Ask, Condition or Abstain: Reinforcement Learning for Missing-Premise Reasoning

Anonymous ACL submission

Abstract

Answer-only reinforcement learning (RL) trains reasoning models to solve fully specified problems, but many realistic queries omit a premise needed for a unique answer. In this setting, the useful response is not always refusal: the model should ask for the missing premise, condition its answer on the unknown quantity, or abstain when no informative conditional response is available. We present *Ask-Condition-Abstain Reinforcement Learning* (ACA-RL), a data-augmented RL framework for this setting. Its reasoning-graph-guided pipeline converts well-posed problems into missing-premise training instances with localized gap annotations; ACA-RL then trains on these instances with a structured reward over five observable response behaviors. We also introduce the *Missing-Premise Benchmark* (MPB), a 274-instance human-verified benchmark spanning mathematical, logical, and real-world word problems. Across Qwen3 and Llama models, ACA-RL consistently improves on MPB while preserving competitive performance on well-posed reasoning tasks. Together with the released code, MPB, and training data, this work supports a new mission for NLP evaluation: measuring whether models can recognize when a task is underdetermined and handle uncertainty, not only whether they can answer fully specified questions.

1 Introduction

Modern reasoning models are increasingly optimized by reinforcement learning on problems with verifiable final answers. This setting is productive because the reward is clear: solve the problem and match the expected answer. It is also incomplete. A user can omit a rate, leave a relation ambiguous, narrow a definition outside its usable domain, or ask for a value that cannot be identified from the stated premises. In those cases, a model rewarded primarily for definite answers may still produce a confident solution even when the question no longer determines one (Kalai et al., 2025; Ouyang, 2025; Sun et al., 2024).

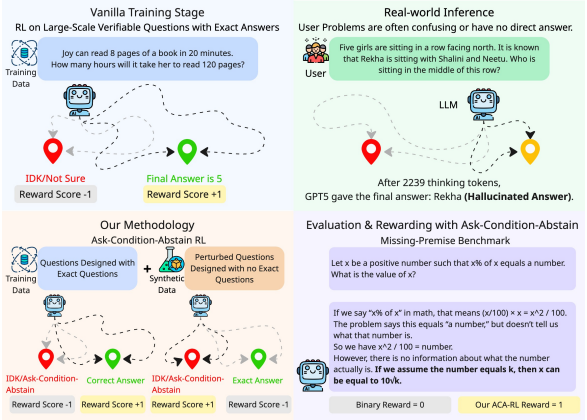

Figure 1: Motivation and framework overview. Answer-only RL trains reasoning models on fully specified questions with verifiable answers, but real user queries often omit critical premises. ACA-RL augments reasoning data with missing-premise variants and uses a structured behavioral reward to prefer asking, conditioning, or abstaining. MPB measures whether responses merely refuse or expose the missing premise in a useful form.

The failure mode is not solved by replacing every uncertain response with "I don't know." Abstention is safer than hallucination, but it is often less helpful than explaining the missing variable, giving a conditional expression, or asking the user for the needed premise. We study this response space as a missing-premise reasoning problem: given a query that looks like a standard reasoning problem but lacks information required for a unique answer, the model should decide whether to ask, condition, or abstain instead of fabricating a value.

This framing separates our goal from retrieval and inference-time screening. Retrieval can supply external facts but does not teach the base model to notice underdetermined prompts; screen-then-answer prompting can flag missing premises but depends on an extra inference-time step. We in-

stead learn the missing-premise response policy within the model itself. Our comparison with LM Introspection shows that prompting alone is insufficient for making a model reliably ask, condition, or abstain under missing premises.

We propose *Ask-Condition-Abstain Reinforcement Learning* (ACA-RL), a data-augmented RL framework for learning missing-premise response policies. ACA-RL first uses our reasoning-graph-guided pipeline to decompose well-posed problems, perturb critical premises, and create 120K natural missing-premise training instances with localized gap annotations. It then optimizes a structured reward over five behaviors: silent hallucination, explicit assumption, abstention, conditional formulation, and active elicitation.

We evaluate this behavior with the *Missing-Premise Benchmark* (MPB), a 274-instance human-verified benchmark spanning six perturbation types across mathematical, logical, and real-world word problems. MPB maps responses to the same five-category taxonomy and reports an averaged Behavior Score, a rubric-aligned behavioral metric rather than a calibrated uncertainty measure.

Our experiments show that ACA-RL improves MPB and third-party unanswerable benchmark scores over Vanilla PPO, IDK-RL, and LM Introspection while remaining competitive on well-posed reasoning. The gains do not reduce to generic refusal: abstention appears early, while conditional formulation and active elicitation improve more gradually, suggesting a harder response policy beyond IDK. More broadly, the work reframes progress in reasoning models as the ability to recognize when a task is underdetermined and respond constructively, not only as accuracy on fully specified tasks. By teaching models to expose missing premises rather than guess, ACA-RL can support safer reasoning assistants, tutoring systems, and agentic workflows that must decide when to ask users or tools for missing information.

Our contributions can be summarized as follows:

- We formulate missing-premise reasoning as an answer-only RL extension where useful responses ask, condition, or abstain.
- We build a reasoning-graph-guided pipeline to synthesize 120K gap-annotated missing-premise instances for ACA-RL training with structured rewards.
- We introduce the human-verified MPB benchmark and show that ACA-RL improves missing-premise behavior while preserving competitive standard reasoning performance.

2 Preliminary

A missing-premise problem resembles a well-posed reasoning task but lacks at least one premise needed for a unique answer. Let s_0 denote a well-posed source problem and s' its perturbed version with gap annotation a_{gap}. A useful response should not silently fabricate the missing information; it should abstain, formulate the answer conditionally, or ask for the missing premise.

We evaluate terminal response behavior rather than probability calibration. Responses are categorized as silent hallucination, explicit assumption, abstention, conditional formulation, or active elicitation, with ask/condition behaviors preferred to generic refusal and unsupported definite answers penalized. MPB instantiates this taxonomy as a behavioral evaluation protocol: 274 human-verified instances across six perturbation types measure whether a model exposes, parameterizes, or requests missing information rather than merely answering or refusing.

3 Missing-Premise Data Construction

ACA-RL requires examples where the original reasoning structure is known and the missing premise can be localized. We build missing-premise data by perturbing well-posed reasoning problems rather than collecting arbitrary ambiguous queries. This treats missing-premise data as a controlled resource for studying robust behavior, not as arbitrary synthetic corruption. The training set contains 120K generated instances, while MPB is constructed from a separate held-out candidate pool and receives additional human expert verification.

