; Zhou et al., 2024b); GSM-IC, GSM-DC, and ProblemMathIC focus more directly on irrelevant or off-path facts in math word problems (Shi et al., 2023; Yang et al., 2025c; Anantheswaran et al., 2025). Together, these benchmarks motivate the central concern of ReBound: distractor robustness depends not only on whether the final answer is

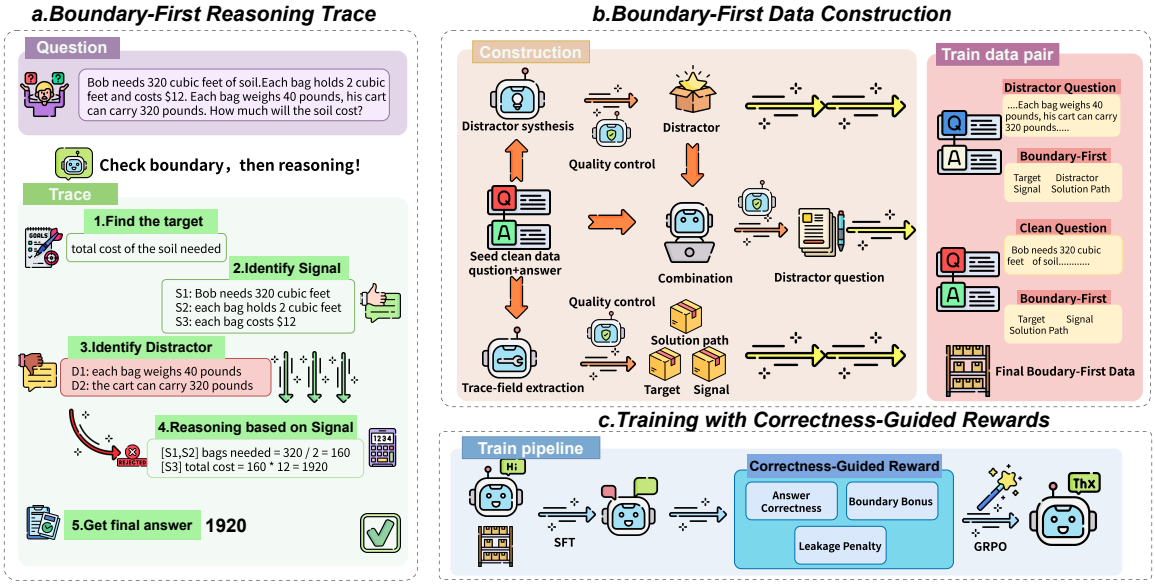


Figure 2: Overview of ReBound. ReBound constructs paired clean and distractor records, formats them as boundary-first traces, and trains the model with supervised fine-tuning followed by correctness-guided GRPO. The trace first separates Signal facts $\mathcal{S}$ from Distractor facts $\mathcal{D}$, and then solves the problem using only $\mathcal{S}$.

correct, but also on whether the model avoids using off-path facts in the solution.

Filtering and Training with Irrelevant Context. Methods for handling irrelevant context typically operate either at inference time or through training. At inference time, filtering methods prompt models to select evidence, rewrite inputs, or ignore irrelevant context before answering (Creswell et al., 2022; Weston and Sukhbaatar, 2023; Chen et al., 2022), and related selection mechanisms have been explored in retrieval-augmented generation (Yu et al., 2024c; Wei et al., 2025). Training-based approaches instead expose models to distractor-containing, adversarial, or synthetic reasoning data to improve robustness (Fang et al., 2024; Yoran et al., 2023; Yu et al., 2024b; Yue et al., 2024; Zhou et al., 2024b). Their supervision, however, is usually applied through prompts or final answers, leaving the link between the relevance decision and later derivation implicit. ReBound differs by serializing this decision in the target trace and by training the derivation to stay within the declared Signal facts.

Structured and Process-Supervised Reasoning. Another line of work improves reasoning by making intermediate steps more explicit and easier to supervise. Structured reasoning methods range from chain-of-thought prompting to abstraction-based formats such as AoT and AbstRaL (Wei et al., 2022; Hong et al., 2024; Gao et al., 2026, 2025; Dutta et al., 2024; Qu et al., 2025). Process supervision and reasoning post-training provide feedback through step verification, process reward models, and GRPO-style optimization (Lightman et al., 2024; Wang et al., 2024; Setlur et al., 2025; Zhang and Zuo, 2025). These methods improve the representation, checking, or optimization of reasoning trajectories. ReBound addresses a more specific constraint for distractor-robust reasoning: a step should be valid not only because it is well formed or locally correct, but also because it uses only facts admitted by the pre-declared relevance boundary.

3 Methodology

3.1 Overview

Figure 2 gives an overview of ReBound. ReBound trains the model to establish a relevance boundary before solving: Signal facts $\mathcal{S}$ are separated from Distractor facts $\mathcal{D}$, and the solution then uses only $\mathcal{S}$. This behavior is supported by boundary-first traces that make the boundary explicit (Section 3.2), paired clean and distractor records that teach both retaining needed facts and excluding irrelevant ones (Section 3.3), and correctness-guided GRPO that favors traces that reach the correct answer while respecting the boundary (Section 3.4).

3.2 Boundary-First Reasoning Traces

In standard chain-of-thought rationales, Signal fact selection and calculation are often mixed in a single free-form sequence. This makes it difficult to verify whether later steps rely only on Signal facts: a model may place a Distractor fact inside $\mathcal{S}$, or label it as irrelevant but still use it later. ReBound instead

uses a *boundary-first trace* that makes the intended order of reasoning explicit.

As illustrated in Figure 2, a boundary-first trace follows five stages. The model first identifies the target of the question, so the subsequent relevance decision is tied to the quantity being solved. It then separates the stated facts into Signal facts $\mathcal{S}$, which may support the solution, and Distractor facts $\mathcal{D}$, which should remain outside the solution path. Only after this boundary is stated does the model produce the solution path, where each step should be based on Signal facts, and finally output the answer. Thus, the trace turns "what to solve" and "which facts may be used" into explicit parts of the reasoning process before calculation begins.

We write this boundary-first trace as

$$z = (\tau, \mathcal{S}, \mathcal{D}, \Pi, a), \tag{1}$$

where τ specifies the target question, $\mathcal{S}$ and $\mathcal{D}$ store the Signal and Distractor facts, Π denotes the solution path, and a is the final answer. In the text form used for training and generation, facts are assigned identifiers such as $S_1, S_2, \ldots$ and $D_1, D_2, \ldots$, allowing each step in Π to explicitly refer to the facts it uses.

