

The Illusion of Insight in Reasoning Models

Liv G. d’Aliberti
Princeton University
Department of Computer Science
Princeton, NJ, USA
liv.daliberti@princeton.edu

Manoel Horta Ribeiro
Princeton University
Department of Computer Science
Princeton, NJ, USA
manoel@cs.princeton.edu

Abstract

Do reasoning models have “Aha!” moments? Prior work suggests that models like DeepSeek-R1-Zero undergo sudden mid-trace realizations that lead to accurate outputs, implying an *intrinsic* capacity for self-correction. Yet, it remains unclear whether such intrinsic shifts in reasoning strategy actually improve performance. Here, we study mid-reasoning shifts and instrument training runs to detect them. Our analysis spans 1M+ reasoning traces, 80 training checkpoints, 3 reasoning domains, 4 decoding temperatures, and 2 model architectures. We find that reasoning shifts are rare, do not become more frequent with training, and seldom improve accuracy, indicating that they do not correspond to prior perceptions of model insight. However, their effect varies with model uncertainty. Building on this finding, we show that artificially triggering *extrinsic* shifts under high entropy reliably improves accuracy. Our results show that mid-reasoning shifts are symptoms of unstable inference behavior rather than an intrinsic mechanism for self-correction.

1 Introduction

Anecdotal evidence suggests that language models fine-tuned with reinforcement learning exhibit “Aha!” moments—episodes of apparent insight reminiscent of human problem-solving. For example, Guo et al. (2025) highlight mid-trace cues such as “Wait... let’s re-evaluate step-by-step,” which sometimes accompany correct answers. Yet, the nature, frequency, and impact of these events (Fig. 1) remain unclear (Yang et al., 2025).

The existence of “Aha!” moments is linked to whether reasoning models can *intrinsically* self-correct, i.e., revise their reasoning mid-response without external feedback. Model improvements often arise from *extrinsic* mechanisms like verifiers, reward queries, prompting techniques, or external

[Code](#), [Data](#)

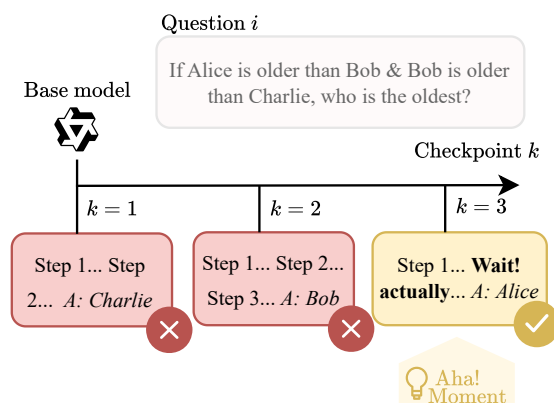


Figure 1: **Anatomy of an “Aha!” Moment.** We illustrate an “Aha!” moment as described in Guo et al. (2025): within a single chain-of-thought, a cue such as “Wait... let’s re-evaluate” marks a shift from an initially failing strategy ($k \in \{1, 2\}$) to one that yields a correct answer (when $k = 3$). The figure also anticipates our methodology: we study “Aha!” moments by systematically GRPO-tuning and annotating the reasoning traces of Qwen2.5 and Llama models.

tools (Lightman et al., 2024; Li et al., 2024a; Zhang et al., 2024). In contrast, intrinsic self-improvement must be inferred from reasoning traces and is arguably more safety-relevant, as it implies that a model can reorganize its reasoning from internal state alone (Tsui, 2025; Liu et al., 2025).

Studying the effect of reasoning shifts is challenging. First, these events may occur (and affect performance) *during* training, yet evaluations are typically conducted only post-training (Zeng et al., 2024; Xia et al., 2025). Second, reasoning models rarely release mid-training checkpoints, limiting longitudinal analyses across the training lifecycle. Third, even when shifts are observed, attributing correctness to a mid-trace change (rather than to general competence or memorization) requires systematically controlled comparisons. *This gap motivates the need for a systematic investigation of whether reasoning shifts reflect genuine insight.*

Present work. Here, we investigate whether mid-trace reasoning shifts (e.g., “Wait... let’s re-evaluate”) signal intrinsic self-correction in reasoning models. Our study is guided by three questions:

RQ1: Do reasoning shifts raise model accuracy?

