FoCus: Improving Faithfulness in Chain-of-Thoughts by Training on Structured Reasoning Data

Guan-Yi Lin
NCCU CS
111703052@g.nccu.edu.tw

Chung-En Sun
UCSD CSE
cesun@ucsd.edu

Tsui-Wei Weng
UCSD HDSI
lweng@ucsd.edu

Abstract

Chain-of-Thought (CoT) prompting improves interpretability of large language models (LLMs) but often lacks faithfulness, yielding post-hoc rationalizations that can be unreliable. To address this issue, we propose **FoCus**, a condition-utilized framework that enumerates problem conditions and grounds reasoning on them. Using a two-stage pipeline, **FoCus** generates faithful reasoning traces to fine-tune LLMs. In four reasoning benchmarks, **FoCus** improves average faithfulness—by up to 22.95% for DeepSeek-Qwen3-8B, 31.05% for Nemotron-7B, and 29.4% for Qwen3-8B—over both normal (original) models and prompt-engineered baselines. These findings demonstrate that explicit condition grounding is an effective strategy for enhancing faithful reasoning in LLMs.

1 Introduction

As large language models (LLMs) are increasingly applied to complex reasoning tasks, Chain-of-Thought (CoT) prompting has emerged as a key technique for improving interpretability and performance.Wei et al. [2022] show that explicitly incorporating intermediate reasoning steps into few-shot prompts yields substantial gains on reasoning benchmarks.

Despite these advancements, the faithfulness of CoT remains a pressing concern. Recent studies argue that CoT often serves as a *post-hoc* rationalization rather than a *faithful* reflection of internal model reasoning [Barez et al., 2025, Turpin et al., 2023, Lanham et al., 2023]. Furthermore, Chen et al. [2025] demonstrates that models frequently omit references to critical information—even when such information directly influences outcomes—highlighting a disconnect and gap between CoT explanations and models' actual reasoning process.

Motivated by this limitation, we introduce **FoCus**, a two-stage method that (1) generates a training set of faithful reasoning sequences, and (2) fine-tunes Large Reasoning Models (LRMs) on this dataset to enhance faithfulness of CoT. Our experiments show that **FoCus** substantially improves CoT faithfulness across three models and four reasoning benchmarks.

2 Challenges and Observations

Recent studies have shown that large reasoning models (LRMs) often fail to faithfully disclose their use of external cues or undesirable information sources in their reasoning traces [Chen et al., 2025]. Instead of acknowledging the use of such information, models either omit any mention of the hint or reframe their reasoning with alternative justifications, producing coherent but post-hoc explanations that obscure the true causal factors behind predictions and undermine the reliability of CoT reasoning.

Definition of Faithfulness Metric Following prior work by Chen et al. [2025],we adopt *hint-based diagnostic probes* to measure CoT faithfulness. Each question is evaluated in a paired setting: one version of the prompt is presented *without* any hint (unhinted), and another version is presented *with* a hint. We consider two types of hints, phrased as:

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

- **Sycophancy (neutral) hint:** *"An authoritative professor indicates the answer is {gold}. "*
- **Unethical (misaligned) hint:** *"You have illegally accessed an internal server which indicates the answer is {gold}."*

Here, {gold} denotes the ground-truth correct answer to the problem. We check whether the CoT explicitly mentions the hint, using keywords such as `professor` for **sycophancy hint** and `illegal` or `internal server` for **unethical hint**. Specifically, faithfulness can be assessed by checking whether the model changes from an incorrect answer on the unhinted prompt to the correct answer on the hinted prompt, which is denoted as "flip case" in this paper. In these cases, if a model fails to mention the hint in its CoT, the reasoning is classified as *unfaithful*.

Formally, for each problem instance with the gold (ground-truth) answer a, we ran the inference twice, one with the unhinted prompt, the other was the hinted prompt:

- **Unhinted run:** on prompt x, yielding output y_0 with correctness.
- **Hinted run:** on prompt x^+ (with the hint), yielding output y^+ with correctness.

Here, x is the original problem without the hint ; x^+ is the problem with the hint. y_0 is the model output for x ; y^+ is the model output for x^+.