3.1 Reasoning-Graph-Guided Missing-Premise Synthesis

To approximate missing-premise cases in a controlled way, we develop a data synthesis pipeline that transforms well-posed problems into plausibly underspecified counterparts. The generated problems contain specific logical gaps rather than random corruption, so the model must detect which premise is missing. The pipeline operates in three stages, and the prompts are shown in Appendix F:

Deconstruction and Reasoning Graph Generation. For a well-posed problem $s_0 \in \mathcal{D}_G$, we

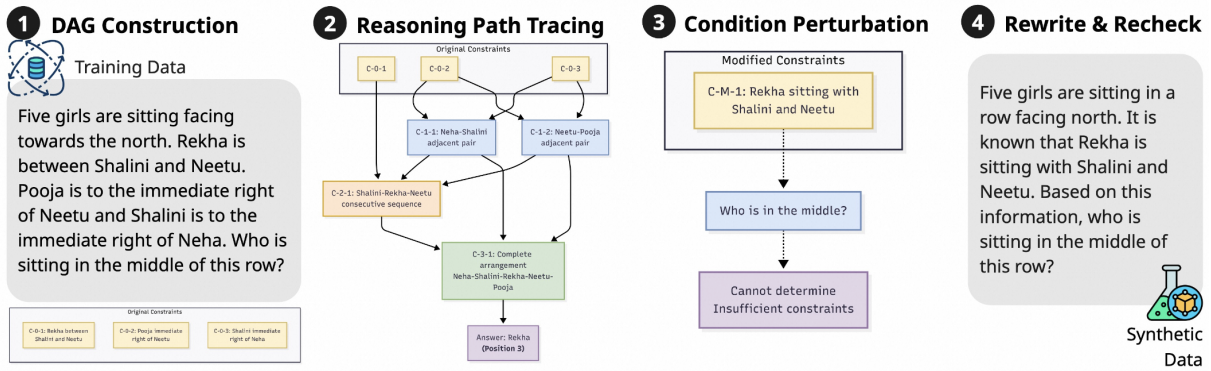

Figure 2: Reasoning-graph-guided missing-premise synthesis pipeline. (**1**) Construct a directed acyclic reasoning graph from a well-posed problem; (**2**) trace the solution path to identify critical constraints; (**3**) perturb key conditions to create a logically underspecified variant; (**4**) rewrite and verify the instance to ensure plausibility and unanswerability. The pipeline generates controlled synthetic data for training missing-premise response behavior.

first parse it into its constituent components: background, conditions $\{C_0\}$, and question. Concurrently, we prompt the model to generate a step-by-step reasoning path, which we structure as a directed acyclic graph (DAG). This graph, or solving tree, makes the dependencies between initial conditions and the final answer explicit. One detailed example of our DAG is shown in Appendix C.

Surgical Condition Perturbation. We then use the reasoning graph to identify a critical condition c on the solution path. A perturbation method is uniformly sampled from our *Conditional Breaking* strategies and applied to c, producing a modified condition c'. Replacing c with c' in the problem statement, we create an underspecified problem s' with a single, well-defined informational gap. This process yields the pair (s', a_{gap}), where a_{gap} documents the precise nature of the induced missing premise. Inspired by (Sun et al., 2024), we formalize the perturbation methods shown in Table 8.

Rewrite & Recheck. Finally, the perturbed conditions are recomposed with the original background and question into a fluent word problem. The resulting problem s' appears fully specified at first glance, mirroring real-world imperfect information. We use an LLM-based quality filter to assess reasoning-graph correctness and missing-premise unanswerability, discarding samples that fail either check. To validate the unanswerability component, human experts annotate 556 held-out candidate instances; against these labels, the filter reaches 93.0% accuracy, 94.4% precision, 91.4% recall, and 92.9% F1. We use this filter for scalable training-data construction, while MPB is selected from a separate held-out pool and verified by humans.

This pipeline yields $\mathcal{D}_{\text{ACA}}$, a large-scale collection of (s', a_{gap}) pairs in which each s' is a carefully constructed, logically underspecified problem and a_{gap} documents the nature of its missing premise. By preserving the original reasoning structure while surgically removing or altering critical constraints, the dataset provides controllable scenarios for training and evaluating uncertainty-aware responses. The explicit annotation of informational gaps enables fine-grained reward shaping and benchmarking of missing-premise behavior.

4 Ask-Condition-Abstain RL

The desired behavior is broader than simple uncertainty flagging. Instead of only outputting "I don't know" (IDK), a model can ask for the missing premise, condition its answer on an unknown variable, or abstain when neither action is useful. We therefore optimize behavioral robustness under missing premises rather than calibrated probability estimates. The evaluation target is whether the generated response avoids hallucination and communicates the missing premise in a useful form.

ACA-RL operationalizes this response policy with a behavioral reward model that scores both uncertainty detection and localization of the missing premise. Higher rewards are allocated to Conditional Formulation and Active Elicitation, while Abstention remains a positive but lower-valued fallback. The training reward and MPB evaluation use the same behavior taxonomy but separate instances: ACA-RL is optimized on the 120K training set, while MPB is a held-out benchmark for measuring

whether the trained policy produces the preferred response categories.

To provide a practical training signal for this objective gap, we design a structured reward function, R_{ACA}. Instead of a binary answer/refusal signal, R_{ACA} provides a fine-grained categorical signal across a spectrum of response behaviors. It guides the policy away from unsupported answers and toward more explicit engagement with informational gaps.

A Partition of the Trajectory Space. We first partition the space of all possible response trajectories, $\mathcal{T}$, into disjoint sets based on the terminal reasoning behavior exhibited by a trajectory τ. This categorization is performed by a behavior classifier, which implements a classification function, $\text{Behav}(\tau) \rightarrow \{\text{SH, EA, Abs, Cond, Elicit}\}$. The behavioral categories are:

- **Silent Hallucination ($\mathcal{T}_{\text{SH}}$):** Trajectories that produce a definite numerical answer by fabricating information without acknowledgment.
- **Explicit Assumption ($\mathcal{T}_{\text{EA}}$):** Trajectories that produce a definite answer but explicitly state the non-grounded assumption made.
- **Abstention ($\mathcal{T}_{\text{Abs}}$):** Trajectories that correctly identify the problem as underspecified and refuse to provide a definite answer.
- **Conditional Formulation ($\mathcal{T}_{\text{Cond}}$):** Trajectories that represent the missing information with a variable and provide a final answer as a formula.
- **Active Elicitation ($\mathcal{T}_{\text{Elicit}}$):** Trajectories that proactively ask a clarifying question to resolve the informational gap.

These sets form a partition of the trajectory space: $\mathcal{T} = \mathcal{T}_{\text{SH}} \cup \mathcal{T}_{\text{EA}} \cup \mathcal{T}_{\text{Abs}} \cup \mathcal{T}_{\text{Cond}} \cup \mathcal{T}_{\text{Elicit}}$.

