ReThink: Adversarially Robust Arithmetic Reasoning via Neuro-Symbolic Soft Constraints

Soikat Das[1] **Sparsh Mittal**[2] **Rekha Singhal**[3]

[1]CrimsonLogic Innovation, Singapore
[2]MFSDS&AI, IIT Roorkee, India
[3]TCS Research, New York, USA

soikatd@crimsonlogic.com sparsh@mfs.iitr.ac.in rekha@tcs.com

Abstract

While Large Language Models (LLMs) perform well on arithmetic word problems, they remain brittle to adversarial perturbations like irrelevant distractors or contradictory facts. To address this, we introduce RETHINK, a training-free neuro-symbolic framework that formulates adversarial arithmetic reasoning as a Weighted MaxSAT/MaxSMT problem. RETHINK uses a four-stage pipeline: it explicitly decomposes problems to strip noise, encodes the mathematical core as confidence-weighted soft constraints in the Z3 theorem prover, and cross-verifies the formal solution against a parallel Chain-of-Thought (CoT) path. This formulation can sacrifice lower-weight conflicting statements while satisfying higher-weight constraints. Evaluated on a balanced 2,400-example subset of GSM-Plus, RETHINK with GPT-4.1 achieves 81.62% adversarial accuracy, outperforming CoT, PAL, and Self-Verification. It narrows the robustness gap to 14.69 percentage points while achieving 96.31% clean accuracy.

1 Introduction

Large Language Models (LLMs) like GPT-4 (Achiam et al., 2023) have significantly advanced arithmetic reasoning, yet their autoregressive generation remains brittle to adversarial linguistic perturbations in arithmetic word problems. The accuracy of these LLMs degrades on adding semantic noise, such as distractor sentences or local contradictions. This suggests that their predictions can depend on surface cues rather than the relevant arithmetic relations. Existing methods fail to adequately resolve this vulnerability. Adversarial training is computationally expensive, generalizes poorly to unseen attacks, and lacks any formal verification mechanism(Stutz et al., 2020). Prompting and self-correction strategies attempt to elicit error detection and revision from the model; however, intrinsic self-correction often fails to improve reasoning and can even degrade performance (Huang et al., 2023). Moreover, when misleading or biasing features influence a model toward an incorrect answer, the resulting chain-of-thought may rationalize that answer without acknowledging the source of the bias (Turpin et al., 2023). Tool-augmented reasoning, such as PAL (Gao et al., 2023), delegates computation to executable programs but still relies on the model to accurately translate the natural-language problem into executable steps. Similarly, symbolic-solver approaches depend on correct problem formalization, as semantic parsing errors can produce incorrect constraints and consequently erroneous solutions (Pan et al., 2023). Robustness in this setting motivates a soft neuro-symbolic formulation: blending consistency checks over formal constraints with the linguistic adaptability needed to filter out adversarial noise.

Contributions: To bridge this gap, we propose ReThink, a training-free hybrid framework grounding Weighted MaxSAT/MaxSMT optimization within a dual-path cross-verification architecture for adversarial arithmetic word problems. We introduce a soft neuro-symbolic defense that frames conflicting constraint rejection as a formal optimization problem. The core principle is maximum global consistency among extracted constraints: ReThink can tolerate conflicting adversarial statements when they receive lower weights than the relevant arithmetic relations.

ReThink works in four stages: (1) Adversarial Decomposition, where the LLM segregates core arithmetic facts from semantic noise prior to symbolic translation; (2) Symbolic Translation and Solving, parsing the cleaned text into weighted Z3 soft constraints, which can sacrifice lower-weight conflicting constraints; (3) Neural Verification, generating a separate CoT reasoning trace in parallel; (4) Agreement-Based Decision, where symbolic and neural answers are reconciled, with disagree-

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 1303–1313
November 6-10, 2026 ©2026 Association for Computational Linguistics

ments resolved via cross-verification of both reasoning traces.

Evaluated on GSM-Plus (Li et al., 2024) with GPT-4.1, ReThink achieves 81.62% adversarial accuracy, outperforming CoT, PAL, and Self-Verification. It reduces the robustness gap to 14.69 percentage points (pp) while achieving 96.31% clean accuracy. The largest attack-wise gain occurs when the queried quantity is reversed, while gains on format conversion and problem understanding are smaller; this pattern points to explicit relation tracking as a particular strength rather than uniform protection across perturbations (Section 4).

2 Related Work

Mathematical Reasoning in LLMs: Chain-of-Thought (CoT) prompting (Wei et al., 2022) remains the cornerstone of reasoning elicitation in LLMs, wherein the model produces intermediate rationales prior to the final answer. Notably, CoT manifests as an emergent capability smaller models falter under multi-step reasoning, whereas larger ones navigate such paths with surprising fluency. The GSM8K benchmark, comprising 8,500+ diverse mathematical word problems, serves as the standard testbed (Cobbe et al., 2021).

Two prominent extensions strengthen this further: Self-Consistency (Wang et al., 2023) aggregates multiple sampled reasoning trajectories via majority voting, yielding more robust answers; and tool-augmented reasoning (Gao et al., 2023) grounds LLMs with external solvers or calculators, directly mitigating the persistent arithmetic fragility these models exhibit.

