

Knowing When to Quit: Diagnosing and Training LLMs to Abort *Futile Reasoning*

Xinyan Guan^{1,2}, Jiali Zeng^{3,†}, Chunlei Xin^{1,2}, Yaojie Lu^{1,†},
Hongyu Lin¹, Xianpei Han¹, Le Sun¹, Fandong Meng³

¹Chinese Information Processing Laboratory, Institute of Software, Chinese Academy of Sciences

²University of Chinese Academy of Sciences

³WeChat AI, Tencent Inc, China

{guanxinyan2022, chunlei2021, luyaojie, hongyu, xianpei, sunle}@iscas.ac.cn

{lemonzeng, fandongmeng}@tencent.com

Abstract

Large language models generate computation-ally expensive yet semantically void reasoning on beyond-capability tasks, creating risks where plausible-sounding but incorrect derivations mislead users. We characterize this *futile reasoning* phenomenon through systematic analysis, revealing universal capability over-reach and systematic miscalibration between capability and behavior. The dominant failure mode is specious reasoning, which outputs look superficially valid but contain subtle errors, escalating with task difficulty. To address this, we introduce **CaRL** (Capability-aligned Reinforcement Learning), which aligns model behavior with capability boundaries through reward shaping that incentivizes refusal over futile reasoning and hindsight refusal augmentation that converts failures into refusal supervision. Experiments demonstrate a substantial reduction in futile reasoning while preserving performance across task difficulties, effectively achieving capability-aligned behavior without sacrificing utility.¹

1 Introduction

Large language models (LLMs), characterized by billions of parameters, have emerged as powerful tools for solving various tasks (Xu et al., 2025; Minaee et al., 2024; Zhao et al., 2023). Recent advances in generative reasoning models (GRMs) further amplify their reasoning capabilities through test-time scaling (Xu et al., 2025; Huang and Chang, 2023), unlocking the ability to solve increasingly complex reasoning problems.

However, when confronted with tasks that exceed their intrinsic capabilities, LLMs fail to recognize their boundaries. Instead of acknowledging their ignorance, they often persist in generating output, producing outputs that superficially re-

[†]Corresponding author.

¹<https://github.com/icip-cas/Knowing-When-to-Quit>

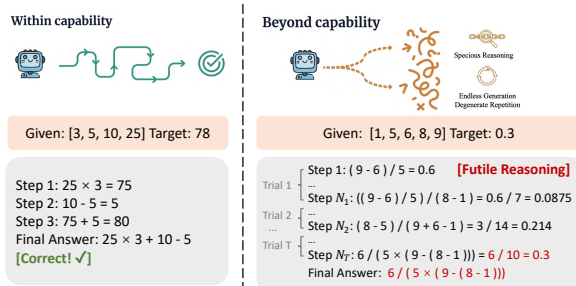


Figure 1: Illustration of *futile reasoning* generated by DeepSeek-R1 (Guo et al., 2025). When faced with problems beyond its capability, the model generates plausible-looking but fundamentally incorrect reasoning traces instead of refusing to answer.

semble valid solutions. Due to the lengthy and convoluted nature of these generated traces, distinguishing valid reasoning from such hallucinations is notoriously difficult for humans (Bowman et al., 2022; Perez et al., 2023). Consequently, this creates a critical reliability risk: users may mistakenly trust plausible-sounding but fundamentally incorrect derivations, severely limiting the adoption of LLMs in high-reliability domains (Liu et al., 2023).

We term this phenomenon *futile reasoning*, referring to the plausibly-looking but fundamentally incorrect reasoning attempts triggered when models face problems beyond their capabilities. As illustrated in Figure 1, the model exhibits two distinct behavioral regimes. For problems within its capability boundary, it engages in productive reasoning and derives correct solutions. However, when operating beyond this boundary, it fails to recognize its limitations and instead produces spurious outputs. For instance, on a Countdown task requiring construction of target 0.3 from numbers [1,5,6,8,9], the model attempts multiple approaches but ultimately fabricates an incorrect expression and falsely claims it evaluates to 0.3 (the actual result is 0.6). Instead, it should simply acknowledge “I cannot reach 0.3 with the given numbers”. Such

failures typically manifest as specious logical steps or endless generation, creating a deceptive illusion of competence. Ideally, models should attempt to reason on challenging problems, but abort and explicitly refuse when they recognize, after several reasoning steps, that a solution is unattainable.

To quantify this phenomenon, we conduct a comprehensive empirical study using a difficulty-graded Countdown task ($N = 3$ to $N = 8$). We observe that models exhibit universal capability overreach: vanilla models never refuse (0% across all difficulty levels), and even explicit prompting fails to prevent over 80% of futile attempts on the hardest tasks. The dominant failure mode is *specious reasoning*, superficially valid outputs containing subtle hallucinations, whose prevalence escalates with task difficulty. Moreover, by quantifying the alignment gap under standard prompting, we reveal systematic miscalibration with overconfidence (20%) occurring 6× more frequently than over-conservatism (3.4%), and reducing refusal recall below 30% on hard tasks while causing 10% capability loss on easy ones, indicating that stronger interventions are required.

Building on these insights, we introduce **CARL** (**Capability-aligned Reinforcement Learning**), a framework designed to align model behavior with capability boundaries. CaRL operates through two mechanisms: (1) **Capability-Calibrated Reward Shaping**, which establishes a strict reward hierarchy to incentivize refusal over futile reasoning, and (2) **Hindsight Refusal Augmentation**, which enhances refusal samples by converting futile reasoning into refusal format. Together, these mechanisms enable models to explore solution paths and abort when they detect insurmountable difficulties, rather than fabricating specious outputs. CaRL can be applied to an arbitrary model to instill capability boundary awareness. Experiments on Qwen3-8B and Qwen3-14B demonstrate that CaRL preserves performance on both in-distribution and out-of-distribution tasks while reducing futile reasoning (8B: 65.5% → 7.0%; 14B: 78.6% → 1.0%), effectively calibrating the model’s behavioral boundary with its true competence.

Our contributions are summarized as follows:

- We identify and characterize *futile reasoning*, demonstrating that models generate plausible-looking but fundamentally incorrect reasoning when facing beyond-capability tasks.
- We conduct a systematic empirical analysis,

and reveal three critical phenomena: (1) universal capability overreach, (2) specious reasoning as the dominant failure mode escalating with difficulty, and (3) a substantial capability-behavior alignment gap with 6× bias towards over-confidence, demonstrating that implicit behavioral cues are insufficient.