These identifiers make the relevance boundary checkable after it is drawn. Let $\pi_t \in \Pi$ denote a derivation step, and let R_t denote the set of facts referenced by that step. ReBound requires $R_t \subseteq \mathcal{S}$ and $R_t \cap \mathcal{D} = \emptyset$, so any fact placed in $\mathcal{D}$ cannot later support the solution path. The training reward further penalizes cases where a step does not cite a Distractor fact but still uses a value or relation unique to it.

The same trace format is used for both clean and distractor-containing inputs. For clean inputs, $\mathcal{D} = \emptyset$, and the model should solve from $\mathcal{S}$ without inventing irrelevant exclusions. This shared format allows the next-stage paired supervision to cover both cases: excluding Distractor facts when they are present, and retaining needed Signal facts when no distractor should be removed.

3.3 Paired Data Construction for Boundary Supervision

Motivation. Boundary-first traces require supervision for both sides of the relevance decision. Training only on distractor-containing inputs can teach exclusion, but may also encourage the model to reject facts even when no distractor is present. We therefore build paired records from clean math word problems: a clean record with $\mathcal{D} = \emptyset$, and a distractor record that adds plausible but irrelevant facts while preserving the target question and gold answer.

Paired Records. Let (x_i, y_i) be a clean seed problem and its gold answer. We create a clean trace $z_i^{\text{clean}} = (\tau_i, \mathcal{S}_i, \emptyset, \Pi_i, y_i)$ and a distractor trace $z_i^{\text{distr}} = (\tau_i, \mathcal{S}_i, \mathcal{D}_i, \Pi_i, y_i)$. The distractor trace is paired with an input $\tilde{x}_i$, obtained by adding $\mathcal{D}_i$ to x_i. Thus, the two records share the same target, answer, Signal facts, and solution path, but differ in whether the relevance boundary contains Distractor facts.

Data Construction. Starting from a clean seed problem with its reference solution and answer, we use an LLM proposal model to construct the boundary-first trace fields. The model first extracts the target τ_i, Signal facts $\mathcal{S}_i$, and reference solution path Π_i from the clean instance. It then creates a distractor-containing version by selecting a distractor pattern and inserting a contextually plausible fact that is irrelevant to Π_i. In our constructed pairs, this inserted fact forms the distractor set $\mathcal{D}_i$. The distractor trace keeps the same τ_i, y_i, $\mathcal{S}_i$, and Π_i, so the inserted fact changes the relevance boundary but not the solution path. The pattern list and prompts are provided in Appendix A.

Quality Control. Since both the trace fields and inserted facts are proposed by the LLM, each pair is validated before training. We keep a pair only when the clean trace matches the seed question and reaches the gold answer using $\mathcal{S}_i$; the inserted fact preserves τ_i and y_i while staying outside the reference solution path; and the distractor trace assigns the inserted fact to $\mathcal{D}_i$ with all derivation steps grounded in Signal facts. Pairs failing any check are regenerated or discarded. The remaining pairs provide the boundary-supervised data for SFT and reward computation.

3.4 Training with Correctness-Guided Rewards

ReBound training proceeds in two stages. Supervised fine-tuning teaches the boundary-first trace format from paired records. GRPO then optimizes sampled traces with *correctness-guided rewards*: answer correctness is the primary signal, while traces that respect the relevance boundary receive additional credit and direct Distractor leakage is penalized.

Supervised Fine-Tuning. We fine-tune the model with the standard next-token objective, mapping each clean input x_i to z_i^{clean} and each distractor input $\tilde{x}_i$ to z_i^{distr}. The resulting model initializes

Method	Clean	Perturbation		Distractor			Summary
	GSM8K	GSM-Sym.	GSM-Plus	GSM-IC	GSM-DC	ProbMathIC	Avg.
Qwen3-0.6B							
CoT-Base	40.33	34.38	24.25	57.93	21.60	25.22	34.92
Distractor CoT-SFT	42.15	38.64	25.62	61.11	13.17	27.84	34.04
AoT	47.08	47.00	30.96	64.45	23.27	31.50	39.74
CoT-RL	58.15	**53.32**	38.21	59.39	18.68	29.84	35.97
AbstRaL-RL	54.21	51.26	33.71	52.60	22.41	28.39	34.47
ReBound	**61.49**	52.86	**39.62**	**77.78**	**23.54**	**43.56**	**48.29**
Qwen3-4B							
CoT-Base	**89.46**	84.66	69.79	95.68	22.90	65.75	61.44
Distractor CoT-SFT	76.72	71.64	56.42	93.44	31.63	64.02	63.03
AoT	82.79	76.60	60.83	94.37	32.35	66.71	64.48
CoT-RL	84.76	80.88	65.08	92.19	31.89	70.50	64.86
AbstRaL-RL	87.34	**85.86**	64.21	86.95	31.30	70.02	62.76
ReBound	88.48	83.66	**70.04**	**97.25**	**32.98**	**78.08**	**69.44**
Ministral-3B							
CoT-Base	81.80	77.34	**62.58**	89.50	25.54	63.34	59.46
Distractor CoT-SFT	74.75	65.74	54.04	90.67	29.40	59.27	59.78
AoT	77.33	72.28	57.17	83.91	25.06	61.34	56.77
CoT-RL	76.50	68.64	54.75	83.73	27.38	54.31	55.14
AbstRaL-RL	80.74	77.54	59.26	82.96	**30.48**	62.51	58.65
ReBound	**82.49**	74.06	61.75	**93.57**	29.42	**64.44**	**62.48**

Table 1: Answer accuracy (%) on clean, perturbation, and distractor-robust math reasoning benchmarks. Bold/underline mark best/second-best within each model block; Avg. is over GSM-IC, GSM-DC, and ProblemMathIC.

GRPO, which uses the same paired inputs and gold trace fields for reward computation.

Reward-Guided Optimization. SFT alone does not ensure that sampled traces obey the relevance boundary: a model may place facts on the wrong side, use a Distractor fact in the solution path, or produce a well-formed but incorrect solution. We therefore parse the generated target, Signal, Distractor, solution path, and final-answer fields, and compute rule-based rewards without training a separate reward model.

For a sampled output o, let $c(o) = I[ans(o) = y]$ indicate answer correctness. At a high level, the reward can be viewed as the following decomposition:

$$\text{Reward}(o) = \lambda_{\text{ans}} c(o) + B(o) - \lambda_{\text{leak}} L(o), \quad (2)$$

where $B(o)$ rewards correct Signal–Distractor placement, empty $\mathcal{D}$ for clean inputs, and derivation steps that reference only Signal facts; $L(o)$ penalizes leakage, including placing an inserted Distractor fact in $\mathcal{S}$, referencing it in Π, or using a value unique to it. In implementation, this decomposition is instantiated with terms for format validity, boundary placement, Signal-only references, parsed computations, final-answer consistency, and bad-case penalties. Malformed, unparsable, or answer-incorrect traces receive capped process credit; details are given in Appendix B.

