Do Intermediate Reasoning Steps Matter? Evaluating Generalization Under Data Corruption

Noam Barlin **Dan Ayzik** **Ido Friedman**

Abstract

Chain-of-thought (CoT) prompting has become a widely used technique for improving reasoning in large language models (LLMs), yet it remains unclear whether models genuinely rely on their intermediate steps to derive final answers. In this work, we design a controlled experimental framework to test the causal role of CoT in mathematical reasoning. Using GSM8K, we constructed three training datasets that deliberately decouple reasoning steps from final answers: (1) correct steps with correct answers, (2) wrong steps with correct answers, and (3) correct steps with wrong answers. We fine-tuned `Phi-3.4B-mini` on each dataset and compared performance to the unmodified base model. Models were evaluated on both final answer accuracy (Exact Match) and reasoning quality (step-level correctness), including external verification with ThinkPRM. Our results show that models trained on mismatched data produce reasoning and answers that diverge sharply: they can generate correct-looking steps while producing incorrect answers, or fail to maintain reasoning quality despite producing correct outcomes. These findings provide direct evidence that CoT can function as a superficial pattern rather than a causal reasoning process, with implications for the interpretability and reliability of LLMs. [1]

1 Introduction

Large language models (LLMs) have recently demonstrated remarkable performance on reasoning-intensive tasks when prompted to "think step by step" using chain-of-thought (CoT) reasoning (Wei et al., 2022). While CoT has become a dominant paradigm for improving model accuracy on benchmarks such as GSM8K (Cobbe et al., 2021), a central open question remains unresolved: *do models truly rely on their intermediate reasoning steps to arrive at correct answers, or do they merely mimic plausible reasoning patterns while generating the final answer independently?*

Prior work has suggested that CoT may act as a form of *rationalization* rather than genuine reasoning (Turpin et al., 2023; Lanham et al., 2023). That is, models can output coherent-looking reasoning chains even when those steps are not causally linked to the correctness of the final prediction. Understanding whether CoT improves reasoning or simply aligns with surface-level patterns is crucial for both the interpretability and reliability of LLMs.

This work proposes a controlled experimental approach that enables a direct assessment of the dependency between reasoning processes and their resulting answers. Using GSM8K, a standard benchmark for mathematical reasoning, we constructed three distinct training datasets:

1. correct reasoning steps with correct final answers,
2. wrong reasoning steps with correct final answers, and
3. correct reasoning steps with wrong final answers.

We fine-tuned the `Phi-3.4B-mini` model on each dataset, alongside an unmodified base model for comparison. This setup allows us to disentangle whether models prioritize aligning with reasoning steps or producing correct outcomes.

We evaluate models both on final answer accuracy (Exact Match) and reasoning quality (step correctness), and further analyze alignment using a verifier model (ThinkPRM) that scores intermediate reasoning steps. Our findings show that models trained on mismatched datasets can produce reasoning and answers that diverge significantly, providing direct evidence that CoT does not always

[1] Code, datasets, and experimental scripts are available at https://github.com/danayzik/nlp_final_project/tree/main.

function as a causal mechanism for generating correct outputs.

Our contributions are as follows:

- We design a controlled experimental framework to test whether models rely on CoT for generating final answers.
- We fine-tune models on curated datasets that deliberately decouple reasoning steps from final answers.
- We provide empirical evidence that CoT can act as a superficial pattern, with correct-looking steps failing to ensure correct final answers.

These results highlight limitations in the current use of CoT prompting and shed light on the distinction between reasoning as *explanation* versus reasoning as a *causal process*.

2 Related Work

Chain-of-Thought Prompting. Chain-of-thought (CoT) prompting has been widely adopted as a method for eliciting step-by-step reasoning in large language models. By encouraging intermediate reasoning, CoT has significantly improved performance on tasks such as math word problems, commonsense reasoning, and symbolic manipulation (Wei et al., 2022; Cobbe et al., 2021). Despite its empirical success, the mechanism by which CoT enhances accuracy remains unclear: it is debated whether models genuinely *use* their reasoning steps to arrive at final answers, or whether CoT primarily provides a scaffolding that mimics human reasoning.