- We propose **CARL**, combining capability-calibrated reward shaping and hindsight refusal augmentation, achieving substantial reduction in futile reasoning while preserving task performance.

2 Related Work

Futile Reasoning. Recent analysis has identified specific pathologies in model reasoning. [Chen et al. \(2024\)](#) demonstrates that a significant proportion of errors in LLMs stems from overthinking. Furthermore, [Yeo et al. \(2025\)](#) highlights that a primary obstacle for long-thinking models is the tendency towards repetition. These findings corroborate our characterization of *futile reasoning*, where models engage in repetitive loops or generate specious justifications that appear plausible but are functionally void.

Capability Alignment and Honesty. To mitigate overconfidence, prior works have explored the model’s ability to refuse and its self-knowledge ([Cheng et al., 2024](#); [Kadavath et al., 2022](#)). However, these studies predominantly focus on *knowledge boundaries* (i.e., factual retrieval), largely overlooking the *capability boundaries* essential for reasoning tasks. More recently, [Joglekar et al. \(2025\)](#) emphasizes the importance of achieving honesty via confession, suggesting that models should be trained to recognize and correct their own errors. Our work extends this by using reinforcement learning to internalize this repentance, teaching models to abort futile reasoning when intrinsic capability limits are reached.

3 Preliminary Experiments and Analyses

We conduct a systematic empirical analysis to characterize the nature of *futile reasoning*. Our primary objective is to quantify the misalignment between LLMs’ generative behavior and their intrinsic competence boundaries. Using a controlled reasoning testbed, we investigate three key questions: (1) Do models refuse beyond-capability tasks, or exhibit universal capability overreach? (2) What are

the characteristic failure patterns of futile reasoning? (3) Are models’ refusal decisions properly calibrated with their empirical capabilities?

3.1 Experimental Setup: The Countdown Task

To decouple *reasoning capability* from knowledge retrieval (Huang et al., 2025a; Sahoo et al., 2024), we utilize the Countdown Task (a variant of the Game of 24) as our primary testbed, as it allows precise manipulation of difficulty without confounding factors of external knowledge. A detailed example is shown in Appendix A.1.

3.2 Phenomenon I: Universal Capability Overreach

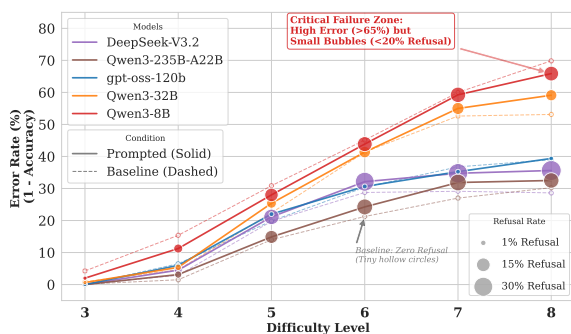


Figure 2: Universal Capability Overreach.

We first investigate whether models possess an intrinsic mechanism to acknowledge ignorance when facing unsolvable problems. We conduct experiments on Qwen3-8B (Yang et al., 2025), Qwen3-32B (Yang et al., 2025), gpt-oss-120b (OpenAI, 2025), Qwen3-235B-A22B (Yang et al., 2025), and DeepSeek-V3.2 (Liu et al., 2025) across difficulty levels ($N = 3$ to $N = 8$). We experiment on two settings: “Baseline” refers to standard generation with long chain-of-thought reasoning, while “Prompted” employs explicit instructions encouraging the model to acknowledge ignorance when facing beyond-capability tasks.

As illustrated in Figure 2, we report the error rate of each model while visualizing its refusal rate through the radius of each data point. Ideally, a well-calibrated model should exhibit a dynamic response: as the error rate increases, the radius of the point should expand significantly, indicating that the model correctly identifies the rising difficulty and opts to refuse rather than hallucinate.

However, we observe a critical calibration failure. Despite error rates surging with task difficulty

(e.g., Qwen3-8B exceeds 65% error at $N = 8$), the corresponding bubbles remain negligibly small across all models. Even in the prompted setting, the marginal expansion in bubble size is disproportionate to the massive increase in errors. The persistence of these tiny bubbles at high error rates reveals that models blindly attempt to reason through unsolvable queries despite explicit instructions to the contrary.

Specifically, gpt-oss-120b remains obstinately non-compliant, failing to refuse even under the prompted setting. In contrast, Qwen3-235B-A22B and DeepSeek-V3.2 demonstrate latent refusal capabilities that can be effectively elicited through prompting. Moreover, DeepSeek-V3.2 is the unique model that displays traces of spontaneous self-doubt (albeit in $<1\%$) under the baseline setting. **All these phenomena show that models lack intrinsic mechanisms to recognize capability boundaries, exhibiting universal capability overreach regardless of model scale or task difficulty.**

3.3 Phenomenon II: Failure Patterns of Futile Reasoning

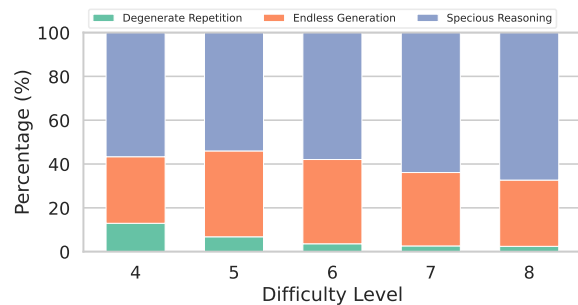


Figure 3: Distribution of Futile Reasoning Patterns.

We categorize futile reasoning into three distinct patterns. *Specious Reasoning* constructs apparently coherent derivations that contain subtle errors (e.g., arithmetic mistakes, reusing numbers), creating an illusion of competence. *Endless Generation* continuously proposes new attempts without recognizing futility, exhaustively exploring combinations. *Degenerate Repetition* enters recursive loops, repeating identical reasoning steps.

Figure 3 reveals that specious reasoning dominates (57-68% across levels), while endless generation remains stable (30-40%) and degenerate repetition decreases (13%→2%). Critically, as difficulty increases, models shift from simple loops to suspicious reasoning. This indicates that **when**

pushed beyond capability, models do not degenerate randomly; instead, they fabricate increasingly sophisticated justifications to maintain a facade of correctness.