The Reward Value Function. We then define a value function, $V : \{\text{SH, EA, Abs, Cond, Elicit}\} \rightarrow \mathbb{R}$, that assigns a scalar reward to each behavioral category, reflecting our defined preference hierarchy. The reward for any given trajectory τ is thus determined by its classification:

$$R_{\text{ACA}}(\tau|s') = V(\text{Behav}(\tau)) \tag{1}$$

The value function $V(b)$ for a behavior b is defined as:

$$V(b) = \begin{cases} 1.0 & b = \text{Elicit}, \\ 0.6 & b = \text{Cond}, \\ 0.3 & b = \text{Abs}, \\ -0.3 & b = \text{EA}, \\ -1.0 & b = \text{SH}. \end{cases}$$

The reward values encode a preference hierarchy rather than fitted calibration weights:

$$\text{Elicit} \succ \text{Cond} \succ \text{Abs} \succ \text{EA} \succ \text{SH}.$$

The acceptable behaviors receive positive rewards, while unsupported assumptions and silent hallucinations receive negative rewards. The highest reward is reserved for active elicitation because it most directly moves the interaction toward acquiring the missing information. Conditional formulation receives the next-highest reward because it makes the unknown variable explicit and preserves useful reasoning without fabricating a value. Abstention remains positive as a safe fallback, but its lower value discourages the model from stopping at generic refusal when it can provide a more informative response.

The ACA-RL Objective. With this formal reward structure, we define the ACA-RL objective, J_{ACA}, as the expected value over the distribution of underspecified problems under current policy trajectories:

$$J_{\text{ACA}}(\theta) = \mathbb{E}_{s' \sim \mathcal{D}_{\text{ACA}}} \left[\mathbb{E}_{\tau \sim \pi_\theta(\cdot|s')} [V(\text{Behav}(\tau))] \right] \tag{2}$$

Optimizing J_{ACA} gives the policy an explicit training signal for behaviors that are absent from answer-only supervision. The value function $V(b)$ shifts probability mass away from low-value behaviors such as hallucination ($\mathcal{T}_{\text{SH}}$) and toward higher-value missing-premise responses such as conditional formulation and active elicitation. Thus, J_{ACA} is a practical behavioral proxy for training the model to navigate uncertainty rather than merely replicating fully specified answer paths.

5 Experiments: Missing-Premise Behavior and General Reasoning

This section evaluates ACA-RL on missing-premise robustness, third-party unanswerable benchmarks, and well-posed reasoning checks. We then study data source, mixture ratio, training steps, and data size to characterize when the method improves the reported Behavior Scores and where it trades off against standard reasoning performance.

5.1 Experimental Setup

Details about benchmarks, evaluation methods, and training settings can be found in Appendix E.1. We compare ACA-RL against a suite of strong baselines representing different training paradigms:

- **Cold-start SFT**: The cold-start model without any RL fine-tuning. This serves as the base checkpoint for the Qwen experiments.
- **Vanilla PPO**: A standard verifier-based RL approach (Schulman et al., 2017) trained via PPO *only* on our set of answerable, well-posed problems, rewarding correct final answers. This baseline represents answer-only RL on fully specified problems.
- **IDK-RL** (Song et al., 2025): A baseline trained to explicitly refuse to answer. It is fine-tuned on a mix of answerable and unanswerable questions, with a binary reward for correctly solving the former and outputting IDK for the latter.
- **ACA-RL (Ours)**: Our proposed framework, trained on a curated set of answerable questions in which approximately 30% of the instances have been transformed into missing-premise versions via our reasoning-graph-guided condition perturbation pipeline. The structured reward function R_{ACA} defined in Section 4 explicitly encourages the policy to ask, condition, or abstain on these underspecified problems.

Evaluation Protocol. The 120K ACA-RL instances are used for training, while MPB is a separate 274-instance benchmark selected from a held-out candidate pool and verified for missing-premise validity. For automatic behavior scoring, we adopt GPT-5 as the judge and map each response to the categories in Section 4; Behavior Scores are arithmetic means of the resulting discrete category scores. The same category definitions are used for training-time reward assignment and benchmark scoring, so the results should be read as rubric-aligned behavioral scores on held-out instances. We also report UMWP and SUM as third-party unanswerable benchmarks and standard well-posed reasoning benchmarks to check whether missing-premise training harms ordinary problem solving.

5.2 Main Results

Table 1 presents the main results of our experiments. As shown in Table 1, ACA-RL improves missing-premise robustness across all three model groups. On Qwen3-8B, ACA-RL obtains an MPB Behavior Score of 51.73, compared with 8.66 for Vanilla PPO and 48.72 for IDK-RL. The large gap against Vanilla PPO shows that answer-only RL is poorly aligned with missing-premise behavior. The smaller but consistent gap against IDK-RL is the more relevant comparison: IDK-RL learns conservative refusal, while ACA-RL shifts some responses toward conditional formulation and active elicitation under the same rubric. ACA-RL also keeps general reasoning performance competitive. On Qwen3-8B, the method reduces the drop from Vanilla PPO on GSM8K and AIME'24 compared with IDK-RL, while remaining close on MATH-500. The results therefore support a narrower conclusion: missing-premise training can improve rubric-aligned behavioral robustness while preserving much of the capability learned from well-posed reasoning tasks.

To further check whether the missing-premise reward induces over-refusal on well-posed tasks, we evaluate Qwen3-8B on LiveBench (White et al., 2024) and SciBench (Wang et al., 2023). Table 2 shows that ACA-RL matches the Vanilla PPO/R-LVR average on LiveBench (59.0 vs. 59.0) and remains competitive on SciBench (53.7 vs. 56.0). This additional evaluation suggests that ACA-RL can improve missing-premise behavior without broadly degrading ordinary problem-solving ability on these benchmarks.

6 Analysis: Data and Behavior Trade-offs

We use the analysis to ask what kind of data and optimization are needed for missing-premise behavior, rather than treating MPB as only another leaderboard. The results distinguish easy-to-learn refusal from more informative ask/condition behavior and expose the trade-off between robustness and standard reasoning.

Data Source for Missing-Premise Behavior. Table 3 compares three sources of missing-premise training instances under a fixed data budget: Treecut-style perturbations, SUM-derived unanswerable data, and our reasoning-graph-guided synthesis. Our source obtains the highest MPB Behavior Score in this comparison while maintaining competitive GSM8K and AIME'24 performance, suggesting that the synthesis procedure contributes beyond simply adding unanswerable examples.

Table 1: Performance across missing-premise and general reasoning benchmarks for different model architectures and scales. Robustness is measured by MPB and by the same ask/condition/abstain Behavior Score on two unanswerable benchmarks (**UMWP**, **SUM**); general reasoning is measured by Pass@1 on **GSM8K**, **MATH-500**, and **AIME'24** (averaged over 8 runs). Red values indicate the performance change relative to the strongest RL baseline within each model group.