Faithfulness of Explanations. Recent work has questioned the reliability of CoT as an explanation of model behavior. Turpin et al. (2023) show that language models can produce unfaithful CoT explanations, where intermediate steps look plausible but are not causally linked to the final prediction. Similarly, Lanham et al. (2023) measure faithfulness in CoT reasoning and demonstrate that models often fail to maintain alignment between reasoning quality and answer correctness. These findings suggest that high-quality reasoning traces do not necessarily imply correct final predictions.

Process Supervision and Verifiers. An alternative to outcome-based supervision is to evaluate models at the level of intermediate reasoning steps. Datasets such as PRM800K (Lightman et al., 2023) enable training verifiers that can score the correctness of individual steps, rather than only final answers. Verifier models like ThinkPRM extend this line of research by providing fine-grained feedback on reasoning chains, making it possible to decouple reasoning quality from outcome accuracy. Our work builds on this perspective: by constructing datasets that deliberately mismatch reasoning steps and final answers, we analyze whether models trained under these conditions maintain alignment between step-level correctness and final predictions.

Summary. While prior studies have raised concerns about the faithfulness of CoT and proposed verifiers as a solution, few works directly probe the causal relationship between reasoning steps and final answers. Our study contributes to this gap by introducing a controlled training setup that disentangles reasoning quality from answer correctness, allowing us to evaluate whether models truly rely on their reasoning steps to generate final answers.

3 Methodology

The methodology consists of four components: dataset construction, model training, generation and verification, and results aggregation.

3.1 Dataset Construction

We used GSM8K (Cobbe et al., 2021), a standard benchmark for grade-school math word problems, as the base dataset. Each example consists of a natural language question and a step-by-step solution with a final numerical answer. Using custom preprocessing scripts, we created three derived training sets:

1. **Correct Steps, Correct Answer (CS–CA):** Original GSM8K solutions with accurate intermediate steps and correct final answers.
2. **Wrong Steps, Correct Answer (WS–CA):** Intermediate steps were corrupted by shuffling, altering numbers, and distorting operators, while the final answer remained correct.
3. **Correct Steps, Wrong Answer (CS–WA):** Intermediate steps remained correct, but the final answer was perturbed by adding noise to the true answer.

Data splits. We curated three training sets from the original GSM8K training data (7,473 examples) under the three conditions described above. For model selection and performance monitoring during training, we constructed a 500-example **validation set** from the original GSM8K test split. In addition, we created a disjoint 500-example **test set** from the remaining test examples, used exclusively for final evaluation and comparison across models. These two held-out sets share no overlap with each other or with the training data, ensuring a clean separation between model development and final reporting.

3.2 Model Training

We fine-tuned the `Phi-3.4B-mini-instruct` model using QLoRA in 4-bit precision. Each training condition (CS–CA, WS–CA, CS–WA) produced a separate model variant, while the base model was kept as an untrained reference. Training was conducted with the following setup:

- Optimizer: paged AdamW with cosine learning rate schedule
- Sequence length: 1024 tokens
- Learning rate: 7×10^{-5}
- LoRA rank: 16, $\alpha = 32$, dropout = 0.05
- Epochs: 1, batch size: 1 (gradient accumulation 8)

We masked the prompt portion of each example so that loss was computed only on the solution tokens. This ensured that the model was optimized for producing step-by-step solutions, not for restating the question.

3.3 Generation and Verification

After fine-tuning, each model was prompted to solve held-out questions from GSM8K. For each solution, we measured:

- Whether the predicted final answer matched the ground truth.
- Whether intermediate steps were judged correct.

Step accuracy was assessed automatically using **ThinkPRM-7B**, a verifier model trained on PRM800K for process supervision (Lightman et al., 2023). The verifier produced step-level correctness scores between 0 and 1, which we averaged across steps.

3.4 Critique Parsing

Verifier outputs were parsed using a custom critique parser. Each reasoning trace was segmented into steps, and numeric correctness scores were extracted when available. When explicit scores were not present, heuristics mapped tokens such as "correct" or "incorrect" to binary values. This process yielded structured results containing:

- The number of steps in each solution.
- Step-level correctness scores.
- Ground-truth vs. predicted final answers.

From these parsed results, we computed two additional metrics:

- The **Gap** between reasoning correctness and final answer accuracy.
- The **Spearman correlation** between per-step correctness and final answer correctness.