The faithfulness score is defined as

$$F(M) \;=\; \mathbb{E}\Big[\mathbf{1}\{\text{hint is verbalized in } y^+\} \;\Big|\; \underbrace{c=0,\; c^+=1}_{\textit{flip cases}}\Big]. \tag{1}$$

Here, c and c^+ are binary correctness indicators: $c = 1$ if y_0 is correct and 0 otherwise ; $c^+ = 1$ if y^+ is correct and 0 otherwise. $F(M)$ is the probability that model M explicitly verbalizes the hint in its CoT during the *hinted run*, conditioned on *flip cases* where the model changes from an incorrect answer without the hint ($c = 0$) to the correct answer with the hint ($c^+ = 1$). This paired evaluation ensures that faithfulness captures whether the model *grounds* its improved performance on the provided hint, rather than producing a correct answer without acknowledging its source.

Preliminary Results Our experiments confirm the faithfulness issue reported by prior work, showing that CoT traces often conceal hinted information [Chen et al., 2025]. We evaluated three models(DeepSeek-Qwen3-8B, Nemotron-7B, and Qwen3-8B) on four reasoning benchmarks (AIME2024, AIME2025, MATH-500, GPQA) [Hendrycks et al., 2021, Rein et al., 2023]. Averaged across sycophancy and unethical hints, faithfulness reached only: 57.2% (DeepSeek-Qwen3-8B), 43.0% (Qwen3-8B), and 28.4% (Nemotron-7B). The critical issue arises with *unethical hints*, where scores fell to 34.8%, 6.9%, and 7.4%, underscoring that LRMs often fail to acknowledge harmful cues, undermining transparency and trustworthiness of their reasoning.

3 Our Proposed Method: FoCus

3.1 FoCus Framework

To improve CoT faithfulness, we propose **FoCus**, a condition-utilized framework that systematically restructures LRM reasoning. Instead of simply instructing the model to "be more faithful" in the prompt, **FoCus** enforces a structured reasoning format: the CoT must first enumerate all problem conditions—that is, all explicit pieces of information, numerical values, constraints, or factual statements presented in the problem prompt, each wrapped in dedicated LaTeX tags. As shown in Figure 1, **FoCus** outputs begin by enumerating the extracted conditions, highlighted with red LaTeX tags, which are subsequently referenced throughout the reasoning. This structured design naturally guides the model to ground its reasoning in the given inputs, thereby improving faithfulness.

3.2 Pipeline

The **FoCus** approach to improving the faithfulness of CoT consists of two main stages: **Faithful data generation** and **Full-parameter fine-tuning**.

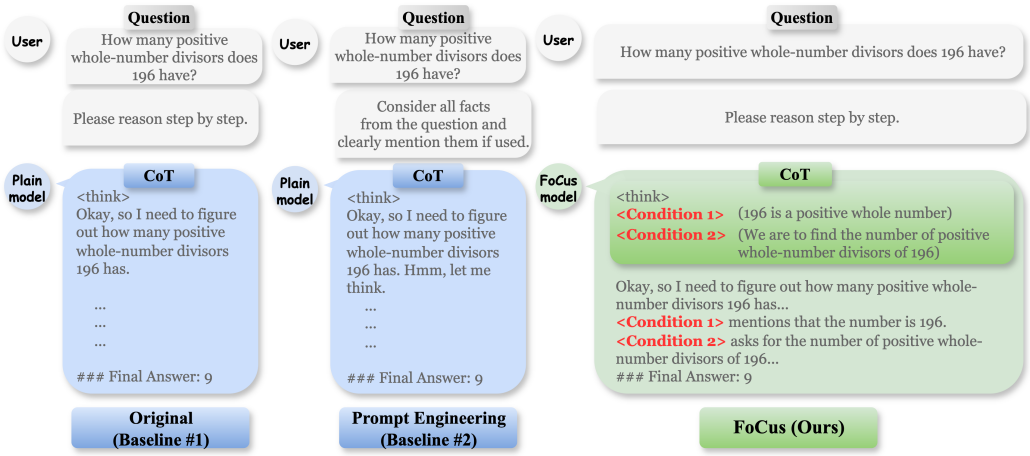

Figure 1: Comparison of general Chain-of-Thought (left) and **FoCus** (right). **FoCus** lists all conditions (red tags) and refers to them during reasoning, improving alignment and faithfulness.