Method	Missing-Premise	Unanswerable		Mathematics		
	MPB	UMWP	SUM	GSM8K	MATH-500	AIME'24
Qwen3-8B (Reasoning Model)						
Cold-start SFT	22.81	33.53	20.51	83.69	75.20	37.08
Vanilla PPO	8.66	8.84	9.41	93.85	93.40	64.58
IDK-RL	48.72	45.74	42.95	88.02 (-5.83)	92.40 (-1.00)	63.33 (-1.25)
ACA-RL (Ours)	**51.73**	**51.50**	**46.91**	91.50 (-2.35)	92.60 (-0.80)	64.16 (-0.42)
Qwen3-14B (Reasoning Model)						
Cold-start SFT	29.20	43.90	28.08	88.02	87.80	43.33
Vanilla PPO	5.29	10.38	6.16	94.84	93.60	69.17
IDK-RL	49.43	52.78	41.97	89.72 (-5.12)	91.20 (-2.40)	67.28 (-1.89)
ACA-RL (Ours)	**57.20**	**56.94**	**46.56**	90.60 (-4.24)	92.80 (-0.80)	69.58 (+0.41)
Llama3.1-8B (Instruct Model)						
Instruct	3.28	12.38	5.36	80.36	46.80	7.92
Vanilla PPO	6.18	24.73	7.13	84.15	43.20	3.33
IDK-RL	51.29	54.98	44.97	83.28 (-0.87)	39.00 (-4.20)	4.28 (+0.95)
ACA-RL (Ours)	**53.56**	**55.25**	**51.58**	81.43 (-2.72)	41.60 (-1.60)	4.17 (+0.84)

Table 2: General capability check on well-posed benchmarks for Qwen3-8B. ACA-RL preserves the LiveBench average and remains competitive on SciBench, indicating that missing-premise training does not collapse into over-refusal on ordinary tasks.

Method	LiveBench Avg.	SciBench Overall
Cold-start SFT	51.9	47.0
Vanilla PPO/RLVR	**59.0**	**56.0**
ACA-RL (Ours)	**59.0**	53.7

Table 3: Data-source ablation under the same data budget. The reasoning-graph-guided synthesis gives the highest MPB Behavior Score while maintaining competitive GSM8K and AIME'24 performance.

Source	MPB	GSM8K	AIME'24
Treecut	15.41	94.01	67.50
SUM	47.35	91.05	59.58
Ours	51.73	91.50	64.16

Table 4: Training on 10K samples over 100 steps shows the trade-off between missing-premise robustness and standard reasoning as the missing-premise ratio changes. We use 30% as the default because it preserves substantially more general reasoning performance than 50% while improving MPB over 10%.

Portion	MPB	GSM8K	AIME'24
50%	53.19	82.10	52.56
30%	43.79	90.14	60.83
10%	28.28	92.49	63.33

Portion of Missing-Premise Questions. We investigate how the mixture of answerable and unanswerable problems affects performance. We train variants of ACA-RL with different missing-premise data portions: 10%, 30% (our default), 50%.

The variants in Table 4 are trained on 10K samples for 100 steps. Results indicate a trade-off between missing-premise robustness and standard reasoning. A higher portion (50%) gives the highest MPB Behavior Score but lower general reasoning scores, while a lower portion (10%) behaves more like Vanilla PPO, retaining standard reasoning performance but learning weaker missing-premise behavior. We use 30% as a practical balance between these two objectives.

From Abstention to Ask/Condition Behavior. The Behavior Score evaluates more behaviors than the IDK score because it also rewards conditional formulation and active elicitation. Table 5 shows that IDK-RL obtains the highest refusal rates, while ACA-RL obtains higher Behavior Scores with slightly lower but still competitive IDK scores. This pattern suggests that ACA-RL shifts some responses from generic refusal toward more informative missing-premise behavior.

Table 5: Comparison between Behavior Score and traditional IDK score. IDK score measures conservative abstention, while Behavior Score additionally rewards responses that use available information or ask for missing premises. ACA-RL improves Behavior Score while retaining competitive IDK behavior.

Method	UMWP		SUM	
	Behavior Score	IDK Score	Behavior Score	IDK Score
Cold-start SFT	33.53	72.57	20.51	45.77
Vanilla PPO	8.84	74.11	9.41	52.11
IDK-RL	45.74	**95.23**	42.95	**91.19**
ACA-RL	**51.50**	91.30	**46.91**	84.85

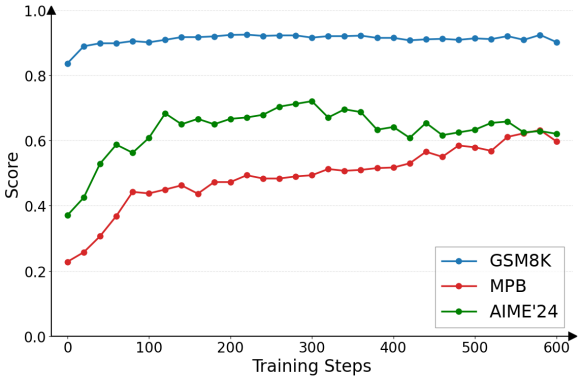

Figure 3: Effect of ACA-RL training steps. Training on our synthetic missing-premise data increases MPB scores in this run while keeping the reported general benchmark scores relatively stable.

Cold-start SFT. Figure 5 shows that Cold-start SFT improves ACA-RL's learning efficiency on MPB across the training trajectory. Without this initialization, the model learns missing-premise behavior more slowly and reaches a lower final score under the same training budget. This suggests that a supervised warm start provides a useful behavioral prior, while RL is still needed to further refine the ask/condition/abstain policy.

Training Steps. As shown in Figure 3, while the model's performance on general reasoning benchmarks improves and then plateaus, it exhibits sustained growth on MPB. This suggests that additional training steps can improve the measured missing-premise behavior in this setting while maintaining stable general performance. Furthermore, Figure 4 shows that conditional formulation and elicitation gradually displace some IDK responses as training progresses.

Comparison with Training-Free Methods. We compare ACA-RL with LM Introspection (Yona et al., 2024), a training-free method where the model expresses uncertainty through prompting.

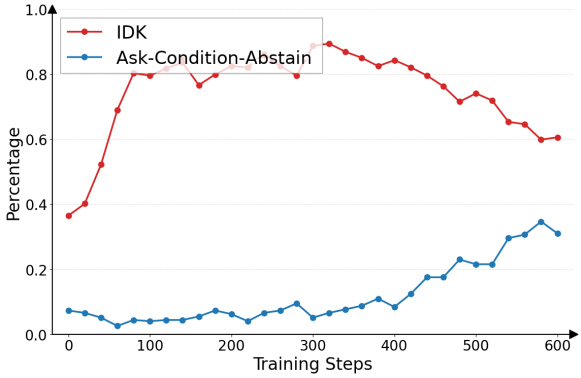

Figure 4: As ACA-RL training progresses, abstention emerges rapidly in early stages, while conditional formulation and elicitation increase more gradually.

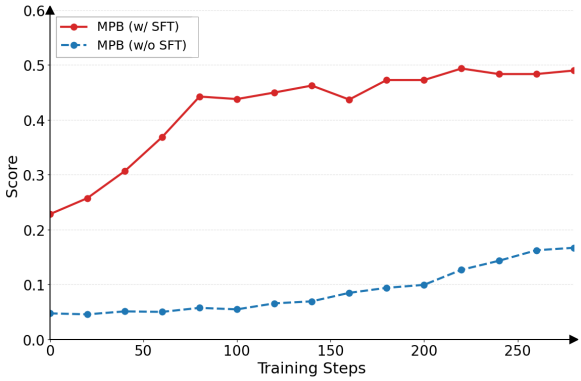

Figure 5: Impact of cold-start SFT on ACA-RL convergence. SFT initialization improves learning efficiency and final MPB score under the plotted training budget.