RQ2: How does the effect of reasoning shifts vary with training stage and decoding temperature?

RQ3: Are reasoning shifts more effective when reasoning models are uncertain?

To answer these, we (i) formalize “Aha!” moments as measurable mid-trace shifts in reasoning that improve performance on problems that were previously unsolved by the model (Yang et al., 2025; Zhou et al., 2025; Hu et al., 2025) (Fig. 2; §3); (ii) curate a diverse evaluation suite (§4) spanning cryptic crosswords (Efrat et al., 2021), mathematical problem-solving (MATH-500) (Lightman et al., 2024), and Rush Hour puzzles (Fogleman, 2018); and (iii) GRPO-tune and annotate the reasoning traces of Qwen2.5 and Llama models (§5).

Our analysis spans 1M+ annotated reasoning traces across hundreds of checkpoint evaluations (10–20 per model/run), 3 domains, 4 temperatures, 2 model sizes, and 2 model architectures, providing a longitudinal view of how mid-trace reasoning evolves during RL fine-tuning. With this setup, we connect shift behavior to both correctness and token-level uncertainty signals (Ton et al., 2025).

Our results show that reasoning shifts are rare (overall $\sim 6.31\%$ of traces) and generally do not improve model accuracy (**RQ1**). We further find that their impact on accuracy does not reliably flip sign across training stages, but varies substantially with decoding temperature (**RQ2**). Finally, we find that spontaneously occurring shifts do not become reliably helpful under high uncertainty; however, *externally triggered* reconsideration under high entropy improves accuracy across benchmarks, including a **+8.41pp** improvement on MATH-500 (and smaller gains on crosswords and Rush Hour) (**RQ3**). Our results are robust across datasets, prompts, and model families.

Contributions. We make three key contributions:

1. **Definition & framework.** We formalize “Aha!” moments as measurable mid-trace shifts and introduce an experimental framework for studying intrinsic self-correction during RL fine-tuning.
2. **Empirical characterization at scale.** Across 1M+ traces spanning domains, temperatures,

training stages, and model families, we show that reasoning shifts are rare and typically coincide with *lower* accuracy, challenging the view that they reflect genuine insight.

3. **Intervention.** We develop an entropy-gated intervention that *induces* reconsideration when models are uncertain, yielding measurable accuracy gains.

2 Related Work

Emergence, scaffolding, and post-training.

Large language models often *appear* to acquire abilities abruptly with scale, including multi-step reasoning and planning (Wei et al., 2022a; Berti et al., 2025), but such discontinuities may reflect evaluation artifacts rather than intrinsic cognitive change (Schaeffer et al., 2023; Shojaee et al., 2025). Many reasoning behaviors also arise only under *extrinsic* scaffolds: Chain-of-Thought prompting (Wei et al., 2022b), zero-shot cues such as “Let’s think step by step” (Kojima et al., 2022), and Least-to-Most prompting (Zhou et al., 2023) elicit intermediate reasoning that models may not produce unaided. Post-training methods such as SFT (Wolfe, 2023), RLHF (Ouyang et al., 2022), and GRPO (Shao et al., 2024) can then reinforce these behaviors, potentially amplifying the appearance of intrinsic ability gains.