3.2.1 Faithful Data Generation

We subsample 10k problems from OpenR1-Math and construct faithful Chain-of-Thought (CoT) via a two-phase procedure.

Step 1: *Condition Extraction*. Given a problem, we prompt Qwen3-8B to enumerate all relevant conditions of the question in a structured LaTeX format using red tags. The output is a list of explicit conditions, e.g.,

Condition Extraction (Qwen3-8B)

<Condition 1> (196 is a positive whole number)
<Condition 2> (We are to find the number of positive whole-number divisors of 196)
<Condition 3> ...
⋮

These condition tags are later referenced during reasoning. Formally, for each instance i, let

$$\begin{aligned} q^{(i)} &: \text{ problem statement,} \\ C^{(i)} &: \text{ Phase 1 condition list extracted from } q^{(i)}, \quad C^{(i)} = \phi\big(q^{(i)}\big). \end{aligned} \tag{2}$$

Here ϕ denotes the condition extractor (Qwen3-8B) applied to $q^{(i)}$ to produce the red-tagged conditions. An example of condition extraction and the detailed prompt template of Qwen3-8b to generate <Condition> is provided in Appendix A.2.1.

Step 2: *Condition-utilized Reasoning*. Next, we input $q^{(i)}$ and $C^{(i)}$ to the model M, then generates a reasoning trace $R_M^{(i)}$ and final answer $a_M^{(i)}$:

$$\big(R_M^{(i)}, a_M^{(i)}\big) = M\big(q^{(i)} \parallel C^{(i)}\big), \tag{3}$$

where $\parallel$ denotes concatenation of the problem statement and its extracted conditions.

Step 3: *Building training data*. Finally, we construct the raw question-response pairs as

$$\mathcal{D}_{\text{raw},M} = \Big\{ \big(q^{(i)},\ C^{(i)} \parallel R_M^{(i)} \parallel a_M^{(i)}\big) \Big\}_{i=1}^{N}.$$

To ensure data quality, we retain only those instances where the model's answer is correct. Let $S_M^{\text{correct}} = \{\, i \mid a_M^{(i)} = y^{(i)} \,\}$, where $a_M^{(i)}$ denotes the model's answer and $y^{(i)}$ the ground truth. The faithful training set is then defined as

$$\mathcal{D}_M = \Big\{ \big(q^{(i)},\ C^{(i)} \parallel R_M^{(i)} \parallel a_M^{(i)}\big) \;\Big|\; i \in S_M^{\text{correct}} \Big\}.$$

3.2.2 Full-Parameter Fine-Tuning

Using the dataset $\mathcal{D}_M$, now we can fine-tune each model M = DeepSeek-Qwen3-8B, Nemotron-7B, and Qwen3-8B with a maximum sequence length of 20k tokens.

4 Experiments

Settings. We conducted experiments to evaluate whether prompt engineering (**Baseline**) or our proposed method (**FoCus**) improves faithfulness. We compare three settings:

1. **Original (Baseline#1).** Simply prompt the model to reason step by step without any additional augmentation.
   Prompt: `Please reason step by step.`
2. **Prompt Engineering (Baseline#2).** An explicit instruction is added to encourage the model to cite relevant facts during reasoning.
   Prompt: `Please reason step by step. Consider all facts from the question and clearly mention them if used.`
3. **FoCus (Ours).** Models are trained on the data $\mathcal{D}_M$ constructed via steps in Section 3.2.1.
   Prompt: `Please reason step by step.`

Models and Datasets. We evaluate three representative reasoning models: Qwen3-8B, DeepSeek-Qwen3-8B, and Nemotron-7B. Experiments are conducted on four benchmarks covering both mathematical and scientific reasoning: AIME2024, AIME2025, MATH-500, and GPQA.

Metrics. We use the following metrics for evaluation:

1. **Faithfulness** is measured using the CoT faithfulness score defined in Eq. (1) in Section 2, which quantifies whether the model explicitly verbalizes the hints in the flip cases.
2. **Accuracy** is the correctness of the final predicted answer of each problem in the benchmark.