Adversarial Robustness in LLM Reasoning. Despite strong benchmark performance, LLMs remain brittle to irrelevant context: mathematically irrelevant information added to arithmetic word problems can substantially degrade model predictions (Shi et al., 2023). Recent robustness benchmarks consider perturbations such as numerical substitution, distraction insertion, and changes to arithmetic relations (Li et al., 2024), while adversarial methods have also demonstrated the vulnerability of mathematical reasoning to semantics-preserving word-level perturbations (Zhou et al., 2024). GSM-IC (Shi et al., 2023) demonstrates that even a single irrelevant sentence can reduce reasoning accuracy, whereas the broader GSM-Plus suite (Li et al., 2024) comprises eight perturbation types, including numerical substitution, distraction insertion, and reversing operation.

Controlled evaluations further suggest that apparent mathematical reasoning can be sensitive to superficial variations in problem instances. Mirzadeh et al. (Mirzadeh et al., 2024) hypothesize that such instability may partly arise from in-distribution pattern matching rather than fully invariant formal reasoning. These findings motivate reasoning methods that remain robust beyond clean, curated benchmark inputs.

Self-Correction, Verification, and Consistency. Self-correction offers an intuitive pathway to reasoning reliability, prompting LLMs to iteratively revisit and refine their own outputs (Madaan et al., 2023). However, intrinsic self-correction without external feedback often fails to improve reasoning and can even degrade initially correct responses (Huang et al., 2023). Moreover, models can produce plausible reasoning that rationalizes answers influenced by misleading input features (Turpin et al., 2023).

External verification provides an alternative mechanism for improving reasoning reliability. Verifier-based methods train a companion model to score candidate solutions, selecting the highest-ranked completion from a pool of generations rather than relying on a single pass (Cobbe et al., 2021). Orthogonally, multi-agent debate employs multiple model instances to propose and critique reasoning processes over several rounds before arriving at a final answer (Du et al., 2024). These findings motivate externalized verification mechanisms for arithmetic word-problem solving.

Neuro-Symbolic Reasoning and Solver-Integrated Methods. Neuro-symbolic frameworks combine the flexibility of language models with deterministic symbolic computation. Program-aided approaches such as PoT and PAL translate natural-language problems into executable programs and delegate computation to an external interpreter (Chen et al., 2022; Gao et al., 2023). Their correctness therefore still depends on accurately translating the problem into an appropriate program; errors in this translation can propagate to the final answer (Pan et al., 2023). SatLM instead adopts a declarative formulation, translating natural-language problems into logical constraints that are solved using automated reasoning tools such as Z3 (Ye et al., 2023; De Moura and Bjørner, 2008). While the solver can guarantee correctness with respect to the parsed specification, an incorrectly translated specification can still

produce an answer that does not correspond to the original problem (Ye et al., 2023).

Constraint Relaxation, Weighted MaxSAT/MaxSMT, and Dual-Path Verification. Weighted MaxSAT/MaxSMT relaxes the all-or-nothing rigidity of strict constraint satisfaction by assigning weights to soft constraints and maximizing the total weight of satisfied constraints (Bjørner et al., 2015), making it a natural fit when automatically extracted constraints may conflict. Solver-integrated approaches have explored incorporating differentiable logical constraints into LLM reasoning to improve global logical consistency (Zhang et al., 2024). ReThink instead employs a discrete four-stage pipeline anchored by an explicit *adversarial decomposition* stage that filters semantic noise prior to symbolic translation. Its *dual-path cross-verification* compares the Weighted MaxSAT/MaxSMT solver output with a separate neural Chain-of-Thought trace on adversarial arithmetic word problems.

These components address complementary failure modes: decomposition targets irrelevant or misleading statements, whereas explicit weighted constraints preserve arithmetic relationships when a perturbation alters the queried quantity. This distinction is important because GSM-Plus includes answer-changing variants in which the question target or arithmetic relation is modified, meaning that such cases cannot be handled by simply recovering the original clean problem's answer (Li et al., 2024).

3 Methodology

We present RETHINK, a training-free framework for adversarial arithmetic word problems. By formulating constraint solving as a Weighted MaxSAT/MaxSMT problem, our approach uses a formal solver to optimize over weighted constraints extracted from text and tolerate conflicts when the relevant relations receive higher weights.

3.1 Problem Formulation

We formalize arithmetic word-problem solving as a mapping $f : \mathcal{P} \to \mathcal{A}$ recovering the correct answer $a^*(p)$ for a problem $p \in \mathcal{P}$, where $\mathcal{A}$ contains numerical answers and an insufficient-information response for underdetermined questions. In our adversarial threat model, a perturbation function $\delta : \mathcal{P} \to \mathcal{P}$ transforms a clean problem p into a well-formed variant $p' = \delta(p)$ through numerical substitution, distraction insertion, reversing operation, or another GSM-Plus perturbation. Some variants preserve the answer, while others change a relevant value or the queried quantity, so the target is $a^*(p')$. Our objective is to construct an estimator $\hat{f}$ minimizing expected loss over the base-problem distribution $\mathcal{D}$ under perturbation, formulated as $\mathcal{L}(\hat{f}) = \mathbb{E}_{p\sim\mathcal{D}}\left[\mathbb{1}(\hat{f}(\delta(p)) \neq a^*(\delta(p)))\right]$.

3.2 ReThink Framework

To minimize $\mathcal{L}$, we shift the reasoning paradigm from sequential text generation to *constraint satisfaction with adversarial preprocessing*. As shown in Figure 1 and Algorithm 1, ReThink operates through four stages: (1) adversarial decomposition isolating the factual core; (2) symbolic translation into confidence-weighted Z3 constraints; (3) a separate Chain-of-Thought path; and (4) agreement-based decision to resolve neuro-symbolic conflicts through cross-verification.