4 Experiments

We evaluate ReBound by asking four questions. **RQ1:** Does ReBound improve mathematical reasoning under distracting context while preserving performance on clean and perturbed-but-non-distractor benchmarks? **RQ2:** Does ReBound improve relevance-boundary behavior, including Signal–Distractor separation and reduced Distractor use in derivations? **RQ3:** Which components of ReBound, including paired boundary supervision and correctness-guided rewards, are responsible for the gains? **RQ4:** Are the gains explained by longer generations, or do they reflect better use of relevant facts?

4.1 Experimental Setup

Models. We evaluate ReBound on Qwen3-0.6B, Qwen3-4B (Yang et al., 2025a), and Ministral-3B-Base-2512 (Liu et al., 2026). For data construction, we use Qwen2.5-72B-Instruct (Yang et al., 2025b) as the LLM proposal model for trace extraction and distractor synthesis; it is not used as an evaluated student model.

Benchmarks. We evaluate on three benchmark groups: clean reasoning with GSM8K (Cobbe et al., 2021), perturbation robustness with GSM-Symbolic and GSM-Plus (Mirzadeh et al., 2025; Li et al., 2024), and distractor-robust reasoning with GSM-IC, GSM-DC, and ProblemMathIC (Shi et al., 2023; Yang et al., 2025c; Anantheswaran et al., 2025). Benchmark details are provided in Appendix D.

Baselines. We compare ReBound with CoT-Base, Distractor CoT-SFT, CoT-RL (Shao et al., 2024), AoT (Hong et al., 2024), and AbstRaL-RL (Gao

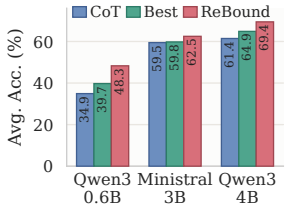
et al., 2026). These baselines cover standard prompting, distractor-data SFT, answer-oriented RL, and structured or abstraction-based reasoning without explicit Signal–Distractor supervision.

4.2 Main Results on Distractor-Robust Reasoning

Table 1 reports answer accuracy across clean, perturbation, and distractor benchmarks. Figure 3 provides two complementary views of the main distractor results: panel (a) summarizes the distractor average for each backbone, and panel (b) decomposes the ReBound gain over the strongest non-ReBound baseline by benchmark.

The main pattern is consistent across model scales: ReBound achieves the best distractor average for every backbone and is strongest when irrelevant facts directly compete with the target solution. The average gain is largest for Qwen3-0.6B, suggesting that explicit relevance-boundary supervision is especially helpful when the base model is more vulnerable to distracting context. Panel (b) shows that the improvement is not uniform across benchmarks: ReBound gains are concentrated on GSM-IC and ProblemMathIC, while GSM-DC shows smaller differences.

The GSM-DC results also clarify where ReBound is less dominant. On Ministral-3B, ReBound is slightly below AbstRaL-RL on GSM-DC. One plausible factor is transfer: the ReBound traces are synthesized with Qwen2.5-72B-Instruct



(a) Distractor average.

	GSM-IC	GSM-DC	Prob.
Qwen3 0.6B	+13.3	+0.3	+12.1
Ministral 3B	+2.9	-1.1	+1.1
Qwen3 4B	+1.6	+0.6	+7.6

(b) Per-benchmark gain.

Figure 3: Main distractor-robustness gains across backbones. (a) compares CoT-Base, the strongest non-ReBound baseline, and ReBound using the distractor benchmark average. (b) decomposes the ReBound gain over the strongest non-ReBound baseline on each distractor benchmark.

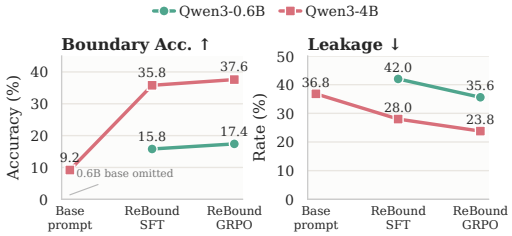


Figure 4: Boundary diagnostics across prompting, ReBound-SFT, and ReBound-GRPO.

as the teacher, which may align more naturally with Qwen-family students than with Ministral. Another factor is benchmark structure. GSM-DC is built from controlled dependency graphs and template-like distractors, rather than naturally written irrelevant context; abstraction-based methods that first list conditions and then derive equations can be well matched to this construction. This helps explain why GSM-DC is the least favorable distractor benchmark for ReBound, even though the overall distractor average remains strongest.

These gains do not come from simply giving up ordinary mathematical reasoning. ReBound remains competitive on clean and perturbation benchmarks, and in several columns matches or exceeds methods that do not explicitly supervise the Signal–Distractor boundary. The pattern suggests that ReBound mainly changes how models use facts under irrelevant context, rather than broadly trading clean accuracy for distractor robustness. We therefore examine whether higher answer accuracy is accompanied by better boundary behavior.

4.3 Boundary Diagnostics on ReBound-Test

Final-answer accuracy alone does not reveal whether a model solved a distractor problem using Signal rather than Distractor facts. We therefore evaluate process-level boundary behavior on **ReBound-Test**, a held-out set of distractor pairs constructed with the same validation criteria as the training data. Each example has a known inserted Distractor fact, but the model is not given the gold boundary at inference time. We report ANSWER ACC., DISTRACTOR ID, LEAKAGE, and BOUNDARY ACC.; the last metric requires a correct answer, correct Distractor identification, and no Distractor use in the derivation. The Base row uses the untrained backbone with a boundary-first prompt, and ANSWER ACC. is parsed from this boundary-first output.

Table 2 and Figure 4 show that a boundary-first prompt alone does not reliably induce the intended behavior. The Qwen3-4B base model can often answer correctly under the prompt, but it rarely

Method	Ans. ↑	ID ↑	Leak. ↓	Bound. ↑
Qwen3-0.6B				
Base w/ prompt	/	/	/	/
ReBound-SFT	40.8	29.8	42.0	15.8
ReBound-GRPO	**42.2**	**32.6**	**35.6**	**17.4**
Qwen3-4B				
Base w/ prompt	73.6	10.4	36.8	9.2
ReBound-SFT	75.6	44.2	28.0	35.8
ReBound-GRPO	**75.8**	**47.2**	**23.8**	**37.6**

Table 2: Boundary diagnostics on ReBound-Test. All numbers are percentages. Ans., ID, Leak., and Bound. denote answer accuracy, distractor identification, leakage, and boundary accuracy. A slash marks omitted process metrics for the Qwen3-0.6B base model, whose boundary-first prompt outputs are too unstable to parse reliably.

satisfies the stricter boundary criterion; for Qwen3-0.6B, prompt outputs are too unstable to parse reliably. Paired ReBound-SFT changes this behavior: the model begins to produce usable boundary-first traces, identifies more Distractor facts, and reduces leakage relative to prompting.