Self-correction and “Aha!” moments. Self-correction can arise through extrinsic mechanisms, such as verifiers, tool use, or iterative refinement (Lightman et al., 2024; Li et al., 2024a; Madaan et al., 2023), or through intrinsic shifts that occur without external feedback (Liu et al., 2024). Recent work studies trained self-correction (Kumar et al., 2025), self-correction benchmarks (Tsui, 2025), and mid-inference adjustments (Wu et al., 2024). Claims around DeepSeek-R1 suggest that reward optimization can induce reflection-like artifacts (Guo et al., 2025), while other analyses question whether these shifts reflect genuine insight rather than superficial self-reflection (Liu et al., 2025; Ton et al., 2025). However, prior work has not systematically tested whether RL-trained models exhibit intrinsic “Aha!”-style self-correction *throughout fine-tuning*, nor whether such shifts reliably improve correctness across checkpoints and decoding regimes.

Insight, faithfulness, and reliability. In cognitive psychology, insight is classically defined as an abrupt restructuring of the problem space (Köhler,

1921; Ohlsson, 1987). Recent work seeks analogous signatures in reasoning models, including mid-trace uncertainty spikes and linguistic markers of reconsideration (Ton et al., 2025; Yang et al., 2025), but such markers can misclassify superficial hesitations as insight (Zheng et al., 2023b; Xia et al., 2025). This distinction matters for alignment and faithfulness: reasoning traces support oversight (Uesato et al., 2022; OpenAI, 2023), but ungrounded self-corrections may also obscure objective shifts or deceptive rationales (Su et al., 2025; Baker et al., 2025; Lanham et al., 2023; Zhang et al., 2025). Understanding whether mid-trace shifts reflect genuine correction or uncertainty-driven artifacts is therefore central to evaluating the reliability of reasoning models.

3 Formalizing “Aha!” Moments

We define an “Aha!” moment as a discrete point within a model’s chain-of-thought where the model abandons its initial reasoning strategy and adopts a qualitatively different one that improves performance. We formalize this notion below.

Let $\{f_{\theta_k}\}_{k=0}^K$ denote a sequence of checkpointed reasoning models. At checkpoint k , the model defines a policy $\pi_{\theta_k}(a_t | a_{<t}, q)$ over token actions $a_t \in \mathcal{V}$. A reasoning trace is a trajectory $\tau = (a_1, \dots, a_T)$ whose quality is measured by its correctness $R(\tau)$. For a question q_j , let

$$P_{\theta_k}(\checkmark | q_j) = \mathbb{E}_{\tau \sim \pi_{\theta_k}}[R(\tau)]$$

denote expected correctness. Let $S_{q_j,k}(\tau) \in \{0, 1\}$ indicate whether a mid-trace shift occurs in a sampled trajectory τ at checkpoint k . This binary label is produced by our shift-detection pipeline, which identifies lexical and structural changes in reasoning strategy (detailed in App. B.1). We write $P(S_{q_j,k} = 1)$ for the probability (under $\tau \sim \pi_{\theta_k}$) that a sampled trace contains a detected shift.

Definition 3.1 (“Aha!” Moment). *Let $\delta_1, \delta_2, \delta_3 \in [0, 1]$ be thresholds for prior failure, prior stability, and required performance gains. An “Aha!” moment occurs for (q_j, k) iff:*

1. $\forall i < k, P_{\theta_i}(\checkmark | q_j) < \delta_1$ (Prior failures),
2. $\forall i < k, P(S_{q_j,i} = 1) < \delta_2$ (Prior stability),
3. $P_{\theta_k}(\checkmark | q_j, S_{q_j,k} = 1) - P_{\theta_k}(\checkmark | q_j) > \delta_3$ (Performance gain).

In plain terms, a checkpoint k qualifies as an “Aha!” moment for q_j if: (1) all earlier checkpoints consistently fail on the problem (*prior failures*);