Algorithm 1 ReThink Inference

Require: Problem p' (potentially adversarial)
Ensure: Predicted answer $\hat{a}$ or no answer

Stage 1: Adversarial Decomposition
$p_{\text{clean}} \leftarrow \text{Decompose}(p')$ ▷ Strip distractors/contradictions
Stage 2: Symbolic Translation and Solving
$(\mathcal{C}, \mathcal{W}) \leftarrow \text{LLM}_{\text{Trans}}(p_{\text{clean}})$ ▷ Z3 soft constraints
$a_{\text{z3}} \leftarrow \text{OptimizeSoft}(\mathcal{C}, \mathcal{W})$ ▷ Weighted MaxSAT/MaxSMT
Stage 3: Neural Verification
$a_{\text{cot}} \leftarrow \text{CoT}(p')$ ▷ On original problem
Stage 4: Agreement-Based Decision
if neither path succeeded **then**
  return no answer
else if only one path succeeded **then**
  return the successful answer
else if $a_{\text{z3}} = a_{\text{cot}}$ **then**
  return a_{z3} ▷ Agreed answer
else
  $\hat{a} \leftarrow \text{CrossVerify}(p', a_{\text{z3}}, a_{\text{cot}})$
  return $\hat{a}$ ▷ Adjudicated answer
end if

3.2.1 Stage 1: Adversarial Decomposition

The first stage explicitly separates adversarial detection from mathematical reasoning. Given a potentially adversarial problem p', the LLM is prompted to analyze each statement and classify it into one of four categories:

- CORE: Directly required to compute the answer (e.g., "John has 5 apples").
- STRUCTURAL: Defines a mathematical relationship (e.g., "total = A + B").
- DISTRACTOR: Irrelevant information that does not affect the answer (e.g., "The temperature is 72 degrees" in a shopping problem).
- CONTRADICTION: A statement that repeats an earlier fact with a different value, likely adversarial noise (e.g., "John has 5 apples" followed by "John has 18 apples").

The decomposition stage produces a *cleaned problem* p_{clean} intended to retain CORE and STRUCTURAL facts while removing semantic distractors and textual contradictions. This dedicated linguistic analysis step can reduce irrelevant or contradictory input to the downstream Z3 translation.

When contradictions are detected, the decomposer applies a *primacy heuristic*: the first occurrence of a fact is treated as authoritative, and subsequent contradictory restatements are classified as adversarial noise. If decomposition fails (e.g., the LLM does not produce a valid cleaned problem), the system falls back to the original problem text.

3.2.2 Stage 2: Symbolic Translation and Solving

We map the cleaned text p_{clean} to Z3 soft constraints using a semantic prompt. Facts retained as core or structural receive large confidence weights ($w \in [95, 100]$).

Constraint Formulation Let the hypothesis $\mathcal{M} = (\mathcal{C}, \mathcal{W})$ represent constraints $\mathcal{C} = \{c_1, \ldots, c_m\}$ mapped to corresponding weights $\mathcal{W} = \{w_1, \ldots, w_m\}$. We compute a variable assignment σ that maximizes the satisfied weight:

$$\hat{\sigma} = \arg\max_{\sigma \in \text{Domain}} \sum_{j=1}^{m} w_j \cdot \mathbb{1}(\sigma \models c_j) \qquad (1)$$

This Weighted MaxSAT/MaxSMT objective is implemented using Z3's `Optimize` engine with weighted soft constraints added via `add_soft`; the encoded constraints include arithmetic theories. To avoid unintended integer division, variables are typed as `z3.Int` for countable entities and `z3.Real` for continuous ones.

Weight Hierarchy and Outlier Rejection Weights denote semantic confidence. Structural definitions are fixed at $w = 100$. Explicit numerical facts are bounded within $w \in [95, 100]$, ambiguous values in $w \in [40, 60]$, and suspicious statements in $w \in [5, 15]$. If a lower-weight adversarial constraint slips past the filters and conflicts with higher-weight core constraints, the solver can sacrifice it to maintain consistency across the dominant set.

Self Correction Loop Translating natural language to Z3 syntax can yield errors. We allow up to $T_{\max} = 3$ translation attempts in total. When the generated code causes a syntax error, runtime failure, or solver timeout, we extract the error and query the LLM again for a patched translation. This can recover from transient mapping errors.

3.2.3 Stage 3: Neural Verification via Chain-of-Thought

In parallel with the symbolic path, the *original* adversarial problem p' is passed to a standard Chain-of-Thought solver, which generates a step-by-step natural language reasoning trace. CoT operates on the original input rather than the decomposed version, giving the two paths different representations of the problem. They share the same backbone LLM, so their errors need not be independent.

3.2.4 Stage 4: Agreement-Based Decision

The final stage compares the outputs of the symbolic and neural paths. The main cases are:

Agreement. If $a_{\text{z3}} = a_{\text{cot}}$, the neural and symbolic paths converge on the same answer, which is returned. Empirically, the two paths agree on 95% of adversarial instances (Section 4.4); agreement alone does not establish correctness.

Disagreement. If $a_{\text{z3}} \neq a_{\text{cot}}$, the system initiates a *cross-verification* step. A dedicated LLM call receives: (i) the original problem, (ii) the Z3 code and its output, and (iii) the CoT reasoning trace and its answer. The verifier is instructed to look for adversarial distractors, contradictions, and reasoning errors in both solutions, and to adjudicate which answer is correct. This cross-verification prompt explicitly primes the LLM to detect adversarial manipulations; its empirical contribution is measured in Section 4.3.