GRPO mainly improves the process metrics rather than raw answer accuracy. After SFT has made the trace format learnable, correctness-guided rewards further reduce Distractor use and improve boundary accuracy on both backbones. This supports the intended division of labor between the two stages: SFT teaches the model to express the relevance boundary, while GRPO refines answer-correct traces by discouraging Distractor facts from entering the solution path.

4.4 Ablation Studies

Table 3 ablates trace format, paired supervision, and rewards. ReBound-GRPO is separated as the common reference for paired-supervision and reward ablations; +GRPO rows continue from their corresponding SFT checkpoints, and the two ReBound-SFT rows use the same checkpoint.

Trace Design. This comparison keeps the distractor-containing inputs fixed and changes only the supervision format. Standard CoT rationales on the same data, shown as CoT-SFT w/o Boundary Trace, reach 34.76 average accuracy, compared with 45.71 for Full ReBound-SFT. Removing the Target or Distractor field keeps the boundary-first trace but weakens the relevance-boundary signal: the Target field fixes the question being solved, and the Distractor field names plausible facts to exclude from the solution path. Full ReBound-SFT gives the best average and leads on four of the six benchmarks.

Paired Supervision. Here the trace format is unchanged and only the data source varies. Clean-only variants retain clean and perturbation performance but drop sharply on GSM-IC and ProblemMathIC, indicating that clean supervision alone does not teach the model to exclude irrelevant context. Distractor-only variants reverse this trade-off: they improve the distractor columns, especially GSM-IC, while giving up clean and perturbation accuracy. Paired ReBound-SFT achieves the best average by exposing the model to both regimes: clean examples emphasize retaining Signal facts, and distractor examples emphasize excluding Distractor facts. The final ReBound-GRPO row shows this paired checkpoint after reward training.

Reward Design. Starting from ReBound-SFT, answer-only GRPO brings only a small average gain. Without the correctness cap, performance falls below the other GRPO variants on average and on the main distractor benchmarks, supporting the choice to gate process rewards by answer correctness. Removing the leakage penalty gives the highest answer accuracy within the reward ablations, so this term should not be read as an answer-accuracy objective. Its role is process-level: it discourages the use of Distractor facts, consistent with the reduced leakage observed in Section 4.3. Combining paired supervision with the full reward yields the best overall average and the strongest results on GSM8K, GSM-IC, and ProblemMathIC.

Together, these ablations show that no single ingredient accounts for the gains: boundary-first traces make SFT effective, paired data balances clean and distractor settings, and reward training improves the paired checkpoint while retaining pressure against Distractor use.

4.5 Output Length Analysis

Because ReBound uses an explicit boundary-first trace, a natural question is whether its gains mainly come from longer generations. We examine this by comparing answer accuracy with average output length on the distractor benchmarks, using generated outputs from the main evaluation. The analysis covers three backbones and four representative methods: CoT-Base, CoT-RL, AbstRaL-RL, and ReBound. GSM-IC aggregates IC-2 and IC-M. Further details are provided in Appendix E.

Table 4 shows that ReBound is not simply the most verbose method. Across all three backbones and distractor benchmarks, at least one baseline produces longer outputs than ReBound, and this is especially clear on GSM-DC and ProblemMathIC, where CoT-Base or AbstRaL-RL often generate

Method	Clean	Perturbation		Distractor			Summary
	GSM8K	GSM-Sym.	GSM+	GSM-IC	GSM-DC	Prob.	Avg.
Trace design							
CoT-Base	40.33	34.38	24.25	57.93	21.60	25.22	33.95
CoT-SFT w/o Boundary Trace	42.15	38.64	25.62	61.11	13.17	27.84	34.76
w/o Target Field	55.95	50.88	35.88	68.22	**23.90**	33.49	44.72
w/o Distractor Field	**56.25**	52.64	37.17	68.96	22.27	33.98	45.21
Full ReBound-SFT	55.65	**52.80**	**37.67**	**69.60**	23.65	**34.87**	**45.71**
Paired supervision							
Clean-only SFT	53.75	50.36	34.71	52.88	23.57	29.50	40.80
Clean-only SFT+GRPO	53.68	50.04	36.21	52.87	23.54	28.39	40.79
Distractor-only SFT	51.55	49.40	33.42	72.93	23.17	35.84	44.39
Distractor-only SFT+GRPO	52.16	48.92	33.08	**73.13**	**24.00**	**35.98**	44.55
Paired ReBound-SFT	**55.65**	**52.80**	**37.67**	69.60	23.65	34.87	**45.71**
Reward design							
ReBound-SFT	55.65	52.80	37.67	69.60	23.65	34.87	45.71
Answer-only GRPO	54.51	52.72	36.58	72.33	23.62	36.94	46.12
w/o Boundary Bonus	55.65	52.86	36.67	72.40	23.49	37.28	46.39
w/o Correctness Cap	56.18	52.18	37.08	68.83	23.38	35.08	45.46
w/o Leakage Penalty	**58.38**	**55.00**	**40.08**	**74.35**	**23.89**	**39.42**	**48.52**
ReBound-GRPO	**61.49**	**52.86**	**39.62**	**77.78**	**23.54**	**43.56**	**49.81**

Table 3: Ablation accuracy (%) on Qwen3-0.6B. GSM-IC merges IC-2/IC-M with 4K/2K weighting; Avg. is over all six benchmarks. Bold/underline mark best/second-best within each panel; Prob. denotes ProblemMathIC.

Model	Method	GSM-DC	Prob.	GSM-IC
Qwen3-0.6B	CoT-Base	512	489	463
Qwen3-0.6B	CoT-RL	915	378	191
Qwen3-0.6B	AbstRaL-RL	839	335	244
Qwen3-0.6B	ReBound	511	433	355
Qwen3-4B	CoT-Base	511	356	248
Qwen3-4B	CoT-RL	351	144	108
Qwen3-4B	AbstRaL-RL	682	255	201
Qwen3-4B	ReBound	624	196	160
Ministral	CoT-Base	758	316	231
Ministral	CoT-RL	460	194	126
Ministral	AbstRaL-RL	760	275	217
Ministral	ReBound	565	223	165

Table 4: Average generated tokens on distractor benchmarks. GSM-IC uses sample-size weighting over IC-2/IC-M; Prob. denotes ProblemMathIC.