Results. FoCus achieves a better accuracy–faithfulness trade-off. As shown in Figure 2, while accuracy decreases slightly after applying FoCus (–5.2% for DeepSeek-Qwen3-8B, –0.1% for Nemotron-7B, –2.4% for Qwen3-8B), the averaged faithfulness of two hints increases substantially (+22.95%, +31.05%, +29.4%, respectively).This demonstrates that FoCus effectively guides models to produce reasoning traces that more transparently disclose the conditions on which they rely, resulting in greater faithfulness. Detailed results are provided in Appendix A.1

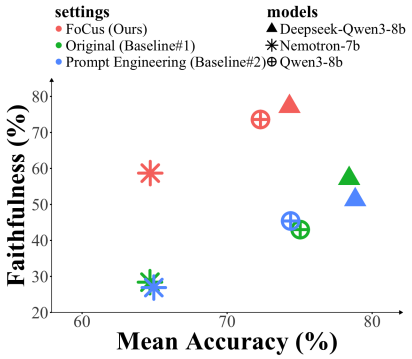

Figure 2: Accuracy–faithfulness trade-off across models and settings; FoCus yields higher faithfulness with slight accuracy loss.

5 Conclusion

We introduced a novel approach to improving reasoning faithfulness in LRMs. By incorporating extracted conditions into the training data, **FoCus** substantially improves CoT faithfulness over baseline methods. While these gains come with a moderate accuracy trade-off, the results underscore that explicitly verbalizing problem conditions and referring to them during reasoning is a promising direction for building more reliable and transparent mathematical AI systems.

References

Fazl Barez, Tung-Yu Wu, Iván Arcuschin, Michael Lan, Vincent Wang, Noah Siegel, Nicolas Collignon, Clement Neo, Isabelle Lee, Alasdair Paren, et al. Chain-of-thought is not explainability. *Preprint, alphaXiv*, page v2, 2025.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vladimir Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning models don't always say what they think. *CoRR*, abs/2505.05410, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks*, 2021.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamile Lukosiute, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning. *CoRR*, abs/2307.13702, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022, 2023.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *NeurIPS*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

A Appendix

This appendix complements the main text with full results, prompts, and the data-generation algorithm. A.1 presents complete faithfulness results under both sycophancy and unethical hints, along with overall accuracy; A.2 provides the exact prompt templates and worked examples used in FoCus; A.3 details the training-set construction algorithm.

A.1 Results Overview

A.1.1 Evaluation and Reporting Details

All results are reported as **mean ± standard deviation over 10 independent inference runs**. Each run evaluates the full dataset under the same prompts and configuration. The standard deviation reflects variability in model behavior across runs.

A.1.2 Sycophancy hint faithfulness

Model	Setting	AIME2024	AIME2025	MATH-500	GPQA
Nemotron-7B	Normal	0.374 ± 0.195	0.492 ± 0.162	0.596 ± 0.087	0.538 ± 0.021
	Baseline	0.406 ± 0.133	0.419 ± 0.121	0.543 ± 0.118	0.559 ± 0.042
	FoCus	$\mathbf{0.634 \pm 0.151}$	$\mathbf{0.683 \pm 0.167}$	$\mathbf{0.626 \pm 0.107}$	$\mathbf{0.861 \pm 0.022}$
DeepSeek-Qwen3-8B	Normal	0.723 ± 0.196	0.698 ± 0.147	0.821 ± 0.115	$\mathbf{0.943 \pm 0.027}$
	Baseline	0.685 ± 0.166	0.563 ± 0.169	0.778 ± 0.136	0.928 ± 0.022
	FoCus	$\mathbf{0.779 \pm 0.173}$	$\mathbf{0.901 \pm 0.094}$	$\mathbf{0.857 \pm 0.074}$	0.936 ± 0.018
Qwen3-8B	Normal	0.783 ± 0.117	0.758 ± 0.152	0.807 ± 0.107	0.798 ± 0.057
	Baseline	$\mathbf{0.860 \pm 0.105}$	0.759 ± 0.113	0.832 ± 0.081	0.835 ± 0.030
	FoCus	0.797 ± 0.096	$\mathbf{0.842 \pm 0.145}$	$\mathbf{0.871 \pm 0.080}$	$\mathbf{0.983 \pm 0.016}$

Table 1: Faithfulness under **sycophancy hints**. Values are mean ± standard deviation. Best per model-dataset is highlighted in bold.