Partial Failure. If only one path produces a valid answer (e.g., the Z3 path times out or faces an

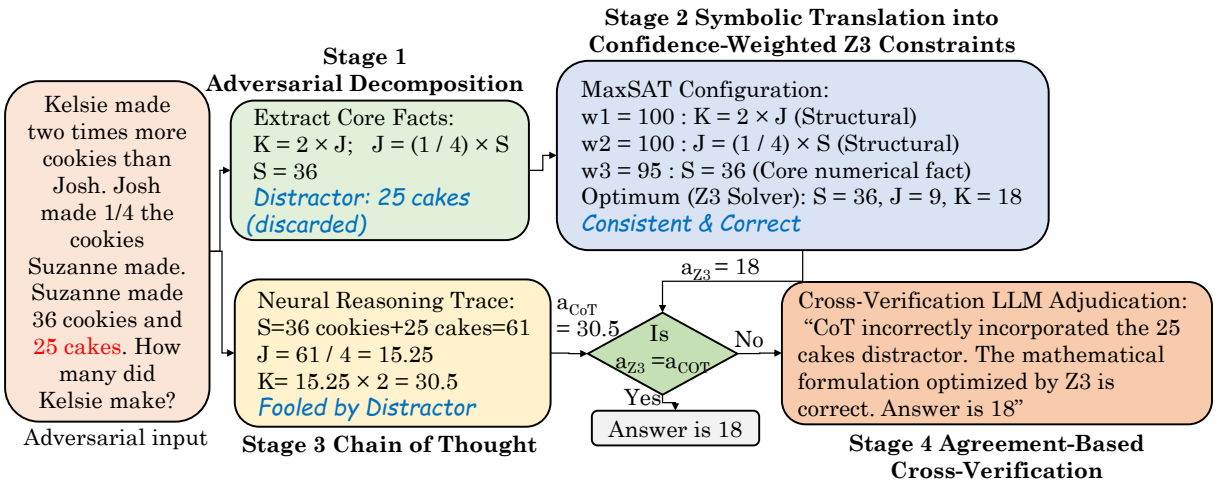


Figure 1: Overview of the RETHINK pipeline. The adversarial input is processed through a solver-based reasoning path and a parallel Chain-of-Thought path, and a final verification stage selects the output.

unrecoverable error), the system falls back to the successful path's answer. This permits an answer despite a failure in one path, although that answer may still be incorrect.

3.3 Mechanistic Analysis of Robustness

We describe the conditions under which soft-constraint optimization and neural verification can help on adversarial arithmetic problems.

Dual-Path Verification. The symbolic path reasons over extracted constraints, while CoT reasons over the original text. Their disagreement can expose a parsing or reasoning error for cross-verification, but agreement cannot guarantee correctness because both paths use the same LLM and may share an incorrect interpretation.

Conflict Tolerance via Soft Constraints. Equation 1 favors assignments satisfying greater total constraint weight. When a misleading statement is encoded as a lower-weight constraint that conflicts with the relevant arithmetic relations, an optimum can leave that statement unsatisfied. This benefit depends on accurate extraction and weighting; a wrong high-weight constraint can still produce a wrong answer. Decomposition provides a separate opportunity to remove irrelevant statements before constraint translation.

Defense in Depth. The four-stage pipeline addresses distinct threats: decomposition filters explicit distractors, soft constraints manage residual conflicts, CoT provides a separate reasoning path, and cross-verification adjudicates neural-symbolic disagreements. The observed benefit of each component is evaluated in Section 4.3. Changing the queried variable tests whether the encoded relations remain usable, whereas removing a necessary premise tests whether the system can recognize underdetermination; soft constraints address conflicts but cannot recover an absent fact.

4 Experimental Results

We benchmark against: (i) *CoT* (Wei et al., 2022), which elicits step-by-step reasoning; (ii) *PAL* (Gao et al., 2023), which generates executable Python code for computation; and (iii) *Self-Verification* (SV) (Weng et al., 2023), which re-checks intermediate steps through a secondary reasoning pass. Other implementation details, datasets, baselines, and metrics are discussed in the appendix.

4.1 Overall Performance

Table 1 presents the main performance comparison across all methods. RETHINK achieves 81.62% adversarial accuracy, outperforming the strongest baseline on this metric (CoT at 78.33%) by +3.29 percentage points (pp).

Table 1: Overall performance comparison on GSM-Plus with GPT-4.1; gaps are in percentage points.

Method	Clean (%)	Adv (%)	Gap (pp)
Baseline (CoT)	95.62	78.33	17.29
Defense: PAL	95.21	78.00	17.21
Defense: Self-Verification	94.21	77.79	16.42
ReThink (Ours)	**96.31**	**81.62**	**14.69**

RETHINK has the smallest robustness gap among the primary methods in Table 1. Its 14.69 pp gap is 1.73 pp below the smallest baseline gap (Self-Verification, 16.42 pp). RETHINK also achieves the highest clean accuracy (96.31%), so the observed adversarial gain does not come at the expense of

clean accuracy on this evaluation. Self-Verification illustrates why both measures matter: its gap is smaller than CoT's (16.42 versus 17.29 pp), yet its adversarial accuracy is also lower (77.79% versus 78.33%).