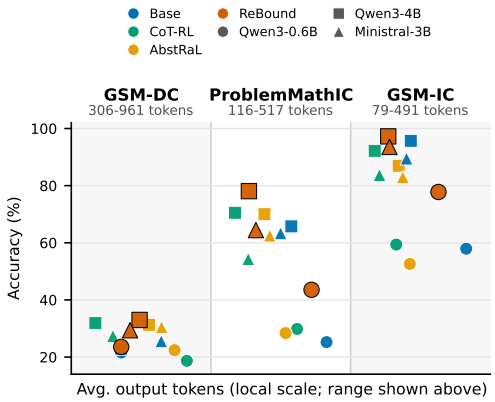


Figure 5: Output length and accuracy on distractor benchmarks. Each point is a method–backbone pair; bands show benchmark-specific token ranges.

more tokens without matching ReBound's robustness. The length pattern also varies substantially by backbone and benchmark, suggesting that output length is not a stable proxy for distractor handling.

Figure 5 pairs these token counts with answer accuracy and leads to the same conclusion. Higher accuracy does not consistently appear at the longest output lengths: several baselines generate comparable or longer responses without matching ReBound, while ReBound often reaches higher accuracy from moderate-length traces. Thus, the robustness gains are unlikely to be an artifact of producing longer chains of thought alone. Together with the boundary diagnostics in Section 4.3, this suggests that the gains are tied to better use of relevant facts rather than output length. A case study is provided in Appendix F.

5 Conclusion

We presented ReBound, a relevance-boundary supervision framework for distractor-robust mathematical reasoning. ReBound frames robustness to irrelevant context as controlled fact use: before solving, the model separates Signal facts from Distractor facts and derives the answer only from Signal facts. This makes evidence use an explicit target for training and evaluation, rather than leaving it implicit in free-form reasoning.

Across multiple models and benchmarks, ReBound improves distractor robustness while preserving general mathematical reasoning ability. Diagnostics and ablations show that these gains come from the combination of boundary-first traces, paired clean–distractor supervision, and correctness-guided rewards, rather than longer rationales or answer-only optimization. These results suggest that post-training can improve not only answer accuracy, but also whether derivations remain grounded in admissible facts.

Limitations

This work focuses on mathematical word problems with explicit quantities, entities, and conditions, where the relevance boundary is relatively observable. Tasks such as commonsense reasoning, open-domain question answering, code reasoning, or long-context analysis may involve facts that are only partially or conditionally relevant. Extending ReBound to these settings is important but outside the scope of this paper, whose goal is to study distractor robustness in controlled mathematical reasoning.

Our experiments are also limited to open base models up to 4B parameters. Training boundary-first traces and running GRPO require substantial GPU memory and sampling cost, and our available compute resources do not support a systematic study of larger models. This limits conclusions about scaling, but the evaluated models match the setting where distractor vulnerability is most pronounced and where post-training interventions are practically useful.

Finally, ReBound relies on LLM-assisted data construction and operational process metrics. The validation pipeline reduces manual annotation cost, but can inherit bias from the seed problems and proposal model, and more ambiguous distractors may require human verification. Likewise, DISTRACTOR ID, LEAKAGE, and BOUNDARY ACC. provide structured evidence of boundary-compliant behavior, but they are not complete proofs of internal model reasoning.

Ethics Statement

We hereby confirm that the entire work presented in this paper, including all content, analysis, and conclusions, is the original work of the human authors. While the core research, analysis, and conclusions are entirely the work of the human authors, generative AI was consulted for the limited purpose of improving the grammar and readability of the text.

References

Ujjwala Anantheswaran, Himanshu Gupta, Kevin Scaria, Shreyas Verma, Chitta Baral, and Swaroop Mishra. 2025. Cutting through the noise: Boosting llm performance on math word problems. *Preprint*, arXiv:2406.15444.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert Van Rooij, Kun Zhang, and Zhouchen Lin. 2025. Empowering llms with logical reasoning: A comprehensive survey. *arXiv preprint arXiv:2502.15652*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Subhabrata Dutta, Ishan Pandey, Joykirat Singh, Sunny Manchanda, Soumen Chakrabarti, and Tanmoy Chakraborty. 2024. Frugal lms trained to invoke symbolic solvers achieve parameter-efficient arithmetic reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17951–17959.

Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10028–10039, Bangkok, Thailand. Association for Computational Linguistics.

Silin Gao, Antoine Bosselut, Samy Bengio, and Emmanuel Abbe. 2026. Abstral: Augmenting llms' reasoning by reinforcing abstract thinking. *Preprint*, arXiv:2506.07751.

Silin Gao, Jane Dwivedi-Yu, Ping Yu, Xiaoqing Ellen Tan, Ramakanth Pasunuru, Olga Golovneva, Koustuv Sinha, Asli Celikyilmaz, Antoine Bosselut, and Tianlu Wang. 2025. Efficient tool use with chain-of-abstraction reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2727–2743, Abu Dhabi, UAE. Association for Computational Linguistics.

Ruixin Hong, Hongming Zhang, Xiaoman Pan, Dong Yu, and Changshui Zhang. 2024. Abstraction-of-thought makes language models better reasoners. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1993–2027, Miami, Florida, USA. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, Sawsan Alqahtani, M Saiful Bari, Mizanur Rahman, Mohammad Abdullah Matin Khan, Haidar Khan, Israt Jahan, Amran Bhuiyan, Chee Wei Tan, Md Rizwan Parvez, Enamul Hoque, Shafiq Joty, and Jimmy Xiangji Huang. 2024. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural*

Language Processing, pages 13785–13816, Miami, Florida, USA. Association for Computational Linguistics.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024. GSM-plus: A comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2961–2984, Bangkok, Thailand. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. Ministral 3. *Preprint*, arXiv:2601.08584.

Iman Mirzadeh, Keivan Alizadeh-Vahid, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, volume 2025, pages 94743–94765.

Yuxiao Qu, Anikait Singh, Yoonho Lee, Amrith Setlur, Ruslan Salakhutdinov, Chelsea Finn, and Aviral Kumar. 2025. Rlad: Training llms to discover abstractions for solving reasoning problems. *arXiv preprint arXiv:2510.02263*.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2025. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, volume 2025, pages 60808–60838.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand. Association for Computational Linguistics.