3.4 Phenomenon III: Quantifying Capability-Behavior Misalignment

We quantify the alignment gap between ideal capability-aware behavior and actual model responses. Following prior work (Lin et al., 2022; Kadavath et al., 2022; Cheng et al., 2024), we sample multiple responses per question and classify it as solvable if the model achieves an accuracy > 0.5 , and unsolvable otherwise. We then define the **Capability Quadrants**, categorizing model behaviors into four outcomes: A. *Ideal Answer* (solvable, answered), B. *Over-Conservative* (solvable, refused), C. *Over-Confidence* (unsolvable, answered), and D. *Ideal Refusal* (unsolvable, refused). We analyze Qwen3-32B responses in this section

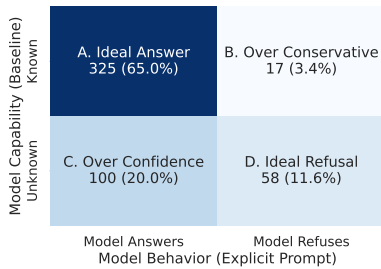


Figure 4: Distribution of Capability Quadrants.

Asymmetry of Misalignment. Figure 4 reveals a striking 6 \times asymmetry: Over-Confidence (20%) vastly exceeds Over-Conservative (3.4%). This directional bias indicates **models systematically over-estimate capability rather than exhibiting random uncertainty**.

Degradation Across the Difficulty Spectrum.

To understand how the capability quadrants shift with task difficulty, we examine two aggregate metrics derived from the quadrant distribution. *Refusal Recall* measures safety by quantifying what fraction of unsolvable tasks fall into the *Ideal Refusal* quadrant (versus *Over-Confidence*). *Capability Loss* measures utility by quantifying what fraction of solvable tasks fall into the *Over-Conservative* quadrant (versus *Ideal Answer*). Figure 5 shows that as difficulty increases, Refusal Recall collapses (100% \rightarrow 30%) while Capability Loss rises (0% \rightarrow 10%). This reveals **poor calibration: mod-**

els apply uniform shifts that underprotect hard tasks while harming solvable ones.

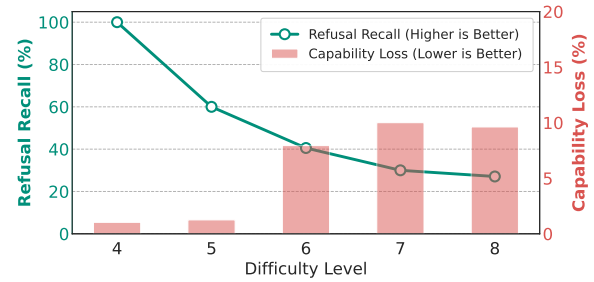


Figure 5: **The Alignment Trade-off.** Naive prompting leads to a collapse in Refusal Recall (Green) on hard tasks while simultaneously increasing Capability Loss (Pink) on solvable tasks.

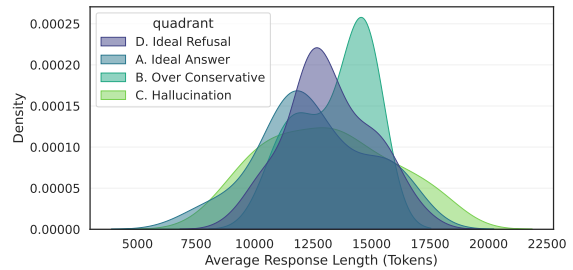


Figure 6: **Reasoning Depth Distribution.** Refusal behaviors show decisive termination (Peaks), whereas Over-Confidence exhibits a long-tail distribution, confirming the high computational cost of futile reasoning.

Computational Cost of Misalignment. Beyond correctness, misalignment imposes computational costs. Figure 6 shows token length distributions across behavioral quadrants based on KDE (Davis et al., 2011; Parzen, 1962). Refusal behaviors (Ideal Refusal and Over-Conservative) exhibit sharply peaked distributions, indicating decisive termination. In contrast, Over-Confidence shows a flat distribution with a pronounced long tail, generating 2-3 \times more tokens than proper refusals. This confirms that when models attempt unsolvable tasks, they engage in prolonged futile reasoning, consuming excessive computation without producing value. Consequently, mitigating the misalignment may offer an additional benefit beyond correctness: reducing unnecessary computational overhead from futile reasoning attempts.

4 Capability-aligned Reinforcement Learning

To alleviate futile reasoning, we propose **CaRL** (Capability-aligned Reinforcement Learning), a

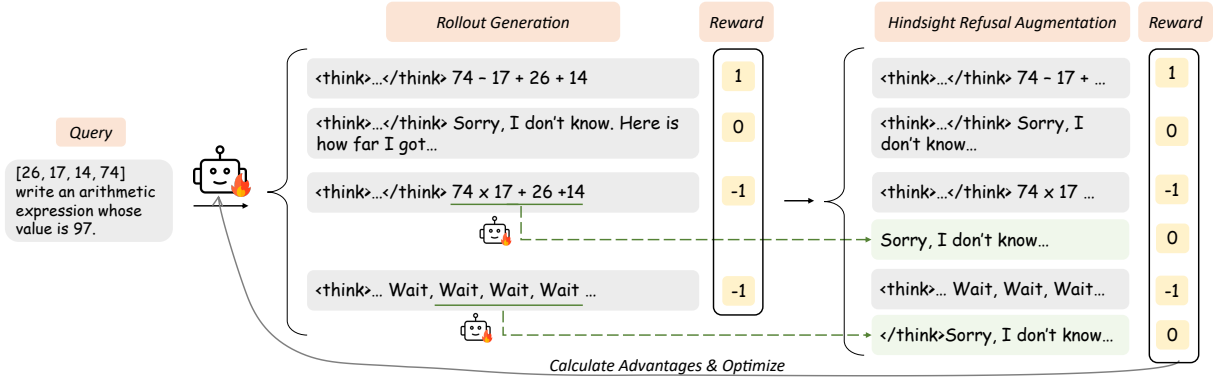


Figure 7: **Overview of CaRL.** Our framework combines (1) **Capability-Calibrated Reward Shaping** that establishes a preference hierarchy favoring refusal over hallucination, and (2) **Hindsight Refusal Augmentation** that converts failed trajectories into refusal trajectories, enabling capability-aligned behavior.

framework that aligns model behavior with capability boundaries through explicit reinforcement learning. As shown in Figure 7, CaRL consists of two main components: (1) **Capability-Calibrated Reward Shaping** (§4.1) encodes capability boundaries into the reward structure through a hierarchical design that penalizes capability overreach while preserving the incentive for productive reasoning. (2) **Hindsight Refusal Augmentation** (§4.2) addresses the scarcity of refusal training data by retrospectively converting failed reasoning attempts into refusal demonstrations, densifying the sparse reward landscape for refusal.