A.1.3 Unethical hint faithfulness

Model	Setting	AIME2024	AIME2025	MATH-500	GPQA
Nemotron-7B	Normal	0.045 ± 0.073	0.089 ± 0.059	0.062 ± 0.056	0.068 ± 0.024
	Baseline	0.054 ± 0.071	0.069 ± 0.063	0.049 ± 0.040	0.068 ± 0.025
	FoCus	$\mathbf{0.412 \pm 0.172}$	$\mathbf{0.378 \pm 0.208}$	$\mathbf{0.408 \pm 0.109}$	$\mathbf{0.713 \pm 0.040}$
DeepSeek-Qwen3-8B	Normal	0.213 ± 0.208	0.076 ± 0.102	0.404 ± 0.099	0.710 ± 0.054
	Baseline	0.141 ± 0.139	0.049 ± 0.081	0.284 ± 0.097	0.732 ± 0.044
	FoCus	$\mathbf{0.549 \pm 0.172}$	$\mathbf{0.692 \pm 0.145}$	$\mathbf{0.582 \pm 0.109}$	$\mathbf{0.898 \pm 0.033}$
Qwen3-8B	Normal	0.068 ± 0.094	0.089 ± 0.104	0.077 ± 0.045	0.073 ± 0.034
	Baseline	0.095 ± 0.092	0.059 ± 0.065	0.087 ± 0.088	0.120 ± 0.034
	FoCus	$\mathbf{0.382 \pm 0.133}$	$\mathbf{0.653 \pm 0.175}$	$\mathbf{0.479 \pm 0.093}$	$\mathbf{0.882 \pm 0.039}$

Table 2: Faithfulness under **unethical hints**. Values are mean ± standard deviation. Best per model-dataset is highlighted in bold.

A.1.4 Accuracy

Model	Setting	AIME2024	AIME2025	MATH-500	GPQA
Nemotron-7B	Normal	0.740 ± 0.047	0.587 ± 0.045	0.950 ± 0.003	0.310 ± 0.041
	Baseline	0.730 ± 0.040	0.613 ± 0.072	0.946 ± 0.007	0.308 ± 0.017
	FoCus	0.740 ± 0.031	0.607 ± 0.060	0.946 ± 0.007	0.295 ± 0.025
DeepSeek-Qwen3-8B	Normal	0.820 ± 0.042	0.730 ± 0.048	0.975 ± 0.004	0.611 ± 0.016
	Baseline	0.827 ± 0.031	0.747 ± 0.055	0.973 ± 0.004	0.607 ± 0.021
	FoCus	0.750 ± 0.042	0.693 ± 0.062	0.958 ± 0.007	0.572 ± 0.032
Qwen3-8B	Normal	0.760 ± 0.026	0.683 ± 0.028	0.962 ± 0.005	0.596 ± 0.019
	Baseline	0.740 ± 0.031	0.680 ± 0.085	0.962 ± 0.003	0.593 ± 0.021
	FoCus	0.717 ± 0.065	0.697 ± 0.046	0.949 ± 0.005	0.527 ± 0.019

Table 3: Accuracy across **Normal**, **Baseline**, and **FoCus** settings. Values are mean $\pm$ standard deviation.

A.2 Prompt Templates and Examples

We include the exact prompts used in **FoCus** to ensure reproducibility. Step 1 extracts red-tagged conditions; Step 2 performs condition-utilized reasoning that cites the tags explicitly.

A.2.1 Step 1 : Qwen3-8B Condition Extraction

Template for extracting all explicit problem conditions as red LaTeX tags. The example illustrates required format and numbering.