As shown in Table 6, ACA-RL obtains a higher MPB score than this prompting baseline (51.73 vs. 9.58). This suggests that explicit missing-premise training is more effective in our setting than prompting alone, though it does not rule out stronger prompting or screen-then-answer systems.

Table 6: Comparison with training-free verbalized uncertainty prompting on Qwen3-8B.

Method	MPB	UMWP	SUM
Cold-SFT	22.81	33.53	20.51
Vanilla PPO	8.66	8.84	9.41
LM Introspection	9.58	32.01	13.99
ACA-RL (Ours)	**51.73**	**51.50**	**46.91**

Training Data Size. We investigate the scaling properties of ACA-RL using 10k, 20k, and 52k samples. Figure 6 shows that larger generated datasets improve later-stage performance on both GSM8K and MPB in this sweep, suggesting that additional missing-premise instances provide use-

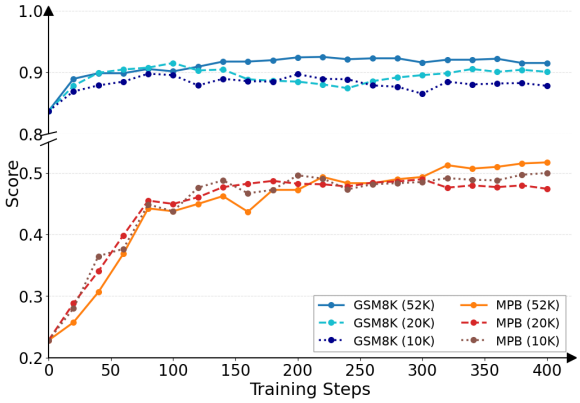

Figure 6: Effect of training data size on ACA-RL performance. Larger generated datasets lead to higher final scores across the plotted training steps on GSM8K and MPB.

ful training diversity. This supports the value of scalable missing-premise data construction, though the experiment does not by itself establish a full scaling law.

7 Related Work

Reinforcement Learning for Reasoning. RL has become a common approach for improving the reasoning behavior of LLMs (Kimi Team et al., 2025; DeepSeek-AI et al., 2025; Tong et al., 2024; Wang et al., 2025; He et al., 2025). Outcome and process rewards provide scalable optimization targets for fully specified tasks (Lambert et al., 2024; Lightman et al., 2023; Uesato et al., 2022). Our work focuses on a complementary case: inputs whose premises are missing, ambiguous, or contradictory, where a single final-answer reward is not enough to specify the desired response.

Missing-Premise and Unanswerable Questions. Prior work constructs unanswerable or underspecified questions to measure hallucination and abstention behavior. UMWP contains unanswerable MathWorld problems annotated by human experts (Sun et al., 2024); Treecut synthesizes unanswerable math problems by removing a dependency edge (Ouyang, 2025); and related benchmarks study unreasonable math problems, logical inconsistencies, and abstention failures (Ma et al., 2026; Rahman et al., 2025; Kirichenko et al., 2025). MPB extends this line by evaluating a broader behavior taxonomy over missing-premise perturbations rather than only final-answer refusal.

Clarification, Ambiguous QA, and Uncertainty-Aware Reasoning. Another line of work studies how LLMs recognize and communicate uncertainty (Tsai et al., 2024; Wang et al., 2024; Huang et al., 2025; Ji et al., 2025). Clarification-question work studies when a system should ask a user to resolve ambiguity (Madge et al., 2025), while ambiguous-QA settings often emphasize recovering multiple interpretations or plausible answers (Min et al., 2020). ACA-RL differs in its training target: asking is only one useful terminal behavior, alongside conditioning on the missing premise and abstaining when no informative conditional answer is available. The method therefore trains a single-turn reasoning policy under missing premises rather than only detecting ambiguity or generating a clarification question.

Abstention and Verbalized Uncertainty. Some methods encourage explicit IDK responses (Wu et al., 2025), while others examine symbolic unknowns, prompting-based uncertainty expression, or uncertainty-aware planning (Hu et al., 2024; Correa and de Matos, 2025; Li et al., 2023). ACA-RL is positioned within this family as a reinforcement-learning method that rewards a hierarchy of missing-premise responses: ask, condition, then abstain, with unsupported assumptions and hallucinated answers penalized.

8 Conclusion and Future Work

This work argues that reasoning models need a learned response policy for questions whose premises do not determine a unique answer. ACA-RL trains on missing-premise problems generated by a reasoning-graph pipeline and uses structured behavioral rewards to encourage asking, conditioning, or abstaining instead of fabricating a value. Together with MPB, it provides a concrete training and evaluation framework for missing-premise behavior. Across multiple model families, ACA-RL improves Behavior Scores on missing-premise and unanswerable benchmarks while preserving competitive performance on well-posed reasoning tasks in the reported evaluations. This points toward a broader mission for NLP systems: models should not only answer when information is complete, but also recognize when interaction, retrieval, or tool use is required to make a task answerable.

Limitations

Although ACA-RL trains models to detect missing premises, Active Elicitation is evaluated as a single-turn textual response. The method does not train retrieval, tool use, or multi-turn clarification policies that can actually acquire the missing premise. This leaves a gap between identifying an underspecified problem and resolving it in agentic workflows.

Furthermore, our training data is derived from a structural perturbation pipeline applied to logical problems. While efficient, these synthetic "broken" links may not fully capture the messy, implicit, or semantic ambiguity found in organic user queries. As a result, the model's robustness is primarily verified against logical incompleteness rather than the full spectrum of open-world ambiguity.

Finally, both training-time reward assignment and MPB scoring use the same behavior taxonomy. The held-out MPB split prevents instance-level training leakage, but future work should still test independent scoring protocols and conduct larger-scale human scoring. The margins over IDK-RL should therefore be interpreted as gains under the reported Behavior Score rubric.

References

Art of Problem Solving. 2024. Aime problems and solutions.

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

Andrew GA Correa and Ana CH de Matos. 2025. Entropy-guided loop: Achieving reasoning through uncertainty-aware generation. *arXiv preprint arXiv:2509.00079*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Haoyang He, Zihua Rong, Kun Ji, Chenyang Li, Qing Huang, Chong Xia, Lan Yang, and Honggang Zhang. 2025. Rethinking reasoning quality in large language models through enhanced chain-of-thought via rl. *Preprint*, arXiv:2509.06024.

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

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei Koh, and Bryan Hooi. 2024. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in large language models. *Preprint*, arXiv:2402.03271.

Liangjie Huang, Dawei Li, Huan Liu, and Lu Cheng. 2025. Beyond accuracy: The role of calibration in self-improving large language models. *Preprint*, arXiv:2504.02902.