Yanbo Wang, Zixiang Xu, Yue Huang, Gao Chujie, Siyuan Wu, Jiayi Ye, Pin-Yu Chen, Xiuying Chen, and Xiangliang Zhang. 2026. Adaptive distraction: Probing llm contextual robustness with automated tree search. *Advances in Neural Information Processing Systems*, 38:165093–165131.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2025. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales. In *International Conference on Learning Representations*, volume 2025, pages 82731–82754.

Jason Weston and Sainbayar Sukhbaatar. 2023. System 2 attention (is something you might need too). *arXiv preprint arXiv:2311.11829*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Minglai Yang, Ethan Huang, Liang Zhang, Mihai Surdeanu, William Yang Wang, and Liangming Pan. 2025c. How is LLM reasoning distracted by irrelevant context? an analysis using a controlled benchmark. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13329–13347, Suzhou, China. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. 2024a. Natural language reasoning, a survey. *ACM Computing Surveys*, 56(12):1–39.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024b. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, volume 2024, pages 45040–45061.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024c. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14672–14685, Miami, Florida, USA. Association for Computational Linguistics.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. Mammoth: Building math generalist models through hybrid instruction tuning. In *International Conference on Learning Representations*, volume 2024, pages 40320–40341.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, and 1 others. 2024. A careful examination of large language model performance on grade school arithmetic. *Advances in Neural Information Processing Systems*, 37:46819–46836.

Jixiao Zhang and Chunsheng Zuo. 2025. GRPO-LEAD: A difficulty-aware reinforcement learning approach for concise mathematical reasoning in language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5642–5654, Suzhou, China. Association for Computational Linguistics.

Zhanke Zhou, Rong Tao, Jianing Zhu, Yiwen Luo, Zengmao Wang, and Bo Han. 2024a. Can language models perform robust reasoning in chain-of-thought prompting with noisy rationales? *Advances in Neural Information Processing Systems*, 37:123846–123910.

Zihao Zhou, Qiufeng Wang, Mingyu Jin, Jie Yao, Jianan Ye, Wei Liu, Wei Wang, Xiaowei Huang, and Kaizhu Huang. 2024b. Mathattack: Attacking large language models towards math solving ability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19750–19758.

A Data Construction and Trace Format

Seed Instances. In our experiments, we use the GSM8K training split as the seed dataset for constructing ReBound training data. Each seed instance provides a question and an answer field. We treat the question as the clean math word problem x_i, use the solution text in the answer field as the reference chain-of-thought solution, and parse the final numeric answer from the same field as the gold answer y_i. These seed instances are then used to extract the target, Signal facts, and reference solution path before constructing the paired distractor record.

Trace Format. Figure 6 shows the serialized boundary-first trace used as the training target. The same format is used for clean and distractor records: clean records leave the Distractor field empty, while distractor records place the inserted irrelevant fact in the Distractor field. In both cases, derivation steps cite only Signal fact identifiers.

Prompt Templates. Figures 7 and 8 provide the two LLM proposal prompts used during data construction. We use Qwen2.5-72B-Instruct as the proposal model for structured-field extraction and distractor synthesis. The extraction prompt normalizes a clean problem and reference solution into structured fields. The distractor synthesis prompt selects a plausible distractor pattern, writes an answer-preserving irrelevant fact, and inserts it into the problem context. The final training traces are formatted by program templates from the accepted structured fields.

Distractor Patterns. During distractor synthesis, the proposal model first selects one of five common irrelevant-context patterns and then realizes it as a natural sentence in the problem context:

Trace format

Input Problem:
{problem_text}

Target Output:
<filter>
Target: {target_quantity}
Scope: {effective_scope}

Signal:
- S1: {signal_fact_1}
- S2: {signal_fact_2}
- S3: {signal_fact_3}
...
Distractor:
- D1: {distractor_fact_1}
...
</filter>
<calc>
- [S1, S2] {derivation_step_1}
- [S1, S2, S3] {derivation_step_2}
- [S4] {derivation_step_3}
...
</calc>
{final_answer}

Figure 6: Serialized boundary-first trace format used for SFT and GRPO generation.

Prompt：Extract structured field

Task：
You are given a clean mathematical word problem, its reference solution, and its gold answer.
Your task is to extract structured fields that describe the reference solution: the target quantity, the effective scope, the minimal supporting facts, and a fact-grounded derivation.
Do not solve the problem using a new reasoning path. Instead, normalize the provided reference solution into the required JSON format.

Definitions:
- Target: the quantity asked by the problem.
- Scope: the effective condition under which the target should be solved, such as the relevant entity, time, total group, or condition.
- Signal facts: supporting facts from the problem statement that are used by the reference solution. A fact should usually contain a quantity together with its entity, unit, and condition. A rate, ratio, or conversion rule may be one fact if it is used as one relation in computation.
- Derivation: concise calculation steps grounded in Signal facts. Each step must cite the Signal fact IDs that support the calculation.
- Final answer: the provided gold final answer.

Rules:
1. Use only information stated in the problem as Signal facts.
2. Do not introduce external assumptions or unstated quantities.
3. Do not include intermediate computed values as Signal facts; put them in derivation steps.
4. Keep Signal facts minimal but sufficient for the reference solution.
5. If a problem statement contains facts not used by the reference solution, do not include them as Signal facts.
6. Every derivation step must cite the original Signal fact IDs that support the calculation, including facts that justify intermediate values reused in that step.
7. The derivation should follow the reference solution rather than inventing a different solution path.
8. The final derivation step must reach the provided gold answer.
9. If the reference solution is inconsistent with the problem or the gold answer, set "valid": false and explain the issue in "warnings".
10. Output valid JSON only. Do not output Markdown or any extra text.

Output schema:
```
{
  "target": "...",
  "scope": "...",
  "signal_facts": [
    {"id": "S1", "text": "..."},
    {"id": "S2", "text": "..."}
  ],
  "derivation": [
    {"refs": ["S1", "S2"], "text": "..."},
    {"refs": ["S1", "S2",, "text": "..."}
  ],
  "final_answer": "...",
  "valid": true,
  "warnings": []
}
```

Example 1：
.....
Example 2：
.....

Figure 7: Prompt template for structured-field extraction from a clean seed problem and its reference solution.

- *Outside time*: a fact from a different time, past event, future plan, or obsolete condition.
- *Wrong entity*: a quantity or relation belonging to another person, object, order, group, or owner.
- *Wrong category*: a fact about a different item type, attribute, unit, or measurement target.
- *Not actual*: a hypothetical, planned, canceled, conditional, or counterfactual fact.
- *Other context*: a fact from a nearby but separate location, activity, or subcontext.

These patterns cover common ways in which a fact can be plausible in the story while falling outside the current solution path.