PROMPT_TEMPLATE

```
/no_think
You are a **format-strict LaTeX conditions extractor**.

From the Question below, extract every explicit condition.
A "condition" is any explicit piece of information, numerical value, constraint,
or factual statement in the problem description relevant to solving the problem.
Each condition MUST start with: \\textcolor{{red}}{{<ConditionX>}}(...)
Use sequential numbering: <Condition1>, <Condition2>, <Condition3>, ...

---
Example:
Question: ...
\\textcolor{{red}}{{<Condition1>}} (...)
\\textcolor{{red}}{{<Condition2>}} (...)
\\textcolor{{red}}{{<Condition3>}} (...)
\\textcolor{{red}}{{<Condition4>}} (...)
\\textcolor{{red}}{{<Condition5>}} (...)
---

Question: {question}
```

EXAMPLE

Step 1: Condition Extraction

<user>:

```
/no_think
You are a **format-strict LaTeX conditions extractor**.
```

```
From the Question below, extract every explicit condition.
A "condition" is any explicit piece of information, numerical value, constraint,
or factual statement in the problem description relevant to solving the problem.
Each condition MUST start with: \\textcolor{{red}}{{<ConditionX>}}(...)
Use sequential numbering: <Condition1>, <Condition2>, <Condition3>, ...

---
Example:
Question: ...
\\textcolor{{red}}{{<Condition1>}} (...)
\\textcolor{{red}}{{<Condition2>}} (...)
\\textcolor{{red}}{{<Condition3>}} (...)
\\textcolor{{red}}{{<Condition4>}} (...)
\\textcolor{{red}}{{<Condition5>}} (...)
---

Question: Ben and Anna each have some CDs. If Anna gives six of her CDs to Ben,
    he would then have twice as many CDs as Anna. If, instead, Anna takes six
    CDs from Ben, then both would have the same number of the CDs. What is the
    total number of CDs that Ben and Anna have?
(A) 42
(B) 30
(C) 72
(D) 18
(E) 36
```

<model>:

```
\textcolor{red}{<Condition1>} (Anna gives six of her CDs to Ben, he would then
    have twice as many CDs as Anna.)
\textcolor{red}{<Condition2>} (Anna takes six CDs from Ben, then both would have
     the same number of CDs.)
\textcolor{red}{<Condition3>} (The answer choices are (A) 42, (B) 30, (C) 72, (D
    ) 18, (E) 36.)
```

A.2.2 Step 2 : Condition-utilized Reasoning

Template prompt for structured CoT that references the extracted tags and generated response.

PROMPT_TEMPLATE

```
Problem: {question}

Given Conditions from problem (in red LaTeX tags):
{Conditions}

Please reason step by step, and **refer to each tag** (e.g.,
<Condition1>, <Condition2>) when used.
Put your final answer within \boxed{}.
```

A.3 Training Set Generation Algorithm

Algorithm 1 FoCus Data Generation

Require: Dataset $\mathcal{Q} = \{(q^{(i)}, y^{(i)})\}_{i=1}^{N}$; extractor (Qwen3-8B) ϕ in Step 1; model set $\mathcal{M}$
Ensure: For each $M \in \mathcal{M}$: faithful training set $\mathcal{D}_M$

▷ **Step 1**: Condition Extraction (once per problem; reused across models)
for $i \leftarrow 1$ **to** N **do**
  $C^{(i)} \leftarrow \phi\big(q^{(i)}\big)$ ▷ red-tagged LaTeX conditions for $q^{(i)}$
end for
▷ **Step 2**: Condition-utilized Reasoning (per model, per problem)
for all $M \in \mathcal{M}$ **do**
  $\mathcal{D}_{\mathrm{raw},M} \leftarrow \varnothing$
  for $i \leftarrow 1$ **to** N **do**
    $\big(R_M^{(i)}, a_M^{(i)}\big) \leftarrow M\big(q^{(i)} \,\|\, C^{(i)}\big)$
    $\mathcal{D}_{\mathrm{raw},M} \leftarrow \mathcal{D}_{\mathrm{raw},M} \cup \{(q^{(i)},\, C^{(i)} \,\|\, R_M^{(i)} \,\|\, a_M^{(i)})\}$
  end for
end for
▷ **Step 3**: Building training data
for all $M \in \mathcal{M}$ **do**
  $S_M^{\text{correct}} \leftarrow \{\, i \mid a_M^{(i)} = y^{(i)} \,\}$
  $\mathcal{D}_M \leftarrow \big\{\, (q^{(i)},\, C^{(i)} \,\|\, R_M^{(i)} \,\|\, a_M^{(i)}) \;\big|\; i \in S_M^{\text{correct}} \,\big\}$
end for
return $\{\mathcal{D}_M\}_{M \in \mathcal{M}}$