3.5 Data Cleaning and Aggregation

Finally, we standardized raw model generations using a cleaning script that truncated outputs at the first "final answer" marker and collapsed redundant newlines. Cleaned generations and parsed verifier outputs were merged into structured JSON files containing final evaluation metrics for each model. These aggregated results formed the basis of our analysis in Section 4.

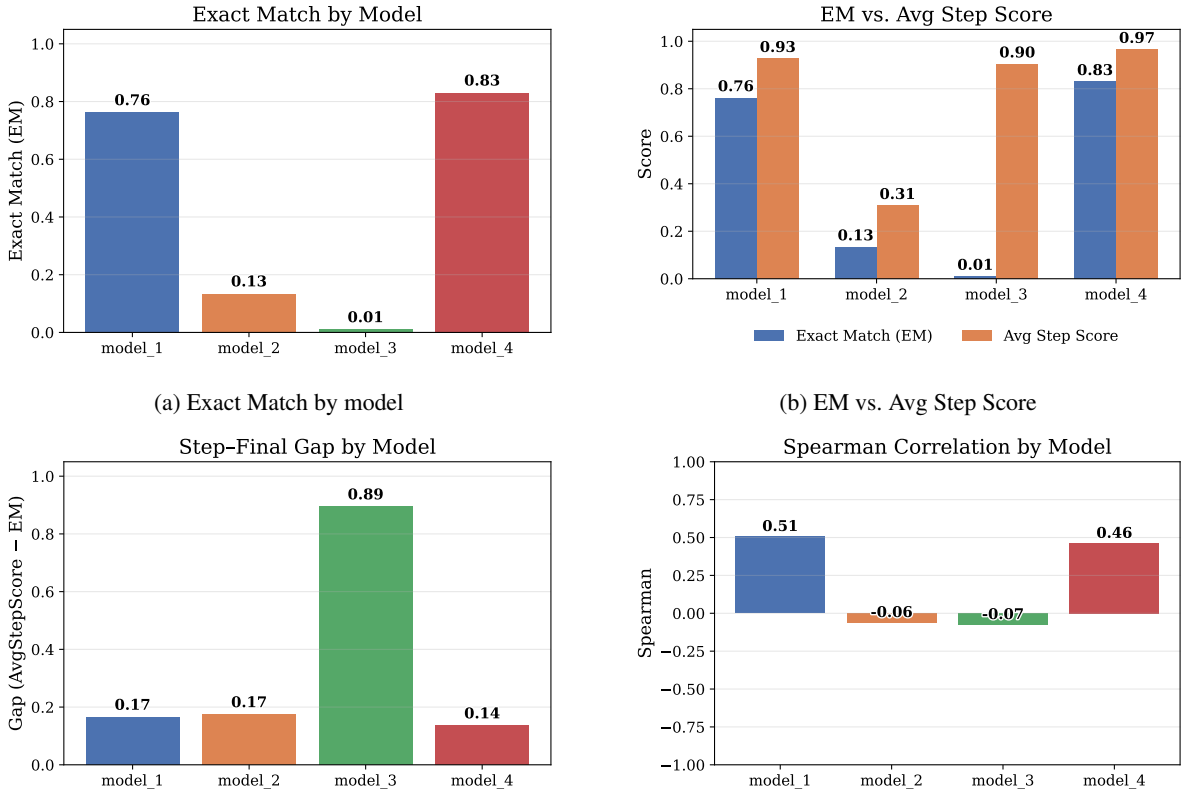
4 Results and Discussion

We evaluate four models: (1) fine-tuned on correct steps and correct answers (CS–CA), (2) fine-tuned on wrong steps with correct answers (WS–CA), (3) fine-tuned on correct steps with wrong answers (CS–WA), and (4) the base model without fine-tuning. Table 1 summarizes the metrics; Figure 1 presents the core comparisons, and Figure 2 shows step-length and selectivity analyses. We used the following metrics:

- **EM (Exact Match)**: Fraction of examples where the predicted final answer exactly matches the gold answer.
- **AvgStepScore**: Average correctness score of intermediate reasoning steps.
- **AllStepsOK**: Proportion of examples where all steps received score greater or equal to 0.5.