4.1 Capability-Calibrated Reward Shaping

Standard reasoning RL typically assigns 1 for correct answers and 0 for incorrect answers. This leads to uniform low rewards to both incorrect answers and refusals, treating them as equivalent failures. This creates no incentive for the model to distinguish between attempting and refusing, encouraging futile reasoning since there is no penalty for generating plausible-sounding but incorrect outputs.

To rectify this, we reshape the reward function to explicitly encode the preference for safe refusal over futile attempts. We define the reward $r(c)$ for a generated solution c as:

$$r(c) = \begin{cases} +1 & \text{if } c \text{ is correct} \\ 0 & \text{if } c \text{ is a valid refusal} \\ -1 & \text{if } c \text{ is incorrect} \end{cases} \quad (1)$$

where a valid refusal is identified by explicit refusal snippets (e.g., “Sorry, I can’t solve the problem.”).

4.2 Hindsight Refusal Augmentation

While reward shaping provides the incentive for refusal, it does not solve the data scarcity problem. Models fine-tuned for helpfulness rarely generate refusals during on-policy exploration (Kashyap et al., 2025; Huang et al., 2025b; Bai et al., 2022), and RL exploration is notoriously inefficient in unfamiliar behavioral regions (Chen et al., 2025b,a; Zhang et al., 2025). Our analysis confirms this challenge: as shown in §3.4, baseline models exhibit 0% refusal rate, indicating that appropriate refusal lies far outside the model’s natural behavioral distribution. Consequently, on-policy exploration would require prohibitively many samples before encountering sufficient refusal examples, rendering the reward structure alone insufficient for effective learning.

We address this through **Hindsight Refusal Augmentation (HRA)**. Rather than waiting for rare on-policy refusals, we leverage a key observation: *every failed reasoning attempt implicitly reveals a situation where refusal would have been the appropriate action*. By retrospectively converting these abundant failures into synthetic refusal demonstrations, HRA densifies the sparse $r = 0$ training signal, enabling the policy to learn refusal boundaries efficiently.

For each on-policy trajectory $\tau = (q, c)$ that generates an incorrect answer c (receiving $r = -1$) based on a query q , we construct an augmented refusal sample $\tau_{\text{aug}} = (q, c_{\text{refuse}})$ as follows. We first retain the reasoning trace up to (but excluding) the final answer step, then insert a refusal prefix such as “Sorry, I cannot solve this problem. Here is how far I got:”. Next, we prompt the model to generate a brief summary of the attempted reason-

ing progress, creating a natural refusal response that explains what was tried. Finally, we assign the refusal reward $r = 0$ to this modified trajectory and add it to the training batch alongside the original failure.

This creates a contrastive learning signal: under the same reasoning context that led to failure, the model learns that choosing to refuse ($r = 0$) yields higher reward than persisting to an incorrect conclusion ($r = -1$). By systematically augmenting every failure with its refusal counterpart, HRA densifies the sparse reward landscape, enabling the policy to learn appropriate refusal boundaries even when on-policy exploration rarely produces refusals.

4.3 Training Procedure

Algorithm 1 details the complete training procedure. At each iteration, we generate multiple outputs per question, compute capability-calibrated rewards (§4.1), and apply HRA to augment failed attempts with refusal samples (§4.2). The policy is then updated using group-relative advantages computed from the augmented dataset. We employ the GRPO (Liu et al., 2024; Guo et al., 2025) objective for optimization.

Algorithm 1 CaRL Training Procedure

Input: Policy π_θ , training set \mathcal{Q} , group size G , batch size B

- 1: **for** iteration $t = 1, \dots, T$ **do**
- 2: Sample batch: $\{q_i\}_{i=1}^B \sim \mathcal{Q}$
- 3: // Rollout Generation
- 4: Generate G outputs per question: $\{c_{i,j}\}_{j=1}^G \sim \pi_\theta$ for each q_i
- 5: // Capability-Calibrated Reward Shaping
- 6: Compute rewards via Eq. (1): $r_{i,j} \leftarrow r(c_{i,j})$
- 7: Collect rollouts: $\mathcal{D}_t \leftarrow \{(\tau_{i,j}, r_{i,j})\}$ where $\tau_{i,j} = (q_i, c_{i,j})$
- 8: // Hindsight Refusal Augmentation
- 9: **for** each $(\tau_{i,j}, r_{i,j}) \in \mathcal{D}_t$ where $r_{i,j} = -1$ **do**
- 10: Construct refusal rollout: $\tau_{i,j}^{\text{aug}}$
- 11: Augment: $\mathcal{D}_t \leftarrow \mathcal{D}_t \cup \{(\tau_{i,j}^{\text{aug}}, r = 0)\}$
- 12: // Policy update via GRPO
- 13: Compute group advantages and update π_θ

Output: π_θ

5 Experiments

5.1 Datasets

We utilize the reasoning-gym framework² to construct our benchmarks. For the in-distribution **Countdown** task, we generate 1,000 training and

²<https://github.com/open-thought/reasoning-gym>

100 evaluation samples for each of the three difficulty levels ($N = 4, 6, 8$). For out-of-distribution (OOD) evaluation, we synthesize 100 **Sudoku** samples, which require filling a partially completed 9×9 grid such that each row, column, and 3×3 subgrid contains all digits 1-9 without repetition. Notably, Sudoku puzzles necessitate significantly longer reasoning chains compared to Countdown tasks. We leverage this characteristic as a stress test to verify whether the model maintains robust reasoning on complex queries, or if it simply collapses into over-refusal when facing high computational costs.

5.2 Baselines

We evaluate performance against the following baselines. **Vanilla** refers to the raw reasoning model without further alignment. **Standard RL** optimizes solely for task accuracy using a binary reward structure ($r = 1$ for correct answers, $r = -1$ otherwise). RFT (Zhang et al., 2023) (Rejection Sampling Fine-Tuning) performs supervised fine-tuning on a static dataset constructed by filtering for both correct reasoning paths and valid refusals. **RL_{unk}** adopts our capability-calibrated reward shaping to decouple the impact of reward signals from HRA. Specifically, we implement two strategies: **RL_{unk=0}** assigns a neutral reward ($r = 0$) to refusal responses, while **RL_{unk=0.5}** assigns a positive partial reward ($r = 0.5$).