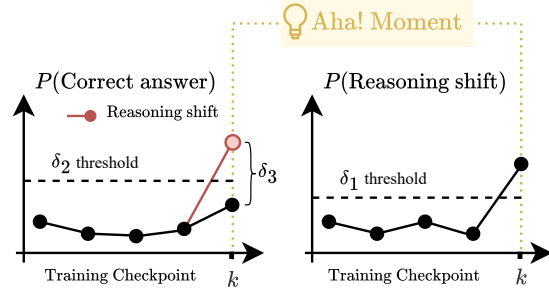


Figure 2: **Schematic of our operational “Aha!” definition.** For a fixed problem q_j (horizontal axis: checkpoint index i), the figure visualizes the three criteria in Def. 3.1. (1) *Prior failures*: empirical correctness $\hat{P}_{\theta_i}(\checkmark | q_j)$ remains below δ_1 at all checkpoints $i < k$. (2) *Prior stability*: the shift rate $\hat{\pi}_i = \Pr[S_{q_j,i} = 1]$ stays below δ_2 for all $i < k$. (3) *Performance gain*: at checkpoint k , correctness on traces *with* a detected shift (red) exceeds correctness over *all* traces (black) by more than δ_3 .

(2) earlier checkpoints show little evidence of mid-trace shifts (*prior stability*); and (3) at checkpoint k , traces containing a detected shift yield a strictly higher correctness rate than traces overall (*performance gain*).¹ Together, these conditions ensure that a detected shift is both *novel* and *beneficial*, preventing superficial or noisy variations from being counted as insight-like events. Figure 2 illustrates this behavior. Algorithm 1 in App. B.1 formalizes the detection procedure.

The thresholds $(\delta_1, \delta_2, \delta_3)$ act as tunable criteria: stricter values prioritize precision by requiring consistent prior failure and rare prior shifts, while looser values increase recall. In our experiments, we select these thresholds on a held-out development slab and validate robustness using bootstrap confidence intervals (App. C.2). In all cases, probabilities such as $P_{\theta_k}(\checkmark | q_j)$ and $P_{\theta_k}(\checkmark | q_j, S_{q_j,k} = 1)$ are estimated from finitely many sampled traces per (q_j, k) .

This definition parallels the classical cognitive characterization of insight: a sudden restructuring of the problem space that enables solution (Jones, 2003; Köhler, 1921; Duncker, 1945; Kaplan and Simon, 1959). Hallmarks of such shifts include explicit self-reflective cues (e.g., “wait,” “let’s reconsider”) and an observable pivot in strategy (Guo et al., 2025). Theoretical accounts such as representational change theory (Ohlsson, 1987), progress monitoring theory (MacGregor et al., 2001), and

¹Formal “Aha!” events are defined over problem–checkpoint pairs (q_j, k) (i.e., a checkpoint-level comparison for a fixed problem), not over individual sampled traces.

Gestalt perspectives on problem-solving (Metcalf, 1987) provide complementary lenses for interpreting analogous shifts in reasoning models.

4 Data

Our evaluation suite spans three complementary reasoning lenses (Appendix A.1, Fig. 5).

Cryptic Xwords. Cryptic Xwords clues hide a wordplay instruction (e.g., anagram, abbreviation, homophone) beneath a misleading surface reading, requiring representational shifts to solve. We train on natural clues from CRYPTONITE (Efrat et al., 2021) and evaluate on a synthetic test set with device-balanced templates (App. A.1), scoring by normalized exact match.

Math. Math word problems test symbolic manipulation and multi-step deduction, with reasoning progress naturally expressed step-by-step. We train on openR1 Math-220k (Hugging Face, 2025) and evaluate on MATH-500 (Lightman et al., 2024), ensuring no train/eval leakage (App. A.1). Answers are scored by normalized exact match.

RHour. We synthetically generate RHOURL sliding-block puzzles, where the goal is to free a target car from a crowded grid by moving obstructing vehicles. We generate balanced 4×4 , 5×5 , and 6×6 boards and evaluate on 6×6 only. Boards are solved optimally via BFS with per-size node caps, discarding timeouts (Fogleman, 2018) (App. A.1). We filter trivial cases and stratify remaining instances into easy (<4 moves), medium (<6), and hard (≥ 6) buckets by solution length.