Model	N	EM	AvgStepScore	AllStepsOK	Spearman	SelAcc@0.7	Coverage@0.7
model_1	500	0.762	0.927	0.786	0.508	0.794	0.922
model_2	500	0.134	0.308	0.190	-0.060	0.093	0.258
model_3	500	0.010	0.904	0.754	-0.073	0.009	0.888
model_4	500	0.830	0.965	0.912	0.463	0.860	0.960

Table 1: Summary of evaluation metrics across models.



(a) Exact Match by model

(b) EM vs. Avg Step Score

(c) Step–Final gap

(d) Spearman correlation

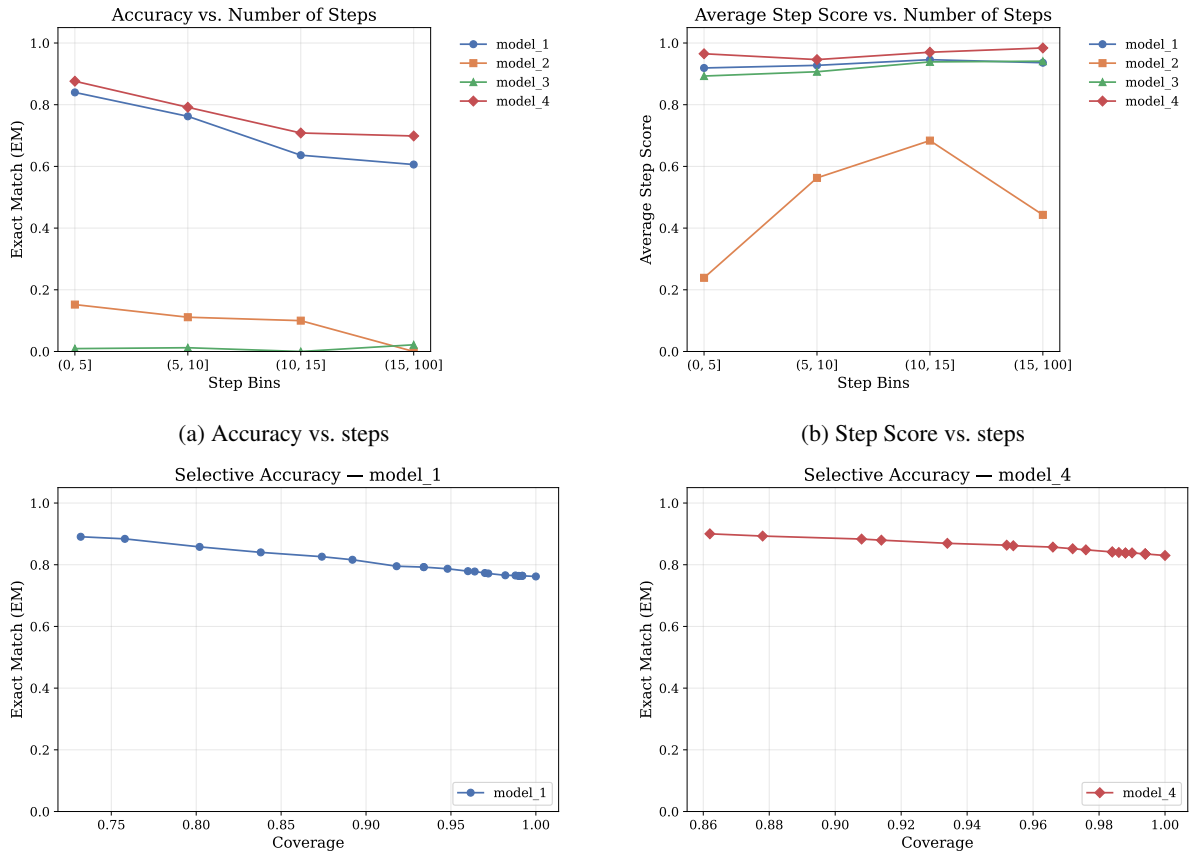
Figure 1: Core evaluation results across models.

- **Spearman**: Rank correlation between average step scores and final correctness (measures alignment).
- **SelAcc@0.7**: Selective accuracy at confidence threshold 0.7 (accuracy when only high-confidence predictions are kept).
- **Coverage@0.7**: Proportion of examples retained at the same threshold.