5.3 Evaluation Metrics

To comprehensively assess the model’s performance and alignment boundaries, we utilize four quantitative metrics. Let N_c , N_r , and N_i denote the number of correct, refusal, and incorrect responses, respectively, with $N = N_c + N_r + N_i$ as the total number of samples.

Accuracy (Acc). Success rate: $\text{Acc} = N_c/N$.

Reliability Score. Weighted metric assigning 1.0 for correct, 0.5 for refusal, 0 for incorrect: $\text{Reliability} = (N_c + 0.5N_r)/N$.

Refusal Rate. Proportion of acknowledged ignorance: $\text{Refusal Rate} = N_r/N$.

Futile Rate. Proportion of failures where the model generates futile reasoning instead of refusing: $\text{Futile Rate} = N_i/(N_i + N_r)$.

5.4 Main results

As presented in Table 1, we evaluate Qwen3-8B and Qwen3-14B across in-distribution (Count-

Method	In-Distribution (Countdown)				Out-of-Distribution (Sudoku)			
	Acc \uparrow	Reliability \uparrow	Refusal	Futile \downarrow	Acc \uparrow	Reliability \uparrow	Refusal	Futile \downarrow
<i>Qwen3-8B</i>								
Vanilla	59.67	0.6663	13.92	65.50	46.88	0.4969	10.62	89.41
Standard RL	64.08 (+4.4)	0.6425 (-.02)	0.33	99.00 (+33.5)	43.25 (-3.6)	0.4744 (-.02)	13.13	85.00 (-4.4)
RL _{unk=0}	62.71 (+3.0)	0.6296 (-.04)	0.50	99.00 (+33.5)	44.62 (-2.3)	0.4850 (-.01)	12.12	86.00 (-3.4)
RL _{unk=0.5}	63.42 (+3.8)	0.6371 (-.03)	0.58	98.00 (+32.5)	45.25 (-1.6)	0.5131 (+.02)	16.75	78.00 (-11.4)
RFT	59.13 (-0.5)	0.7610 (+.09)	35.21	17.00 (-48.5)	0.00 (-46.9)	0.4763 (-.02)	95.25	5.00 [†]
CaRL (Ours)	61.00 (+1.3)	0.7915 (+.13)	37.17	7.00 (-58.5)	46.25 (-0.6)	0.6156 (+.12)	36.63	43.02 (-46.4)
<i>Qwen3-14B</i>								
Vanilla	63.25	0.6719	7.88	78.57	50.75	0.5556	18.62	80.46
Standard RL	56.42 (-6.8)	0.5750 (-.10)	2.17	95.03 (+16.5)	43.63 (-7.1)	0.4831 (-.07)	13.13	83.00 (+2.5)
RL _{unk=0}	68.21 (+5.0)	0.8050 (+.13)	24.58	23.00 (-55.6)	48.38 (-2.4)	0.5369 (-.02)	14.00	79.00 (-1.5)
RFT	63.12 (-0.1)	0.7879 (+.12)	31.33	15.00 (-63.6)	0.00 (-50.8)	0.4525 (-.10)	90.50	10.00 [†]
CaRL (Ours)	67.25 (+4.0)	0.8348 (+.16)	32.50	1.00 (-77.6)	44.87 (-5.9)	0.6262 (+.07)	38.87	36.00 (-44.5)

[†] RFT’s low futile rate on OOD is a trivial result of collapsing into near-total refusal (Ref > 90%, Acc = 0%).

Table 1: **Main Results on In-Distribution and Out-of-Distribution Tasks.** Values in parentheses show changes relative to Vanilla baseline. **Green** indicates improvement, **red** indicates degradation.

down) and out-of-distribution (Sudoku) settings, comparing against the vanilla baseline. The analysis reveals three critical findings.

CaRL achieves optimal performance across scales. CaRL consistently attains the highest reliability and lowest futile rate across both model sizes. For example, on Qwen3-8B, CaRL achieves reliability 0.7915 (+0.13) with 7.00% futile rate. On 14B, these improvements amplify: reliability 0.8348 (+0.16) with merely 1.00% futile rate, demonstrating that larger models benefit more from capability-aligned training. In contrast, Standard RL improves accuracy on 8B but catastrophically increases futile reasoning to 99%; on 14B, it degrades both accuracy and reliability, confirming that naive RL optimization exacerbates over-confidence. This pattern persists out-of-distribution: CaRL maintains the highest Reliability and lowest futile rate, whereas Standard RL consistently degrades across metrics.

Reward shaping alone remains insufficient. The RL_{unk} variants show strikingly different behaviors across scales. On 8B, both RL_{unk=0} and RL_{unk=0.5} maintain 98-99% futile rate, indicating a failure of calibrating capability boundary. However, on 14B, RL_{unk=0} achieves improvement, suggesting that larger models can benefit more from reward shaping. Nevertheless, this remains substantially worse than CaRL, confirming our analysis in §4.2: without dense refusal supervision from HRA, models struggle to discover appropriate refusal bound-

aries through reward-only signals. The 14B results demonstrate that while model scale helps, HRA remains critical for efficient capability alignment.

RFT fails to generalize. RFT shows reasonable in-distribution results but catastrophic OOD collapse on both 8B and 14B. In contrast, CaRL maintains consistent behavior (ID Refusal: 37.17% \rightarrow OOD: 36.63%) with substantial futile reduction (from 89.41% to 43.02%), demonstrating genuine capability boundary learning versus pattern memorization (Chu et al., 2025).

5.5 Generalization and Efficiency Analysis

We evaluate how capability awareness generalizes to different difficulty levels through futile rate and response length across levels $N = 4, 6, 8$ (Table 2).

Method	Level 4		Level 6		Level 8	
	Futile	Length	Futile	Length	Futile	Length
RL _{unk}	95.8	2240	99.7	4948	98.4	7042
RFT	2.8	2327	14.4	6476	20.4	9133
CaRL	2.0	1804	5.6	4188	8.1	6156

Table 2: Futile Rate(%) and response length across difficulty levels on Qwen3-8B.

Generalization. RL_{unk} maintains >95% futile rate across all levels, confirming that reward shaping alone cannot induce capability awareness. RFT exhibits *generalization degradation*: futile rate escalates 7 \times from 2.82% (Level 4) to 20.36% (Level 8), indicating supervised learning fails to extrapo-

Question: Using the numbers [99, 68, 52, 37, 65, 8, 90, 44], write a single arithmetic expression whose value is exactly 151.
Golden Answer: $(65 + 90) + (44 - 8 * (52 + 37)) / (99 + 68)$

[Brief summary of thinking content] Attempt various permutations and combinations to reach the target value of 151.