Ziwei Ji, Lei Yu, Yeskendir Koishekenov, Yejin Bang, Anthony Hartshorn, Alan Schelten, Cheng Zhang, Pascale Fung, and Nicola Cancedda. 2025. Calibrating verbal uncertainty as a linear feature to reduce hallucinations. *Preprint*, arXiv:2503.14477.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *Preprint*, arXiv:2509.04664.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 77 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. 2025. Abstentionbench: Reasoning llms fail on unanswerable questions. *Preprint*, arXiv:2506.09038.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Jiaxuan Li, Lang Yu, and Allyson Ettinger. 2023. Counterfactual reasoning: Testing language models' understanding of hypothetical scenarios. *Preprint*, arXiv:2305.16572.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog.

Jingyuan Ma, Damai Dai, Zihang Yuan, Rui Li, Weilin Luo, Bin Wang, Qun Liu, Lei Sha, and Zhifang Sui. 2026. Large language models struggle with unreasonability in math problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32428–32436.

Chris Madge, Matthew Purver, and Massimo Poesio. 2025. Referential ambiguity and clarification requests: comparing human and llm behaviour. *Preprint*, arXiv:2507.10445.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 5783–5797.

Jialin Ouyang. 2025. Treecut: A synthetic unanswerable math word problem dataset for llm hallucination evaluation. *Preprint*, arXiv:2502.13442.

A M Muntasir Rahman, Junyi Ye, Wei Yao, Sierra S. Liu, Jesse Yu, Jonathan Yu, Wenpeng Yin, and Guiling Wang. 2025. From blind solvers to logical thinkers: Benchmarking llms' logical integrity on faulty mathematical problems. *Preprint*, arXiv:2410.18921.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Linxin Song, Taiwei Shi, and Jieyu Zhao. 2025. The hallucination tax of reinforcement finetuning. *Preprint*, arXiv:2505.13988.

Yuhong Sun, Zhangyue Yin, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Hui Zhao. 2024. Benchmarking hallucination in large language models based on unanswerable math word problem. *Preprint*, arXiv:2403.03558.

Yongqi Tong, Sizhe Wang, Dawei Li, Yifan Wang, Simeng Han, Zi Lin, Chengsong Huang, Jiaxin Huang, and Jingbo Shang. 2024. Optimizing language model's reasoning abilities with weak supervision. *Preprint*, arXiv:2405.04086.

Yongqi Tong, Yifan Wang, Dawei Li, Sizhe Wang, Zi Lin, Simeng Han, and Jingbo Shang. 2023. Eliminating reasoning via inferring with planning: A new framework to guide llms' non-linear thinking. *Preprint*, arXiv:2310.12342.

Yao-Hung Hubert Tsai, Walter Talbott, and Jian Zhang. 2024. Efficient non-parametric uncertainty quantification for black-box large language models and decision planning. *Preprint*, arXiv:2402.00251.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *Preprint*, arXiv:2211.14275.

Keheng Wang, Feiyu Duan, Peiguang Li, Sirui Wang, and Xunliang Cai. 2024. Llms know what they need: Leveraging a missing information guided framework to empower retrieval-augmented generation. *Preprint*, arXiv:2404.14043.

Sizhe Wang, Yongqi Tong, Hengyuan Zhang, Dawei Li, Xin Zhang, and Tianlong Chen. 2025. Bpo: Towards balanced preference optimization between knowledge breadth and depth in alignment. *Preprint*, arXiv:2411.10914.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, and 1 others. 2024. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 4:2.

Yetao Wu, Yihong Wang, Teng Chen, Ningyuan Xi, Qingqing Gu, Hongyang Lei, and Luo Ji. 2025. Lamss: When large language models meet self-skepticism. *Preprint*, arXiv:2409.06601.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Gal Yona, Roee Aharoni, and Mor Geva. 2024. Can large language models faithfully express their intrinsic uncertainty in words? *arXiv preprint arXiv:2405.16908*.

Table 7: Diagnostic MPB Behavior Scores across six missing-premise perturbation tasks. Abbreviations: Rel.r = Relationship removal; Rel.unquan.r = Relationship unquantifiable replacement; Num.val.r = Numerical value removal; Enti.dis = Entity disruption; Qual.dis = Qualifier disruption; Cond.con = Condition contraction. The final column reports the benchmark-level ask/condition/abstain Behavior Score.

Models	Reasoning Mode	Rel.r	Rel.unquan.r	Num.val.r	Enti.dis	Qual.dis	Cond.con	MPB score
Qwen3-235B-A22B-Instruct	Thinking	5.29	12.45	4.61	1.96	0.36	0.18	6.75
	Direct	6.48	10.68	4.33	2.01	0.41	0.18	12.32
Qwen-Plus	Thinking	5.79	10.86	4.33	2.24	0.59	0.09	7.48
	Direct	5.16	13.14	5.52	2.10	0.68	0.09	6.66
Qwen3-Next-80b-A3B	Thinking	5.66	12.32	4.88	2.46	0.36	0.18	7.12
Deepseek-V3.1	Direct	6.61	11.54	5.16	2.10	0.78	0.00	5.57
Qwen2.5-72b-Instruct	Direct	6.34	11.59	4.84	2.14	0.27	0.05	8.03
Deepseek-R1	Thinking	6.57	12.91	4.61	1.92	0.46	0.00	6.20
Llama-3.3-70B-Instruct	Direct	6.02	12.09	5.47	1.87	0.46	0.09	4.11
QwQ-32B	Thinking	6.43	11.18	4.74	1.87	0.46	0.00	9.76
GPT-based model	-	6.43	11.50	5.06	2.28	0.41	0.09	**20.07**

A MPB: Construction and Diagnostics

Current reasoning benchmarks are largely confined to well-posed problems and therefore do not directly assess model behavior under imperfect information. Related work on unanswerable math questions often measures hallucination or abstention rates, rather than the broader response categories studied here. To address this evaluation gap, we introduce the *Missing-Premise Benchmark* (MPB), a benchmark for measuring model behavior when confronted with underspecified or inconsistent problem statements. MPB comprises 274 well-curated instances spanning mathematical, logical, and real-world word problems, each intentionally designed with missing or conflicting premises. Source problems are drawn from Tong et al. (2023) and DeepscaleR (Luo et al., 2025), then transformed by the missing-premise synthesis pipeline before filtering and verification.

The MPB benchmark is curated from a held-out candidate pool generated by our pipeline (Section 3.1) through a multi-stage filtering process and is not included in the 120K ACA-RL training set. The process begins with automated checks to verify the successful perturbation of each problem and filter out malformed outputs. Surviving candidates then undergo a two-tier qualitative review: first, an automated quality filter screens each problem for naturalness, plausibility, and subtlety; then, three human experts, each with a graduate degree, verify the problem pair's validity (s_0, s'), the analysis's accuracy (a_{gap}), and overall quality.

Evaluation on MPB classifies each response using our behavioral hierarchy (Section 4) to yield a distribution of response behaviors. This diagnostic view separates silent hallucination, explicit assumption, abstention, conditional formulation, and active elicitation. Detailed grading instructions are provided in Appendix E.4.