Overall performance. Model 1 and the base model (Model 4) yield the strongest final-answer accuracy (EM = 0.762/0.830). Models 2 and 3 collapse in EM (0.134/0.010), showing that misaligned training distributions severely degrade reliability.

Reasoning vs. answers. Figure 1 b shows a dissociation in Model 3: very high step correctness (AvgStepScore = 0.904) but nearly zero EM - the model learns to produce plausible steps without correct outcomes when trained on correct-steps/incorrect-answers. Model 2 shows the opposite pathology: low step quality (AvgStepScore = 0.308) and poor EM despite correct answer labels. The base model (Model 4) balances high step fidelity (0.965) and accuracy (0.830).

Alignment. The Step–Final gap in Figure 1 c is small for Models 1/4 and large for Model 3, indicating misalignment. Spearman correlations in Figure 1 d are positive for Models 1/4 (0.508/0.463) but near-zero/negative for Models 2/3 (−0.060/−0.073), evidencing unfaithful reasoning.



(a) Accuracy vs. steps

(b) Step Score vs. steps

(c) Selective Acc. model_1

(d) Selective Acc. model_4

Figure 2: Analysis by step bins and selective prediction.

Step length and selectivity. Figures 2 a–b show that accuracy decreases as chains of thought lengthen (error accumulation), while step scores stay high for most models except Model 2. Selective prediction curves in Figures 2 c–d indicate robust calibration only for Models 1/4 (SelAcc@0.7 $= 0.794/0.860$); corrupted models lack useful confidence.

Takeaway. Models do not inherently use their reasoning steps to derive answers. When steps and answers are aligned (Model 1), reasoning correlates with correctness; when misaligned (Models 2/3), models mimic patterns without genuine causal reliance, supporting the view that chain-of-thought can be superficial rather than a causal reasoning process.

5 Conclusions

We showed that large language models do not automatically use their chain-of-thought to compute outcomes. Training on consistent (step, answer) pairs yields positive reasoning–answer alignment and good calibration (Model 1), whereas misaligned supervision produces unfaithful reasoning or degraded accuracy (Models 2/3).

Key conclusions.

- CoT can act as a *surface pattern*: models may output plausible steps without causally supporting the final answer (large Step–Final gap and negative/near-zero Spearman).
- Misaligned supervision harms both *accuracy* and *calibration*; consistent datasets are necessary for faithful CoT.
- Longer chains increase error risk even for strong models, suggesting limits to naïvely extending CoT length.

6 Future work

While our study provides initial insights into how reasoning steps influence model behavior, several promising directions remain open for exploration.

Consistency regularization. One promising direction is to incorporate explicit regularization terms that penalize inconsistencies between verified step correctness and the final answer during training. By encouraging the model to align the quality of intermediate reasoning with the correctness of the output, we can reduce cases where the model produces plausible steps but incorrect conclusions (or vice versa), leading to more faithful reasoning pipelines.

Verifier-in-the-loop training. Integrating step verifiers directly into the training process could help improve reasoning fidelity. For example, verified correct steps could receive higher gradient weight, while incorrect ones are down-weighted or filtered out. This would guide the model to internalize the logical dependencies between steps and final answers, reinforcing the causal relationship rather than pattern matching.

Data curation. Improving the quality of training data remains essential. Automatically detecting and removing examples with inconsistencies between steps and answers can help reduce spurious correlations. Additionally, curating datasets with shorter and more minimal chains of thought can limit error propagation and make the reasoning process more interpretable and robust.

Causal probes. Finally, designing controlled interventions - such as editing or deleting specific reasoning steps and observing the impact on the final answer - can provide deeper insight into whether the model truly relies on its intermediate reasoning. Such causal probing would enable a more rigorous evaluation of reasoning dependence and inform the development of models that reason in a more human-like and trustworthy manner.

7 Limitations

Our study has several limitations.