Quality Control. We apply rule-based checks before accepting generated records. For extracted clean traces, we verify that the target matches the question, all Signal facts are stated in the original problem, and the solution path reaches the gold answer using cited Signal facts. For distractor records, we check that the inserted fact does not change the target question or gold answer, does not duplicate or contradict a Signal fact, and is not used by the reference solution path. We also verify the serialized trace: clean records must have an empty Distractor field, distractor records must place the inserted fact in the Distractor field, and derivation

Prompt：Distractor Fact Synthesis and Insertion Prompt

Task:
You are given a clean mathematical word problem and structured fields extracted from its reference solution.
Your task is to add one plausible but irrelevant fact to the problem context. The added fact should fit naturally into the story and be attention-competitive, but it must not change the target question, the reference derivation, or the gold answer.
First choose the most suitable distractor pattern, then write and insert the distractor fact into the problem.

Inputs:
- Problem: the original clean math word problem.
- Target: the quantity asked by the problem.
- Scope: the effective condition under which the target is solved.
- Signal facts: facts from the problem that support the reference solution.
- Derivation: the reference derivation grounded in Signal facts.
- Gold answer: the final answer that must remain unchanged.

Distractor patterns:
- outside_time: the added fact belongs to a different time, past event, future plan, or obsolete condition.
- wrong_entity: the added fact belongs to another person, object, order, group, or owner.
- wrong_category: the added fact concerns a different category, attribute, item type, or measurement target.
- not_actual: the added fact describes a hypothetical, planned, canceled, conditional, or counterfactual situation.
- other_context: the added fact belongs to a nearby but separate context, location, scene, or activity.

Requirements:
1. Add exactly one new distractor fact.
2. The distractor fact must be stated as part of the problem context, not as an explanation.
3. The inserted problem must preserve the original target question.
4. The inserted problem must preserve the gold answer.
5. The inserted problem must preserve all original Signal facts.
6. The distractor fact must not support any step in the reference derivation.
7. The distractor fact should be plausible in the same story and should not look artificially irrelevant.
8. The distractor fact should be attention-competitive, preferably involving a number, rate, relation, entity, or condition that a careless solver might consider.
9. Do not use words such as "irrelevant", "distractor", "not needed", "unrelated", or "extra information" in the inserted problem.
10. Do not contradict the original problem.
11. Do not introduce external knowledge.
12. Do not rewrite the problem more than necessary; only insert the distractor fact and make minimal grammatical adjustments.
13. If no safe distractor can be inserted without changing the answer, set "valid": false and explain the issue in "warnings".
14. Output valid JSON only. Do not output Markdown or any extra text.

Output schema:
{
  .....
}

Example 1：
.....
Example 2：
.....

Now process the following instance.
Problem:
{problem}
Target:
{target}
.....

Output:

Figure 8: Prompt template for distractor fact synthesis and insertion.

steps must cite only Signal identifiers. Records failing these checks are regenerated or removed. After validation, the construction pipeline yields 7,429 clean–distractor pairs, corresponding to 14,858 training records before splitting and downsampling.

B Training Details

Data Split. For the main ReBound-SFT setting, we use the clean-heavy0.5 configuration: after splitting by seed problem with seed 42, we keep all clean records and downsample distractor records to a 2:1 clean-to-distractor ratio. The aligned Qwen3-0.6B run uses 7,058 clean and 3,529 distractor records for training.

Training Setup. We train ReBound with LoRA adapters. SFT uses teacher forcing on paired boundary-first traces, and GRPO starts from the SFT adapter and optimizes sampled traces with the correctness-guided reward in Section 3.4. The main runs use bf16/tf32 on 4 NVIDIA RTX A6000 GPUs (48 GB each). Hyperparameters vary slightly across backbones in implementation; Table 5 reports the unified setting used for the main ReBound-full configuration.

Stage	Hyperparameter	Value
SFT	Objective	next-token prediction
	Data mix	clean:distractor = 2:1
	Epochs	3
	Learning rate	3×10^{-5}
	Scheduler / warmup	cosine / 0.03
	Max sequence length	1024
	LoRA	$r = 16, \alpha = 32$, dropout 0.05
	Target modules	attention and MLP projections
GRPO	Initialization	ReBound-SFT adapter
	Reward	correctness-guided ReBound reward
	Data mix	clean:distractor = 1:1
	Epochs	1
	Learning rate	5×10^{-7}
	KL coefficient	$\beta = 0.04$
	Generations per prompt	8
	Max completion / temp.	512 / 0.85

Table 5: Default hyperparameters for the main ReBound-full training runs.

Reward Design Details. The GRPO reward is implemented as a capped rule-based score. Let $\hat{a}(o)$ be the parsed final answer and $c(o) = I[\hat{a}(o) = y]$. We first compute

$$r_{\text{raw}}(o) = 2c(o) + r_{\text{fmt}}(o) + r_{\text{bd}}(o) + r_{\text{ref}}(o) + r_{\text{math}}(o) + r_{\text{final}}(o) - p_{\text{bad}}(o), \tag{3}$$

where r_{fmt} rewards a valid ReBound format, r_{bd} rewards correct Signal–Distractor separation, r_{ref} rewards derivation steps that cite Signal facts and avoid Distractor facts, r_{math} checks parsed computations, r_{final} checks consistency between the last derivation value and the final answer, and p_{bad} penalizes missing calculation steps, excessive length, multiple final answers, clean-input over-rejection, and use of unique Distractor values in the derivation. The final reward is

$$R(o) = \begin{cases} \min(r_{\text{raw}}(o), 0.2), & \text{if } \hat{a}(o) = \emptyset, \\ \min(r_{\text{raw}}(o), 0.7), & \text{if } c(o) = 0, \\ r_{\text{raw}}(o), & \text{otherwise.} \end{cases} \tag{4}$$

Thus, incorrect or unparsable completions can receive only limited process credit, while answer-correct completions are further ranked by respect for the relevance boundary and leakage avoidance.

C Baseline Details

We implement CoT-Base by prompting the base model with a standard chain-of-thought instruction. Distractor CoT-SFT uses the same clean and distractor questions constructed for our data, but its target output remains the original GSM8K-style solution rather than a boundary-first trace. CoT-RL performs clean GSM8K chain-of-thought SFT followed by answer-oriented GRPO, without Signal–Distractor supervision. AoT and AbstRaL-RL are reproduced by using Qwen2.5-72B-Instruct to generate trajectories that follow their respective reasoning formats, followed by the corresponding SFT or SFT+RL training. For controlled data scale, Distractor CoT-SFT, AoT, and AbstRaL-RL use the same number of training instances as the corresponding ReBound-SFT setting.