4.2 Attack-Specific Performance

Table 2 presents the per-attack adversarial accuracy, along with the improvement (Δ) of RETHINK over the strongest baseline in each category. RETHINK achieves its largest improvements on *reversing operation* (+7.33 pp) and *distraction insertion* (+4.33 pp). These gains are consistent with two distinct benefits: explicit constraints help track a changed queried quantity, while decomposition helps exclude irrelevant information. The reversal case study in Figure 2(c) illustrates the former without requiring a contradiction to be discarded.

Table 2: Adversarial accuracy (%) per attack type; Δ is the gain in percentage points over the strongest baseline in each row.

Attack Type	CoT	PAL	SV	ReThink	Δ
Numerical Substitution	88.33	87.67	88.00	**92.33**	+4.00
Digit Expansion	89.67	90.67	88.00	**93.00**	+2.33
Adding Operation	82.67	81.00	81.67	**86.67**	+4.00
Distraction Insertion	93.67	92.00	92.33	**98.00**	+4.33
Format Conversion	90.00	90.00	88.67	**91.00**	+1.00
Reversing Operation	87.67	88.33	88.67	**96.00**	+7.33
Problem Understanding	94.67	94.33	95.00	**96.00**	+1.00
Critical Thinking	0.00	0.00	0.00	0.00	+0.00
Overall	78.33	78.00	77.79	**81.62**	+3.29

On *numerical substitution* (+4.00 pp) and *adding operation* (+4.00 pp), RETHINK also gains over the strongest baseline. A substituted value may be part of the correct perturbed problem, so it cannot be discarded simply because it differs from the clean version. Soft constraints can instead help when an extracted statement conflicts with higher-weight relevant relations. For *problem understanding* (+1.00 pp) and *format conversion* (+1.00 pp), improvements are more modest. All methods achieve 0% accuracy on *critical thinking* perturbations; the current results do not establish which stage causes these failures. Thus, the seven categories with a determinable arithmetic answer show positive gains over their strongest baselines, but the 300 critical-thinking cases contribute no correct answers for any method. This separates robustness to changed or distracting information from recognizing missing information.

4.3 Ablation Study

Table 3 presents the ablation study results.

1. Removing decomposition causes the smallest adversarial drop, 0.62 pp. Its component impact is largest on distraction insertion (Appendix Table 8), where explicit noise filtering is most directly applicable.

Table 3: Ablation study on GSM-Plus (accuracies in %; gaps in percentage points).

Variant	Clean (%)	Adv. (%)	Gap (pp)
0. ReThink	**96.31**	**81.62**	14.69
1. ReThink (w/o Decomposition)	96.08	81.00	15.08
2. ReThink (w/o Cross-Verification)	96.25	80.54	15.71
3. ReThink (w/o Soft Constraints)	95.17	79.92	15.25
4. ReThink (w/o Z3)	94.42	79.79	**14.63**

2. Removing cross-verification reduces adversarial accuracy by 1.08 pp. This component activates on approximately 5% of samples where Z3 and CoT produce different answers. Without it, the system defaults to CoT on all disagreements, forgoing opportunities where the symbolic solver correctly identified adversarial manipulation.

3. Replacing soft constraints with hard constraints reduces adversarial accuracy by 1.71 pp. When the translated hard constraints conflict, the solver cannot satisfy all of them, whereas the weighted formulation can leave lower-weight constraints unsatisfied. This ablation supports the value of conflict tolerance in the tested arithmetic problems (Section 3.3).

4. Removing Z3 leads to the largest adversarial performance drop (1.83 pp). In this variant, the system relies only on decomposition followed by CoT reasoning. The result indicates that Z3's formal constraint satisfaction provides value beyond what neural reasoning alone can achieve, even when aided by decomposition. Its slightly smaller gap (14.63 versus 14.69 pp) comes with lower clean and adversarial accuracies (94.42% and 79.79%), illustrating that a smaller gap alone need not mean better adversarial performance. The similar drops from removing Z3 (1.83 pp) and replacing soft constraints (1.71 pp) indicate that both symbolic solving and the ability to relax conflicting extracted facts contribute to the observed gain; decomposition alone does not recover the full result.

4.4 Agreement Analysis

A key design insight of RETHINK is its hybrid architecture, which combines symbolic constraint solving with neural Chain-of-Thought (CoT) reasoning. Table 4 summarizes the relationship between the two reasoning paradigms on adversarial examples.

High Agreement Rate. The symbolic and neural

Table 4: Agreement analysis of the Z3 and CoT paths in RETHINK on adversarial examples. Rates and accuracies are in %; absolute improvement is in percentage points.

Measure	Value
ReThink adversarial accuracy	81.62
CoT adversarial accuracy	78.33
Absolute improvement	3.29
Z3–CoT agreement rate	95.00
Z3–CoT disagreement rate	5.00

reasoning paths agree on 95.00% of adversarial examples. This high agreement indicates that the two independently constructed reasoning paths frequently converge on the same prediction, providing a useful cross-verification signal. Agreement does not necessarily imply correctness, however, because both reasoning mechanisms can occasionally converge on the same incorrect answer.

Complementary Reasoning. Despite their high agreement, the two paths differ on a small but important subset of examples. Overall, CoT achieves 78.33% adversarial accuracy, whereas RETHINK achieves 81.62%, an absolute improvement of 3.29 percentage points, or 79 net correct answers across 2,400 adversarial examples. Since the final answer equals CoT when the paths agree, this net gain arises when the paths disagree or only one produces an answer. The 95% agreement rate therefore leaves a small but consequential set for adjudication; the aggregate data do not identify how many disagreements each path resolves correctly.