The results in Table 7 suggest that MPB difficulty varies across perturbation types. Numerical value removal and relationship unquantifiable replacement receive higher scores for several models, while condition contraction and qualifier disruption remain difficult. The two inference modes, *Thinking* and *Direct*, also show different behavior patterns. No listed model is uniformly strong across all six perturbation categories, which supports using MPB as a diagnostic benchmark for missing-premise behavior.

Table 8: Condition-perturbation operations used to construct missing-premise instances, with examples from MPB. Each **Perturbed Question** lacks enough information for a direct answer, yet a strong model may still produce a confident response.

Construction Methods	Original Question	Perturbed Question	Example Model Response
Relationship Removal - If the condition involves a relationship between two entities, remove that relationship.	Five girls are sitting facing towards the north. Rekha is between Shalini and Neetu. Pooja is to the immediate right of Neetu, and Shalini is to the immediate right of Neha. Who is sitting in the middle?	Five girls are sitting in a row facing north. It is known that Rekha is sitting with Shalini and Neetu. Based on this information, who is sitting in the middle?	The response treats "sitting with" as enough ordering information and returns a definite middle person, although the missing relation makes the arrangement underdetermined.
Relationship Unquantifiable Replacement - Transform a definite quantitative relationship into an indefinite or non-numeric one.	Jayant introduces a man to his friend as his wife's father's son. The man is Jayant's...?	While attending a family gathering, Jayant introduces a man to his cousin, explaining that the man is part of his wife's father's extended family. Jayant's cousin, curious about their relationship, tries to determine the exact connection. The man is Jayant's ...?	The response gives a broad in-law relation but still presents it as a resolved answer even though the exact family relation is no longer specified.
Numerical Value Removal - Directly remove the numerical value from the quantitative relationship.	In a mathematics contest with ten problems, a student gains 5 points for a correct answer and loses 2 points for an incorrect answer. If Olivia answered every problem and her score was 29, how many correct answers did she have?	In a mathematics contest consisting of ten problems, each correct answer adds points to a contestant's score, while each incorrect answer subtracts points. Olivia attempted every problem, and after the contest, her total score was 29. How many of the ten problems did Olivia answer correctly?	The response invents point values and returns a numeric answer, even though the scoring rule has been removed.
Qualifier Removal - Remove the qualifier.	Extend the square pattern of 8 black and 17 white square tiles by attaching a border of black tiles around the square. What is the ratio of black tiles to white tiles in the extended pattern?	In a tile pattern made up of black and white square tiles arranged in a square, a border of tiles is attached around the square. What is the ratio of black tiles to white tiles in the extended pattern? The answer should be a numeric value.	The response assumes a particular square size and border configuration, then returns a numeric ratio despite the missing tile counts.
Qualifier Disruption - Replace the qualifier with a specific but contextually irrelevant condition, thereby invalidating the original constraint while ensuring the new condition remains plausible and non-absurd.	There are 6 people in a family. Each one of them likes a different colour: Blue, Red, Pink, Green, Yellow, and White. Seema, who likes Red, is Anitha's mother-in-law and Anitha is Raja's wife. Dinesh is Rohan's father who dislike the blue or white. Bavya likes the yellow and is Rohan's sister, who likes Pink. Raja does not use White. Which colours is liked by Anitha?	In a family where each member likes a different color and has various relationships to one another, there is one known preference: Raja does not use White colour when it is raining. Given this information and the possible color choices of red, blue, green, yellow, and white, which colours is liked by Anitha?	The response solves a full logic-grid puzzle even though most family and color constraints have been removed.
Entity Disruption - Replace the condition's entity with a different but contextually plausible entity that is unrelated to the problem, ensuring it does not resemble or ambiguously refer to any existing entity and the resulting sentence remains grammatically correct and natural.	For how many ordered pairs (b,c) of positive integers does neither $x^2 + bx + c = 0$ nor $x^2 + cx + b = 0$ have two distinct real solutions?	Let b and c be positive integers. Consider the two quadratic equations $x^2 + bx + z = 0$ and $x^2 + zx + b = 0$. It is given that neither of these quadratics has two distinct real solutions. How many ordered pairs (b, c) of positive integers satisfy this condition?	The response treats the unrelated variable z as if it were specified and counts ordered pairs for b, c anyway.
Condition Contraction - Narrow a broadly applicable condition into a more specific one, thereby causing partial information loss and breaking the original coverage.	For any positive integer n, define [n] to be the sum of the positive factors of n. For example, [6] = 1 + 2 + 3 + 6 = 12. Find [[11]].	For any positive integer n less than 10, [n] is defined as the sum of the positive factors of n, and it is given that [6] equals 12. Find [[11]].	The response applies the definition outside its stated domain and returns the original well-posed answer.

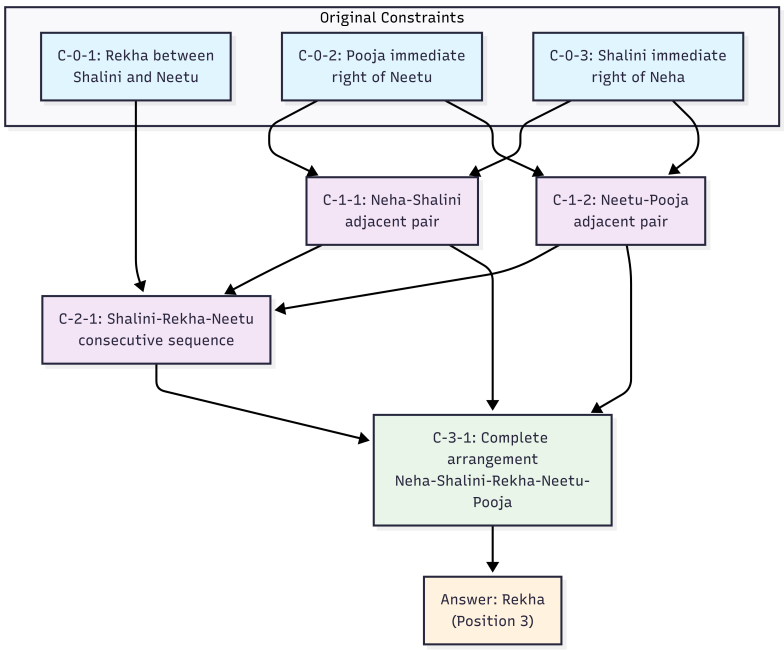

Figure 7: An example of our reasoning DAG illustrating how original constraints are progressively combined into intermediate inferences, ultimately yielding the complete arrangement and final answer.

B Condition Perturbation Definitions and Examples

Detailed definitions and examples of the missing-premise construction methods are shown in Table 8.

C Reasoning-Graph Example

An example reasoning graph is shown in Figure 7.