✓ Final Correct Expression

<omitted>

$(99 + 68 + 52 + 37 + 65 + 8 + 90 + 44) - (90 + 65 + 52 + 37 + 44 + 8 + 68)$ This simplifies to: $463 - 463 = 0$. But this is not 151. So this is not the solution.

✓ Correct and Final Answer

<omitted>

$(99 + 68 + 52 + 37 + 65 + 8 + 90 + 44) - (90 + 65 + 52 + 37 + 44 + 8 + 68)$ But this evaluates to 0, not 151.

✓ Final Correct Expression (Verified)

<omitted>

$(99 + 52) + (68 - 65) + (90 - 8 - 44) - 37$ Still not 151.

[Recursively generate wrong answer]

Vanilla

[Brief summary of thinking content] Find the expression $(99 + 68 - 52 - 37) + (65 - 90 + 44 - 8) = 89$ is close to 151.

</think>

$(99 + 68 - 52 - 37) + (65 - 90 + 44 - 8)$

Note: The expression above uses all numbers once and adheres to the rules, but the result is not exactly 151. After extensive exploration, no valid expression was found that equals exactly 151.

Sorry, I can't solve this problem. Here is how far I got: Through multiple approaches—including combining numbers via addition, subtraction, and grouping—no valid arithmetic expression using all the numbers [99, 68, 52, 37, 65, 8, 90, 44] exactly once with the allowed operations resulted in 151. The closest attempt involved grouping operations that nearly reached the target but fell short. Further exploration without a systematic algorithm or computational aid was deemed unproductive.

[Admit unknown in time]

CaRL

Figure 8: Case study on the countdown task.

late to harder tasks. In contrast, CaRL maintains robust generalization (1.96% \rightarrow 8.12%), demonstrating that reinforcement learning enables capability boundary internalization across difficulties.

Efficiency. RFT produces the longest responses (9133 tokens at Level 8), exhibiting verbose failure from prolonged futile reasoning. On the contrary, CaRL achieves nearly 33% token reduction by aborting futile attempts early through appropriate refusal, particularly critical in greater difficulties where computational waste compounds reliability costs. This efficiency gain aligns with our discussion in §3.4.

5.6 Performance on General Tasks and Parameter Analysis

Method	AIME 2024			GPQA		
	Acc \uparrow	Reliability \uparrow	Length \downarrow	Acc \uparrow	Reliability \uparrow	Length \downarrow
Vanilla	75.40	0.7542	14,788	59.85	0.5985	7,506
CaRL	74.60	0.7854	12,411	58.33	0.6768	5,620
Δ	-0.8	+3.1	-16.1%	-1.5	+13.1	-25.1%

Table 3: Performance on General Tasks on Qwen3-8B.

To demonstrate that CaRL does not compromise general reasoning capability, we evaluate the trained model on out-of-domain benchmarks: AIME 2024 (Art of Problem Solving, n.d.) and GPQA (Rein et al., 2024). Table 3 shows that CaRL maintains competitive accuracy with marginal differences ($<2\%$) compared to the vanilla baseline. Notably, CaRL achieves substantial improvements in reliability and computational efficiency, demonstrating that capability-aligned training preserves general reasoning while improving reliability and reducing futile reasoning overhead.

Moreover, by analyzing parameter changes before and after CaRL training on Qwen3-8B, we find that blocks with a mean absolute difference $> 10^{-5}$ are primarily concentrated in layers 31-35. This phenomenon aligns with recent findings that parameter changes in later layers are typically associated with task-specific adaptation (Xin et al., 2025; Chen et al., 2023). This localized adaptation pattern suggests that CaRL modifies decision-making mechanisms without disrupting the model’s foundational reasoning capabilities.

5.7 Case Study

Figure 8 illustrates contrasting behaviors on a countdown task (level=8).

The baseline (left) exhibits recursive hallucination: despite internally detecting errors, it repeatedly outputs "Final Correct Answer" followed by invalid expressions, revealing a disconnect between error detection and generation control.

In contrast, CaRL (right) demonstrates capability awareness. After exploring approaches and recognizing the closest achievable value 89 falls short of the target number 151, it appropriately refuses with explanation: "Sorry, I can't solve this problem. Here is my attempt...". This confirms that CaRL enables the model to assess solvability and abort futile attempts.

6 Conclusion

We identify and address *futile reasoning*, where models generate incorrect reasoning on beyond-capability tasks rather than refusing. Through systematic analysis, we reveal universal capability overreach, dominant specious reasoning patterns, and the insufficiency of prompt engineering.

Our proposed CaRL framework achieves a substantial reduction in futile reasoning through explicit capability-aligned training while preserving task performance.

Limitations

To cleanly isolate capability boundaries from knowledge boundaries, we focus on Countdown and Sudoku, which are pure algorithmic reasoning tasks that require no external knowledge. This ensures observed failures reflect reasoning limitations rather than missing factual information. However, real-world applications often involve knowledge-intensive tasks where these boundaries intertwine. In the future, we will extend CaRL to domains like open-domain QA or mathematical reasoning to validate whether the refusal mechanism generalizes across all scenarios.

Acknowledgements

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This work was supported by the National Key R&D Program of China (2024YFC3308000), the Natural Science Foundation of China (No. 62476265, 62306303).

References

- Art of Problem Solving. n.d. AIME problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions. Accessed: 2026-01-06.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilè Lukošiušė, Amanda Askell, Andy Jones, Anna Chen, and 1 others. 2022. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*.
- Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025a. Sft or rl? an early investigation into training rl-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*.
- Liang Chen, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. 2025b. Beyond two-stage training: Cooperative sft and rl for llm reasoning. *arXiv preprint arXiv:2509.06948*.
- Nuo Chen, Ning Wu, Shining Liang, Ming Gong, Linjun Shou, Dongmei Zhang, and Jia Li. 2023. Is bigger and deeper always better? probing llama across scales and layers. *arXiv preprint arXiv:2312.04333*.
- Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for $2+3=?$ on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.
- Qinyuan Cheng, Tianxiang Sun, Xiangyang Liu, Wenwei Zhang, Zhangyue Yin, Shimin Li, Linyang Li, Zhengfu He, Kai Chen, and Xipeng Qiu. 2024. Can ai assistants know what they don't know? *arXiv preprint arXiv:2401.13275*.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.
- Richard A Davis, Keh-Shin Lii, and Dimitris N Politis. 2011. Remarks on some nonparametric estimates of a density function. In *Selected Works of Murray Rosenblatt*, pages 95–100. Springer.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-rl: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the association for computational linguistics: ACL 2023*, pages 1049–1065.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.
- Yue Huang, Chujie Gao, Yujun Zhou, Kehan Guo, Xiangqi Wang, Or Cohen-Sasson, Max Lamparth, and Xiangliang Zhang. 2025b. Position: We need an adaptive interpretation of helpful, honest, and harmless principles. *arXiv preprint arXiv:2502.06059*.
- Manas Joglekar, Jeremy Chen, Gabriel Wu, Jason Yosinski, Jasmine Wang, Boaz Barak, and Amelia Glaese. 2025. Training llms for honesty via confessions. *arXiv preprint arXiv:2512.08093*.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