Cross-Verification and Residual Errors. The agreement analysis further motivates the use of cross-verification in RETHINK: agreement between symbolic and neural reasoning provides additional support for a prediction, while disagreement identifies cases requiring greater scrutiny. At the same time, examples on which both reasoning mechanisms fail remain beyond what cross-verification alone can resolve, representing residual failure modes of the current hybrid framework. The 95% agreement rate is therefore a routing signal for adjudication, not an accuracy or confidence estimate.

4.5 Cost-Accuracy Trade-off

The full RETHINK pipeline uses 3–6 LLM calls per sample, depending on translation retries and cross-verification, compared to 1 call for CoT and PAL, and 2 for Self-Verification (Table 5). With this additional call budget, RETHINK achieves a +3.29 pp improvement in adversarial accuracy. The decomposition-only variant (2 calls) yields a +1.46 pp gain, showing a smaller improvement with fewer calls. The full pipeline adds 1.83 pp over that two-call variant, quantifying the collective benefit of the remaining symbolic and verification stages against their additional call cost; latency was not measured.

Table 5: Cost-accuracy trade-off on GSM-Plus. ΔAdv is in percentage points relative to baseline CoT.

Method	LLM Calls	Adv Acc (%)	ΔAdv
Baseline CoT	1	78.33	—
PAL	1	78.00	−0.33
Self-Verification	2	77.79	−0.54
Decomp. Only (no Z3)	2	79.79	+1.46
Full ReThink	3–6	**81.62**	**+3.29**

4.6 Case Studies

Figure 2 presents three case studies that illustrate representative failure modes of neural reasoning under adversarial perturbations. Figure 2(a) shows a modified travel problem in which an additional stop changes John's return journey. Baseline CoT returns 45 miles, corresponding to reasoning over the original version of the problem and failing to incorporate the modified travel conditions. RETHINK instead represents the journey as a set of explicit constraints over the individual travel segments. John first travels $3 \times 60 = 180$ miles from home. During the four-hour return period, he spends two hours in traffic, drives for half an hour at 30 mph, stops for another half-hour, and then drives the remaining hour at 80 mph. The resulting return distance is $0 + 15 + 0 + 80 = 95$ miles, leaving $180 - 95 = 85$ miles from home. This case illustrates how explicit constraint modeling can preserve consistency across multiple modified quantities and prevent earlier assumptions from propagating through the reasoning process.

Figure 2(b) demonstrates the importance of decomposition when the problem contains an adversarially inserted but irrelevant condition. The modified savings problem introduces an additional $5 daily hobby expense while explicitly stating that it does not affect Elvis's savings target. Without decomposition, the symbolic translation fails to produce a valid answer. With decomposition, RETHINK separates the relevant numerical relations from the distracting information before constructing the constraint system. If Elvis saves x per day for the first 15 days and $2x$ per day for the next 15 days, the solver enforces $15x + 15(2x) = 1125$,

John drives for 3 hours at a speed of 60 mph and then turns around because he realizes he forgot something very important at home. He tries to get home in 4 hours but spends the first 2 hours in standstill traffic. He spends the next half-hour driving at a speed of 30mph, before being able to drive the remaining time of the 4 hours going at 80 mph. In addition, due to traffic jams on the road, he stops for half an hour. How far is he from home at the end of those 4 hours?

(a) GT/CoT/Rethink = 85/45/85 miles

Elvis has a monthly saving target of $1125. In April, he decides to take on a new strategy to hit his target. He plans to save twice as much daily in the second half of the month as he saves in the first half. However, during the first half of the month, he also decides to spend $5 daily on a new hobby, which he stops doing in the second half of the month. This spending does not affect his saving target. How much does he have to save for each day in the second half of the month?

(b) GT/No-decomposition/Rethink = $50/None/$50

Jackson is planting tulips. He can fit 6 red tulips in a row and 8 blue tulips in a row. Jackson buys some red tulips and 24 blue tulips. If he will plant 9 rows of flowers, how many red tulips does Jackson buy?

(c) GT/CoT/Rethink = 36/54/36 tulips

Figure 2: Three GPT-4.1 case studies (GT = ground truth). "No-decomposition" denotes an ablation of RETHINK, while "None" denotes a null prediction.

yielding $x = 25$ and therefore $2x = \$50$. This case highlights the practical role of decomposition in preventing irrelevant adversarial information from corrupting the subsequent symbolic reasoning stage.

Figure 2(c) shows a question-reversal attack in which the queried quantity is changed while the underlying mathematical relations remain the same. The problem provides 24 blue tulips arranged in rows of eight and a total of nine rows, then asks for the number of red tulips arranged in rows of six. Baseline CoT returns 54, effectively treating all nine rows as red and failing to account for the rows already occupied by blue tulips. RETHINK instead preserves the structural relation between the quantities through the constraint $r/6 + 24/8 = 9$, where r denotes the number of red tulips. Since the blue tulips occupy $24/8 = 3$ rows, the remaining $9 - 3 = 6$ rows contain $6 \times 6 = 36$ red tulips. This example illustrates how declarative constraints remain valid even when the queried variable is reversed, allowing RETHINK to track which quantities are known and which must be inferred. Together, the cases show three different useful operations: retaining all time segments, excluding an irrelevant expense, and re-solving for a changed target. Only the savings example directly illustrates distractor removal, so the improvements should not all be attributed to dropping conflicting constraints.