D Evaluation Details

Benchmark Setup. We evaluate final-answer accuracy on GSM8K, GSM-Symbolic, GSM-Plus, GSM-IC, GSM-DC, and ProblemMathIC. GSM8K measures clean grade-school math reasoning. GSM-Symbolic and GSM-Plus test perturbation robustness through symbolic substitutions, numerical changes, and reformulations that preserve the underlying reasoning structure. GSM-IC, GSM-DC, and ProblemMathIC evaluate distractor robustness, where the model must solve the target problem without relying on plausible but irrelevant context. For GSM-IC, we follow the original construction protocol and sample 6,000 examples for evaluation, consisting of 4,000 IC-2 and 2,000 IC-M instances. All methods are evaluated with the same answer extraction and normalization script under each backbone. Unless otherwise specified, decoding uses the same maximum generation length and configuration within each benchmark. Reported table accuracies are averaged over repeated inference runs with the same trained checkpoint, evaluation set, and scoring script; no additional retraining is performed.

Answer Scoring. For each generated response, we extract the final answer after the answer delimiter when available; otherwise, we fall back to the last numeric answer in the response. Extracted answers are normalized before exact or numeric-equivalence matching against the gold answer. The same scorer is used for all methods.

ReBound-Test Construction. For the process-level diagnostics in Section 4.3, we construct a held-out ReBound-Test set using the same data construction and validation pipeline as the training data. We use GSM8K test problems as seed instances and sample 500 validated distractor pairs. The gold structured fields are not provided to the model at inference time; they are used only to compute distractor identification, leakage, and bound-

ary accuracy.

Process Metrics. Distractor ID measures whether the inserted distractor is placed in the Distractor field. Leakage measures whether a Distractor fact is selected as Signal, cited in the derivation, or used through a unique distractor value. Boundary Acc. is the strictest metric: the final answer must be correct, the inserted distractor must be identified, and no leakage can occur.

E Output Length Analysis Details

For the output-length analysis in Section 4.5, we reuse generated responses from the main evaluation rather than running a separate decoding setting. The analysis is restricted to distractor benchmarks, covering GSM-DC, ProblemMathIC, and GSM-IC. For GSM-IC, we merge IC-2 and IC-M using sample-size weighting. We include CoT-Base, CoT-RL, AbstRaL-RL, and ReBound as representative methods spanning prompting, answer-oriented RL, abstraction-based RL, and relevance-boundary training. For each method–backbone–benchmark group, we compute the average number of generated tokens with the corresponding backbone tokenizer and pair it with the same final-answer accuracy used in the main results. The token averages are reported in Table 4 and used to draw Figure 5.

F Case Study

Figure 9 provides a qualitative comparison on a ProblemMathIC example. The problem asks for the total number of travel days when the trip takes 6 days by bus and half as many days by car. The speed values are contextually plausible, because speed is normally related to travel time, but they are not needed once the travel times are explicitly given.

CoT-Base uses the speed values to reconstruct a distance and then recomputes the return time, leading to an incorrect answer. AbstRaL-RL produces a structured trace and even derives the correct intermediate travel time of 9 days, but later reintroduces the speed-related distractor when computing distance and changes the final answer. This shows that making a rationale more structured does not by itself prevent irrelevant facts from entering later derivation.

ReBound avoids this failure by making the relevance decision explicit before computation. It places the speed change outside the relevance boundary and derives the answer only from the selected Signal facts: 6 days by bus, half as many days by car, and a return trip by car. The case illustrates the intended behavior behind the process-level diagnostics: ReBound is not merely changing the output format, but constraining the facts that can support the solution path.

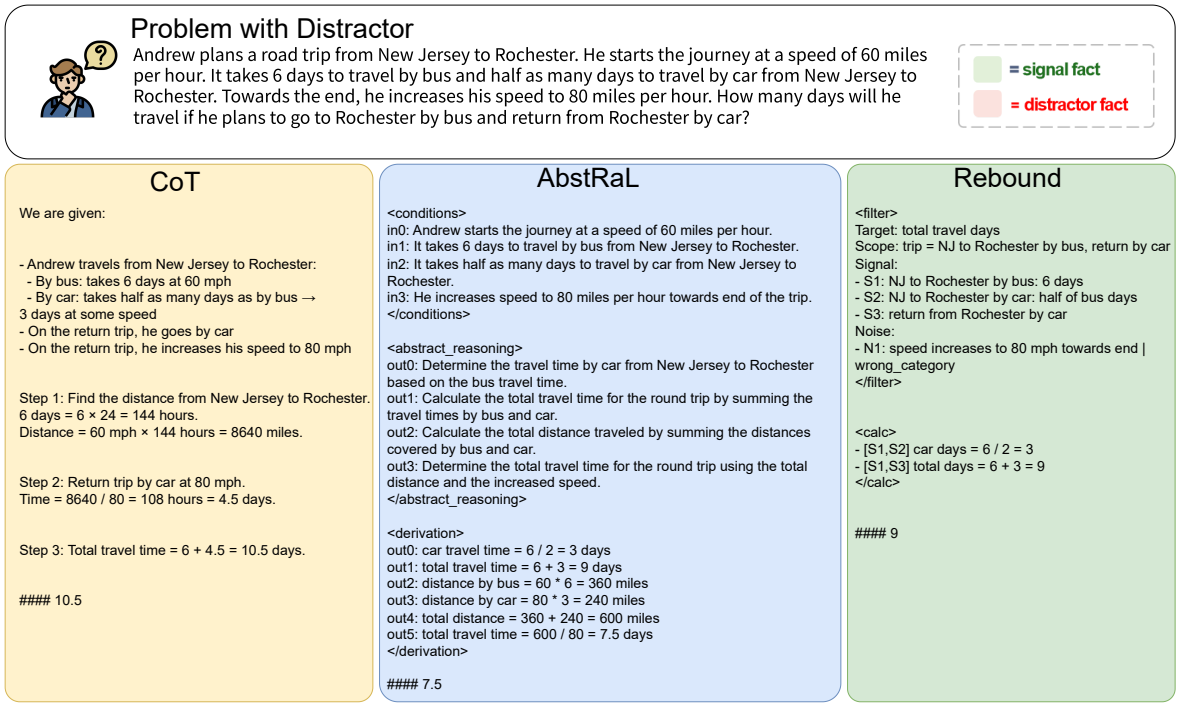


Figure 9: Case study comparing CoT-Base, AbstRaL-RL, and ReBound on a distractor-augmented math word problem. CoT-Base and AbstRaL-RL both allow the speed-related distractor to affect computation, whereas ReBound places the distractor outside the relevance boundary and derives the answer from Signal facts.