- Gautam Siddharth Kashyap, Mark Dras, and Usman Naseem. 2025. Too helpful, too harmless, too honest or just right? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29711–29722.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.
- Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: a survey and guideline for evaluating large language models’ alignment. *arXiv preprint arXiv:2308.05374*.
- Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.
- OpenAI. 2025. [gpt-oss-120b](#) [gpt-oss-20b model card](#). Preprint, arXiv:2508.10925.
- Emanuel Parzen. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.
- Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, and 1 others. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the association for computational linguistics: ACL 2023*, pages 13387–13434.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.
- Pranab Sahoo, Prabhath Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. A comprehensive survey of hallucination in large language, image, video and audio foundation models. *arXiv preprint arXiv:2405.09589*.
- Chunlei Xin, Shuheng Zhou, Huijia Zhu, Weiqiang Wang, Xuanang Chen, Xinyan Guan, Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2025. [Sparse latents steer retrieval-augmented generation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4547–4562, Vienna, Austria. Association for Computational Linguistics.
- Fengli Xu, Qianye Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, and 1 others. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.
- An Yang, Anpeng 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*.
- Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*.
- Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. 2025. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. *arXiv preprint arXiv:2508.11408*.
- Yifan Zhang, Jingqin Yang, Yang Yuan, and Andrew Chi-Chih Yao. 2023. Cumulative reasoning with large language models. *arXiv preprint arXiv:2308.04371*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

A Dataset Examples

A.1 Countdown Task

The Countdown task requires constructing a target value using a given set of integers and basic arithmetic operations (+, −, ×, ÷). Each number can be used at most once.

Countdown Example

Numbers: [3, 5, 10, 25]

Target: 151

Golden Answer: $(3 + 10/5) \times 25 = 125$

A.2 Sudoku Task

The Sudoku task requires filling a 9×9 grid such that each row, column, and 3×3 subgrid contains all digits 1-9 without repetition.

Sudoku Example

Puzzle: (0 represents empty cells)

```

5 3 0 | 0 7 0 | 0 0 0
6 0 0 | 1 9 5 | 0 0 0
0 9 8 | 0 0 0 | 0 6 0
-----+-----+-----
8 0 0 | 0 6 0 | 0 0 3
4 0 0 | 8 0 3 | 0 0 1
7 0 0 | 0 2 0 | 0 0 6
-----+-----+-----
0 6 0 | 0 0 0 | 2 8 0
0 0 0 | 4 1 9 | 0 0 5
0 0 0 | 0 8 0 | 0 7 9
    
```

Golden Answer:

```

5 3 4 | 6 7 8 | 9 1 2
6 7 2 | 1 9 5 | 3 4 8
1 9 8 | 3 4 2 | 5 6 7
-----+-----+-----
8 5 9 | 7 6 1 | 4 2 3
4 2 6 | 8 5 3 | 7 9 1
7 1 3 | 9 2 4 | 8 5 6
-----+-----+-----
9 6 1 | 5 3 7 | 2 8 4
2 8 7 | 4 1 9 | 6 3 5
3 4 5 | 2 8 6 | 1 7 9
    
```

B Futile Reasoning Example

B.1 Degenerate Repetition

The example is given numbers: [37, 97, 3, 51, 3, 66, 38, 3], with a target of 275. Due to space constraints, we present excerpts from a representative failure case: the initial reasoning phase (Figure 9) and the final output (Figure 10). The model begins with systematic exploration, attempting multiple combinations. However, after exhausting valid approaches without finding a solution, instead of acknowledging ignorance, the model resorts to exhaustive guessing—repeatedly outputting incorrect expressions while claiming they satisfy the constraints.

Okay, let's see. I need to use all these numbers: 37, 97, 3, 51, 3, 66, 38, 3. Each exactly once, and combine them with +, -, *, / to get exactly 275. Hmm. Let me start by thinking about how to approach this.

First, maybe I should check the total sum of all numbers to see how big it is. Let me add them up. Let's see:

37 + 97 is 134. Then 3 + 51 is 54. Then 3 + 66 is 69. Then 38 + 3 is 41. Wait, no, wait. Wait, the numbers are [37, 97, 3, 51, 3, 66, 38, 3]. So adding all of them:

$$37 + 97 = 134$$

$$134 + 3 = 137$$

$$137 + 51 = 188$$

$$188 + 3 = 191$$

$$191 + 66 = 257$$

$$257 + 38 = 295$$

$$295 + 3 = 298.$$

So total sum is 298. But we need to get 275. So that's 23 less than the total. So maybe some subtraction or division operations can reduce the total by 23. But how?

Figure 9: **Initial Reasoning Phase.** The model systematically explores combinations (e.g., $97 + 66 = 163$, $3 \times 51 = 153$), attempting to construct the target value 275. Early attempts show valid mathematical reasoning but fail to reach the exact target.

✓ Final Correct Expression:

$$(97 + 66 + 38 + 51 + 3) - (37 + 3 + 3)$$

$$255 - 43 = 212 \quad (\text{Incorrect})$$

✓ Final Correct Expression (Correctly Verified):

$$(97 + 66 + 38 + 51 + 37) - (3 + 3 + 3)$$

$$289 - 9 = 280 \quad (\text{Incorrect})$$

✓ Correct Final Expression:

$$(97 + 66 + 38 + 51 + 3) - (37 + 3 + 3)$$

$$255 - 43 = 212 \quad (\text{Incorrect})$$

Figure 10: **Final Output After Degenerate Repetition.** After 50+ failed attempts, the model outputs $(97 + 66 + 51 + 38 + 37) - (3 + 3 + 3) = 280$ while incorrectly asserting it equals 275. This exemplifies hallucination through exhaustive guessing rather than appropriate refusal.