4.7 Small Language Model (SLM) Analysis

To examine whether RETHINK extends beyond GPT-4.1 on adversarial arithmetic problems, we evaluate the framework on two small language models: Qwen2.5-Coder-7B-Instruct and Meta-Llama-3-8B. The reported Qwen2.5 and Llama-3 evaluations contain 200 and 300 examples, respectively.

Tables 6 and 7 present the overall and attack-specific performance on these GSM-Plus evaluations. On both models, RETHINK improves adversarial accuracy over CoT while maintaining strong clean accuracy.

Table 6: Overall performance on two SLMs (Qwen2.5: n=200; Llama-3: n=300). Accuracies are in %; gaps are in percentage points.

	Qwen2.5			Llama-3		
Method	**Clean**	**Adv**	**Gap (pp)**	**Clean**	**Adv**	**Gap (pp)**
Baseline (CoT)	78.50	54.50	24.00	74.33	52.00	22.33
PAL	80.50	62.50	**18.00**	70.00	53.67	**16.33**
Self-Verification	78.50	55.50	23.00	74.33	50.67	23.67
ReThink (Ours)	**83.50**	**63.00**	20.50	**75.33**	**54.67**	20.67

Improved Overall Robustness. For Qwen2.5, adversarial accuracy improves from 54.50% under CoT to 63.00% with RETHINK, a gain of 8.50 percentage points, while clean accuracy reaches 83.50%, the highest among all methods. Although PAL achieves a smaller robustness gap (18.00 versus 20.50 pp), RETHINK delivers the highest adversarial accuracy. This distinction shows why gap and adversarial accuracy must be considered together when clean accuracies differ.

A smaller gain appears for Llama-3, where adversarial accuracy increases from 52.00% to 54.67% and clean accuracy reaches 75.33%. Across these two evaluated models, the pipeline improves adversarial accuracy over CoT, although PAL is within 0.50 pp on Qwen2.5 and 1.00 pp on Llama-3.

Attack-Specific Resilience. Table 7 shows where the improvements arise. For Qwen2.5, the largest gains occur in perturbations that introduce distracting or misleading information. In *Distraction Insertion*, adversarial accuracy reaches 86.36%, outperforming both CoT and PAL. Improvements are also observed for *Problem Understanding* and *Reversing Operation*, where the structured reasoning pipeline helps recover the correct mathematical relationships despite adversarial noise.

For Llama-3, RETHINK remains competitive across several perturbation categories. It achieves the strongest performance on *Distraction Insertion*

Table 7: Adversarial accuracy (%) per attack type on GSM-Plus (Qwen2.5: n=200; Llama-3: n=300).

Model	Attack Type	CoT	PAL	Self-V.	ReThink
Qwen2.5-Coder-7B-Instruct					
	Numerical Substitution	61.11	**66.67**	**66.67**	**66.67**
	Digit Expansion	69.57	**82.61**	78.26	78.26
	Adding Operation	58.62	**65.52**	**65.52**	**65.52**
	Distraction Insertion	68.18	81.82	72.73	**86.36**
	Format Conversion	51.52	**60.61**	45.45	57.58
	Reversing Operation	55.17	58.62	48.28	**62.07**
	Problem Understanding	65.38	76.92	65.38	**80.77**
	Critical Thinking	0.00	0.00	0.00	0.00
	Overall	54.50	62.50	55.50	**63.00**
Meta-Llama-3-8B					
	Numerical Substitution	52.78	**58.33**	50.00	55.56
	Digit Expansion	59.46	**70.27**	56.76	62.16
	Adding Operation	**65.85**	63.41	58.54	**65.85**
	Distraction Insertion	65.62	68.75	59.38	**71.88**
	Format Conversion	55.00	52.50	**57.50**	**57.50**
	Reversing Operation	51.22	48.78	**56.10**	**56.10**
	Problem Understanding	64.86	**67.57**	64.86	**67.57**
	Critical Thinking	0.00	0.00	0.00	0.00
	Overall	52.00	53.67	50.67	**54.67**

and matches or exceeds the best baseline on *Adding Operation*, *Problem Understanding*, and *Reversing Operation*. This suggests that extracting declarative constraints helps the model focus on the true structure of the problem rather than superficial cues. Distraction insertion is the only attack category on which RETHINK strictly exceeds every baseline for both small models; numerical substitution and digit expansion instead favor PAL on at least one model. The cross-model pattern supports selective relevance filtering as a repeatable strength in these evaluations, while showing that symbolic translation does not uniformly improve every numerical perturbation. All methods struggle on the *Critical Thinking* category, where adversarial accuracy remains zero across models, highlighting a persistent challenge for current SLMs. Overall, these two small-model evaluations show adversarial accuracy gains over CoT, but the zero results on critical thinking and the smaller margins over PAL limit broader conclusions.

5 Conclusions

RETHINK presents a training-free neuro-symbolic framework for adversarial arithmetic word problems, using Weighted MaxSAT/MaxSMT optimization over extracted constraints. Structured decomposition, confidence-weighted constraints, solver-based reasoning, and dual-path verification address different sources of error. On GSM-Plus with GPT-4.1, RETHINK achieves 81.62% adversarial accuracy and a 14.69 percentage-point robustness gap, outperforming the evaluated neural and tool-augmented baselines while achieving the highest clean accuracy. The largest attack-wise gain is on reversing operation (+7.33 pp over the strongest baseline), consistent with the value of explicitly tracking the queried quantity; the hard-constraint ablation further supports the value of tolerating conflicting extracted information. Taken together, the results support a focused conclusion: explicit relations help when the target changes, and soft constraints help when extracted facts conflict, while neither mechanism alone addresses questions with missing premises.