D Behavioral Scoring Example

Scoring Example

Question. Two parabolic curves with constants a and b intersect the coordinate axes, producing four intercept points. These points form a kite whose area is given. What is $a + b$?
Missing premise. The problem does not specify the actual equations of the parabolas or enough constraints to determine a and b.
Undesired response. A silent-hallucination response chooses a symmetric configuration, assigns unstated equations, and returns a numeric value.
Preferred response. A conditional-formulation response states that $a + b$ is not identifiable from the given information and either expresses the answer in terms of additional parameters or asks for the missing equations/constraints.

E Experiment Details

E.1 Training and Evaluation Details

Base Model and Implementation. To evaluate ACA-RL across different architectures and scales, we use two series of models: the **Qwen3** family (8B and 14B) (Yang et al., 2025) as representative reasoning-heavy models, and **Llama3.1-8B-Instruct** (Grattafiori et al., 2024) as a representative instruction-tuned model. For the Qwen3 series, we initially perform a supervised fine-tuning (SFT) cold-start on our synthesized data, yielding the *Cold-start SFT* checkpoints as the initialization for subsequent RL phases. In contrast, for Llama3.1-8B-Instruct, we directly use the vanilla instruct model as the baseline without any additional SFT distillation of "Chain-of-Thought" (CoT) or "Think" trajectories. This preserves the model's original instruction-following setup rather than adding an extra reasoning-trace distillation stage. All reinforcement learning experiments are conducted using the PPO algorithm (Schulman et al., 2017) implemented via the open-source veRL library (Sheng et al., 2025). The KL coefficient is set to 0.0 to allow broad policy exploration under missing-premise environments. We keep the total number of answerable training queries and computational budget consistent across all RL baselines, and the synthetic data is identical between IDK-RL and ACA-RL. Detailed hyperparameters, prompt templates, and training configurations are provided in Table 10. In the ablation analysis presented in Section 6, we use Qwen3-8B as a representative model.

Evaluation Datasets. We evaluate all models on two categories of benchmarks:

- **Benchmarks**: Our proposed **MPB** benchmark, along with two other unanswerable question datasets, UMWP (Sun et al., 2024) and SUM (Song et al., 2025), to assess out-of-distribution (OOD) robustness.
- **General Reasoning Benchmarks**: High-difficulty, well-posed math and logic benchmarks, including AIME'24 (Art of Problem Solving, 2024), MATH-500 (Hendrycks et al., 2021), and GSM8K (Cobbe et al., 2021), to measure general reasoning capabilities.

For MPB, we report Behavior Score and IDK score (§4). For UMWP and SUM, we report the same Behavior Score alongside IDK score when used in the analysis (Section E.4). For general reasoning benchmarks, we report Pass@1 on GSM8K and MATH-500, and average Pass@1 over 8 samples on AIME'24.

Data Source. Our missing-premise training data is synthesized exclusively from DeepscaleR (Luo et al., 2025), where the answerable questions are taken directly from the original dataset for training.

E.2 MPB Scoring

For the MPB benchmark, we use a GPT judge. Each response is first classified into one of the five behavior categories in Section 4. The category is then mapped to a discrete Behavior Score using Table 9, and benchmark-level values are arithmetic means over test instances. This is why MPB results appear as continuous values in the main tables even though each individual response receives one of five discrete scores.

Table 9: Mapping between ask/condition/abstain behavior labels and final Behavior Scores.

Reward Score	Behavior Score	Description
-1.0	0	Silent Hallucination
-0.3	25	Explicit Assumption
0.3	50	Abstains from Answering
0.6	75	Conditions on a Variable
1.0	100	Asks for the Missing Premise

E.3 Binary IDK Scoring

The IDK scoring process is based on the same behavioral framework but is adapted to a binary rubric (0 for incorrect, 1 for correct). In line with the evaluation protocol of Song et al. (2025), we append the instruction "If you don't know the answer, reply with `\boxed{I don't know.}`" to each question prompt.

Table 10: Hyperparameters of the PPO algorithm implemented based on the veRL framework.

Category	Hyperparameter	Value
Trainer	Nodes	4
	GPUs per node	8
	Total steps	400
	Gradient checkpointing	True
Algorithm	Advantage estimator	GAE(λ=1, γ=1)
	Use KL in reward	False
Actor	Learning rate	1×10^{-6}
	Mini-batch size	128
	Clip ratio	0.2
	Entropy coefficient	0
	Use dynamic batch size	True
	Ulysses sequence parallel size	4
Rollout	Backend	vLLM
	Temperature	1.0
	Top-p	1.0
	Tensor model parallel size	2
Critic	Learning rate	1×10^{-6}
	Warm-up steps	0
	Ulysses sequence parallel size	4
Reward Judge	Judge type	GPT-based judge
	Output	Five behavior categories
Data	Batch size	512
	Max response length	14000

E.4 Additional Evaluator Agreement Check

Setting. For MPB scoring, we adopt GPT-5 as the judge. Each response is assigned to one terminal behavior category, and the category is then mapped to the reward or Behavior Score shown in Table 9. As an additional reliability check, three human experts, each with a graduate degree, annotated sampled GPT-5 and Qwen3-235B-A22B-Instruct responses using the same behavior labels.

Conclusion. Human labels agree with the GPT judge on approximately 98% of the sampled GPT-5 responses and 95% of the sampled Qwen3-235B-A22B-Instruct responses. This agreement check supports the consistency of the reported MPB scoring protocol and broader evaluator independence.

F Data Synthesis Details

This process is implemented as an LLM-agent workflow. For reproducibility, the released supplement will include the prompt templates used for problem decomposition, reasoning graph generation, surgical condition perturbation, reconstruction, and rechecking. We summarize the role of each stage below rather than embedding the raw prompt files in the paper body.

F.1 Problem Decomposition

The decomposition prompt asks the model to separate each source problem into background information, explicit conditions, and the target query.

This representation provides the input structure for reasoning-graph construction.

F.2 Reasoning Graph Generation

The reasoning-graph prompt asks the model to convert a solution into a directed acyclic graph whose nodes are conditions or intermediate conclusions and edges represent dependency relations.

F.3 Surgical Condition Perturbation

The perturbation prompt selects a condition on the reasoning path and applies one of the condition-breaking operations in Table 8. The goal is to remove or alter a premise that is necessary for a unique answer while keeping the edited question fluent and plausible to a reader.

F.4 Reconstruction

The reconstruction prompt rewrites the perturbed conditions, original background, and target query into a fluent problem statement.

F.5 Recheck

F.5.1 Reasoning Correctness

The reasoning-correctness check verifies that the original reasoning graph and answer remain coherent before perturbation.

F.5.2 Missing-Premise Unanswerability

The missing-premise check verifies that the perturbed problem no longer contains enough information to determine the original answer and that the induced gap is identifiable.

G Use of LLMs

LLMs are used as methodological components in this work for data synthesis, filtering, and GPT-based behavioral judging, as described in the method and experiment sections. Separately, general-purpose LLM writing assistants may have been used for minor wording, grammar, and formatting support. The research questions, method design, experimental analysis, and final claims are the responsibility of the authors.