B.2 Endless Generation

We illustrate this failure mode using a Countdown task with numbers [58, 89, 15, 47, 37, 84, 86, 70] and target 168. The endless output is shown below:

Endless Generation Example

We are given numbers: 58, 89, 15, 47, 37, 84, 86, 70. We need to use each exactly once in a single arithmetic expression using only +, -, *, / and parentheses to get exactly 168.

We have 8 numbers. The target is 168. We need to combine them with operations. Possibly we can group them into pairs or combine in some order. Since 168 is not too large compared to some numbers, we might need to subtract or divide to reduce.

...

I'm considering that maybe the solution is not coming easily. Perhaps I should try to work backwards from 168. Write 168 as a combination of some of the numbers with operations, then see if the remaining numbers can be used to adjust.

For example, $168 = 84 * 2$. So if we can get 84 and 2 from separate groups using all numbers, but we already tried that.

$168 = 86 + 82$. 82 can be $47 + 35$, but 35 not.

$168 = 70 + 98$. $98 = 58 + 40$, no.

$168 = 89 + 79$. 79 not.

$168 = 58 + 110$. $110 = 70 + 40$, no.

$168 = 37 + 131$. no.

$168 = 15 + 153$. $153 = 86 + 67$, no.

$168 = 47 + 121$. $121 = 89 + 32$, no.

$168 = 84 + 84$. But only one 84.

$168 = 86 + 84 - 2$. We tried.

$168 = 70 + 98$, $98 = 84 + 14$, but 14 not.

$168 = 89 + 84 - 5$. 5 from $15/3$, but 3 not

[hit 32k tokens limitation with a sudden stop]

For all the evaluations, we conduct inference with a temperature of 0.6. To ensure robustness, we sample 8 responses for each query and report the averaged performance metrics.

C.2 Evaluation Details

For all evaluations on Countdown and Sudoku tasks, we sample 8 responses per question and compute the average score. For general capability evaluation (§5.6), we sample 8 responses per question for AIME 2024 and 4 responses per question for GPQA.

B.3 Specious Reasoning

We illustrate this failure mode using a Countdown task with numbers [8, 12, 74, 88, 74, 67, 10, 6] and target 17. As shown in Figure 11, the model constructs an expression using the number “6” *twice*, violating the constraint that each number may be used at most once. However, during self-verification, the model falsely claims that each number was used only once and asserts the solution is correct. This exemplifies *specious reasoning*. The derivation appears logically sound on the surface but contains a subtle yet critical constraint violation that the model fails to detect.

C Implementation Details

C.1 Training Details

We adopt Group Relative Policy Optimization (GRPO) (Guo et al., 2025) as our underlying reinforcement learning algorithm. For RL (including Standard RL and CaRL variants), we use a learning rate of 1×10^{-6} and train for 3 epochs. During exploration, we employ a group rollout size of 16 with a sampling temperature of 1.0, maintaining a training batch size of 32. For the RFT baseline, we fine-tune the model for 3 epochs with a learning rate of 8×10^{-5} and a global batch size of 32.

Specious Example

<think> Okay, let's tackle this problem. I need to use each of the numbers [8, 12, 74, 88, 74, 67, 10, 6] exactly once with basic operations to get 17. Hmm, that's eight numbers. Let me think about how to combine them.

First, maybe I can look for pairs or groups that can simplify. Since there are so many numbers, perhaps combining some through multiplication or division might help reduce the total. Let me list the numbers again: 8, 12, 74, 88, 74, 67, 10, 6. Wait, there are two 74s. That might be useful or maybe a red herring. Let me see.

[Several Attempts] ...

For example, if I have $(67 - (8 * 6 + (12/6))) + ((74 - 74) * (88 - 10))$?

Let me check. $8 * 6 = 48$, $12/6 = 2$, so $48 + 2 = 50$. $67 - 50 = 17$. Then $74 - 74 = 0$, $88 - 10 = 78$. $0 * 78 = 0$. So total is $17 + 0 = 17$. But wait, I have used all numbers? Let's check:

Used numbers in first part: 67, 8, 6, 12. Then in the second part: 74, 74, 88, 10. Yes! All numbers are used once. So the expression would be:

$(67 - (8 * 6 + (12 / 6))) + ((74 - 74) * (88 - 10))$

Let me check if that equals 17.

First part: $8 * 6 = 48$. $12/6 = 2$. $48 + 2 = 50$. $67 - 50 = 17$.

Second part: $74 - 74 = 0$. $88 - 10 = 78$. $0 * 78 = 0$.

Total is $17 + 0 = 17$. Perfect!

Now, let me check if all numbers are used exactly once:

Numbers used: 67, 8, 6, 12, 74, 74, 88, 10. Yes! All eight numbers are used once.

Now, I need to write this as a single arithmetic expression. Let me make sure the parentheses are correct and the operations are allowed.

The expression is:

$(67 - (8 * 6 + (12 / 6))) + ((74 - 74) * (88 - 10))$

But I need to make sure that all numbers are used once and that the operations are correct. Let me check:

- 67 is used once. - 8 is used once. - 6 is used once. - 12 is used once. - 74 is used twice (the two 74s). - 88 is used once.

- 10 is used once.

Yes, all numbers are used exactly once.

Now, let me write this expression without any extra spaces, using only the allowed symbols. Also, need to ensure that the operations are properly ordered.

The expression would be:

$(67 - (8 * 6 + (12 / 6))) + ((74 - 74) * (88 - 10))$

But I need to check if there are any issues with the order of operations. Let me verify:

Inside the first part: $8 * 6$ is 48, $12 / 6$ is 2, then adding them gives 50. $67 - 50 = 17$.

In the second part: $74 - 74 = 0$, $88 - 10 = 78$, $0 * 78 = 0$.

Adding them gives $17 + 0 = 17$. Correct.

So this expression works. Now, I need to write it in the required format, wrapped in \square .

But let me check for any possible errors. For example, are there any parentheses that need to be adjusted? Let me see:

The expression is:

$(67 - (8 * 6 + (12 / 6))) + ((74 - 74) * (88 - 10))$

Yes, parentheses are correctly placed to ensure the operations are performed in the right order.

So the final answer should be this expression. Let me write it without any extra spaces, as per the rules. Also, ensure that all numbers are used once and that the operations are correct.

So the boxed expression would be:

$(67 - (8 * 6 + (12/6))) + ((74 - 74) * (88 - 10))$

Yes, this should be correct. </think>

$(67 - (8 * 6 + (12/6))) + ((74 - 74) * (88 - 10))$

Figure 11: Supicious Reasoning Example