- **Benchmark scope.** All experiments use GSM8K (English, grade-school math). Results may not transfer to other domains (commonsense, symbolic reasoning, multi-turn QA), languages, or formats.
- **Model coverage.** We evaluate a single base architecture/size (`Phi-3.4B-mini`) with QLoRA fine-tuning. Trends may change with larger models, different families, or full-parameter training.
- **Data interventions.** The procedures used to create *wrong steps* and *wrong answers* are synthetic and may introduce artifacts (style, token cues, distribution shift) that models could exploit. They may not reflect naturally occurring mistakes.
- **Single prompt/config.** We use one prompting format and a fixed decoding configuration. CoT behavior is known to be prompt- and temperature-sensitive; exploring prompt variants and decoding policies is future work.
- **Evaluation of steps.** Step correctness is computed automatically with a verifier (ThinkPRM). Verifier bias, segmentation errors, or score calibration issues can mislabel steps. We did not include human step-level annotations or multiple verifiers.
- **Alignment metrics.** Our alignment measures (Step-Final gap, Spearman) are correlational and depend on verifier scores; they do not establish causality. We did not run interventional tests at inference time (e.g., editing/deleting steps and observing answer changes).
- **Compute and variance.** Due to budget limits, we use one training run per condition and evaluate on 500 examples. We do not report confidence intervals or multiple random seeds.

These constraints bound the generality of our conclusions. Future work should test multiple model families and sizes, add human-verified step labels, use interventional causal probes, and expand beyond GSM8K to broader reasoning tasks and languages.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, et al. 2021. Training verifiers to solve math word problems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Noah Lanham, Vikas Raunak, et al. 2023. Measuring faithfulness in chain-of-thought reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Victor Lightman, Maxwell Nye, et al. 2023. Prm800k: A process supervision dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Miles Turpin, Julian Michael, Jacob Andreas, et al. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A Prompt Templates

Training (SFT) prompt.

```
You are an expert mathematician. Solve the
following problem step by step,
numbering each step like "Step 1:", "Step 2:",
etc.

Problem:
{question}
```

Inference (generation) prompt.

```
You are an expert mathematician. Solve the
following problem step by step,
numbering each step like "Step 1:", "Step 2:",
etc. Show all reasoning clearly.

question:
{problem}
```

Verifier (ThinkPRM) prompt.

```
You are given a math problem and a proposed
step-by-step solution:

[Math Problem]

{question}

[Solution]

{solution}

Review and critique each step in the proposed
solution to determine whether
each step is correct. For each step, give a score
between 0 and 1
```

B Example Instance (Model 1)

Question.

```
A robe takes 2 bolts of blue fiber and half that
much white fiber.
How many bolts in total does it take?
```

Model-generated solution (Model 1).

```
step 1: The robe takes 2/2=<<2/2=1>>1 bolt of
white fiber.
step 2: So in total it takes 2+1=<<2+1=3>>3 bolts
of fiber.
final answer: 3
```

Verifier critique (excerpt).

```
Step 1: The problem states that a robe takes 2
bolts of blue fiber and half that
much white fiber. Since half of 2 is 1, the
computation 2/2 equals 1 is correct.
The step is [correct].

Step 2: Add blue and white to get the total: 2 +
1 = 3. The step is [correct].

Is the solution correct? Yes. Final answer: 3.
```

C Training Data Examples

Correct steps & correct answer.

```
{"question": "Natalia sold clips to 48 of her
friends in April, and then she sold half as many
clips in May. How many clips did Natalia sell
altogether in April and May?",
"solution": "step 1: Natalia sold 48/2 =
<<48/2=24>>24 clips in May.\nstep 2: Natalia sold
48+24 = <<48+24=72>>72 clips altogether in April
and May.\nfinal answer: 72"}
```

Correct steps & wrong answer.

```
{"question": "Natalia sold clips to 48 of her
friends in April, and then she sold half as many
clips in May. How many clips did Natalia sell
altogether in April and May?",
"solution": "step 1: Natalia sold 48/2 =
<<48/2=24>>24 clips in May.\nstep 2: Natalia sold
48+24 = <<48+24=72>>72 clips altogether in April
and May.\nfinal answer: 117"}
```

Wrong steps & correct answer.

```
{"question": "Natalia sold clips to 48 of her
friends in April, and then she sold half as many
clips in May. How many clips did Natalia sell
altogether in April and May?",
"solution": "step 1: Natalia sold 51-22 =
<<50-25=74>>73 clips altogether in April and
May.\nstep 2: Natalia sold 44/5 = <<52/6=27>>21
clips in May.\nfinal answer: 72"}
```