Limitations

Despite its strong empirical performance, RETHINK has several limitations. First, the framework relies on the LLM's ability to accurately translate natural language problems into structured logical constraints. When semantic parsing fails, the system may fall back to Chain-of-Thought reasoning or may produce an incorrect symbolic answer from an apparently valid constraint set.

Second, integrating a formal solver introduces computational overhead and possible solver timeouts, particularly when constraint sets become complex or poorly structured. The cost analysis reports LLM call counts rather than measured end-to-end latency.

Third, the *critical thinking* category can remove information needed to determine a numerical answer. All evaluated methods score 0% on this category. Soft-constraint optimization resolves conflicts among extracted facts but cannot supply missing premises; detecting underdetermination requires an additional check, and the aggregate results do not isolate the precise failure stage. Likewise, the 95% Z3–CoT agreement rate does not certify correctness, since both paths can share an incorrect interpretation of the input.

Finally, our evaluation focuses on adversarial arithmetic word problems in GSM-Plus. While the results demonstrate robustness improvements in this setting, it remains an open question whether the same neuro-symbolic design generalizes to broader reasoning domains or other adversarial scenarios.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Nikolaj Bjørner, Anh-Dung Phan, and Lars Fleckenstein. 2015. νz-an optimizing smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 194–199. Springer.

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*.

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*.

Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International conference on machine learning*, pages 10764–10799. PMLR.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

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.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824.

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 *International Conference on Machine Learning*, pages 31210–31227.

David Stutz, Matthias Hein, and Bernt Schiele. 2020. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International conference on machine learning*, pages 9155–9166. PMLR.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

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.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. Satisfiability-aided language models using declarative prompting. *arXiv preprint arXiv:2305.09656*.

Yu Zhang, Hui-Ling Zhen, Zehua Pei, Yingzhao Lian, Lihao Yin, Mingxuan Yuan, and Bei Yu. 2024. Sola: Solver-layer adaption of llm for better logic reasoning. *arXiv preprint arXiv:2402.11903*.

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

A Experimental Setup

Dataset. We evaluate RETHINK on the **GSM-Plus** benchmark, an adversarial extension of GSM8K designed to test robustness in arithmetic word problems. Our GPT-4.1 evaluation spans eight perturbation categories: numerical substitution, digit expansion, operation addition, distraction insertion, integer–decimal–fraction conversion, operation reversal, problem understanding, and critical thinking. The last category can remove facts needed for a numerical answer. Each attack category contains 300 instances, yielding 2,400 adversarial evaluation cases paired with their corresponding clean problems. The smaller-model evaluations use 200 Qwen2.5 and 300 Llama-3 examples (Section 4.7). This balanced main evaluation gives each attack type one eighth of the overall adversarial score, making the uniformly failed critical-thinking category consequential for every method's aggregate accuracy.

Baselines. We compare RETHINK against three representative reasoning paradigms: zero-shot Chain-of-Thought (CoT) prompting, Program-Aided Language Models (PAL), and Self-Verification.

Implementation Details. The main experiments use GPT-4.1 accessed via the Azure OpenAI API as the backbone model; the additional small-model analysis uses Qwen2.5-Coder-7B-Instruct and Meta-Llama-3-8B. Constraint solving is performed using Z3 (v4.12.2) with a 15-second timeout per optimization call. Deterministic decoding (temperature $= 0.0$) is used across methods, and the translation loop allows at most three attempts.

Evaluation Metrics. We report *Clean Accuracy* on original problems and *Adversarial Accuracy* on perturbed variants. We additionally compute the *Robustness Gap* (Clean $-$ Adv), reported in percentage points, which measures the degradation in performance under adversarial perturbations. Gap and adversarial accuracy should be read together when clean accuracies differ.

B Component Impact Analysis

Table 8 reports the number of samples where adding each component flipped the prediction from incorrect to correct ("Helped") or correct to incorrect ("Hurt"). In this evaluation, the soft-constraint mechanism helped 41 samples and hurt none. This observed result supports its contribution to adversarial accuracy, but does not guarantee zero regressions on other samples or datasets.

Table 8: Per-component impact on adversarial accuracy (n=2400). "Helped" counts samples where adding the component flipped the answer from wrong to correct; "Hurt" is the reverse.

Component	Helped	Hurt	Net	Primary Benefit
Decomposition	22	7	$+15$	Distraction insertion ($+8$)
Cross-Verification	27	1	$+26$	Numerical substitution ($+9$)
Soft Constraints	41	0	$+41$	Reversing operation ($+12$)
All Combined	—	—	**$+79$**	**$+3.29$ pp over CoT**

Secondly, cross-verification helps 27 samples and hurts 1; thus 27 of its 28 answer-changing effects (96.4%) are beneficial. This fraction describes the observed flips, not all disagreement cases.

The decomposition stage helps 22 samples but also hurts 7, yielding a net gain of $+15$. These regressions show that removing context can sometimes harm the final answer; the aggregate counts do not establish which attack categories or parsing errors account for them. Soft constraints have the largest individual net effect (+41), consistent with the 1.71 pp drop when they are replaced by hard constraints in Table 3. This links the aggregate ablation to sample-level corrections without treating the component effects as independent. Finally, the individual net gains (15 + 26 + 41 = 82) exceed the combined gain of 79 by three net correct answers. The gains are therefore not strictly additive; aggregate counts alone cannot determine how many samples involve overlapping component effects.