Each domain offers automatic correctness checks, natural opportunities for mid-trace verification, and structured signals of strategy. All data are in English; dataset sizes and splits are summarized in Table 6, and additional filtering and scoring details are provided in App. A.1. Throughout, we score answers by *normalized exact match* (canonicalizing case, whitespace, and punctuation before exact comparison; App. A.1).

5 Methods

We fine-tune reasoning models with GRPO across evaluation domains (§5.1); collect and annotate reasoning traces during training (§5.2); and estimate model uncertainty to trigger entropy-based interventions (§5.3).

| Model | Domain | Step 0 | After | Step | Δ |
|--------------|--------|--------|-------|------|----------|
| Qwen2.5-1.5B | Xwords | 7.69 | 10.00 | 950 | +2.31 |
| Qwen2.5-1.5B | Math | 31.00 | 35.00 | 950 | +4.00 |
| Qwen2.5-1.5B | RHour | 0.00 | 0.01 | 950 | +0.01 |
| Qwen2.5-7B | Math | 61.60 | 66.40 | 500 | +4.80 |
| Llama 3.1-8B | Math | 40.20 | 48.36 | 500 | +8.16 |

Table 1: **Model coverage and learning progress.** Accuracy at initialization (Step 0) and at the final training checkpoint, along with the absolute gain (Δ). All results are 1-shot evaluations at temperature 0 on the fixed test sets described in §4.

5.1 Models and Training

Motivated by claims of mid-trace “Aha!” behavior in DeepSeek-R1 (Guo et al., 2025), we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as our fine-tuning method. GRPO is an RLHF-style algorithm (Ouyang et al., 2022) that optimizes chain-of-thought generation by comparing groups of sampled completions and extends PPO (Schulman et al., 2017) with group-normalized advantages and KL regularization to a frozen reference policy. Full implementation details appear in App. A.4.

We fine-tune Qwen2.5 (Qwen Team, 2024) and Llama 3.1 (Grattafiori et al., 2024) models on each domain for up to 1,000 steps. Our primary experiments use Qwen2.5-1.5B trained for 1,000 steps (≈ 2.5 – 3 epochs per domain), while larger models (Qwen2.5-7B and Llama 3.1-8B) are evaluated at 500 steps due to compute constraints. To verify that models improve during training, we evaluate at multiple checkpoints and report accuracy at initialization (Step 0) and at the final evaluated checkpoint (Step 950 for the 1.5B runs; Step 500 for 7B/8B). Table 1 summarizes coverage and accuracy gains.

We use lightweight, task-specific prompts that structure reasoning into a `<think>` block and a concise final answer in `<answer>`, with domain-level checks that invite reconsideration (App. A.2). Informed by established strategies—zero-shot CoT, self-consistency, and reflection routines (Kojima et al., 2022; Wang et al., 2023; Madaan et al., 2023; Shinn et al., 2023)—these prompts standardize mid-trace events as reasoning shifts (Def. 3.1; Alg. 1), enabling consistent comparison across models, tasks, and checkpoints.

5.2 Trace Collection and Annotation

We evaluate each model at a fixed cadence of every 50 training steps from initialization (Step 0)

to Step 950 *inclusive* (i.e., checkpoints $k \in \{0, 50, \dots, 950\}$), yielding 20 checkpoints per run. At each checkpoint, we generate $G=8$ completions per problem using a fixed decoding policy (temperature $\{0, 0.05, 0.3, 0.7\}$, top- $p=0.95$). Each completion follows the tag-structured output contract in App. A.2, with private reasoning in `<think>` and a machine-checkable final response in `<answer>`; token budgets and stop criteria are domain-specific and held fixed across checkpoints.

Evaluation sets are held fixed across checkpoints: 500 problems for MATH-500, 130 synthetic clues for Xwords, and 500 6×6 RHour boards. For our Qwen2.5-1.5B models, because each item is evaluated at all 20 checkpoints across $T=4$ temperatures with $G=8$ samples, each run yields 320,000 Math traces, 83,200 Xwords traces, and 320,000 RHour traces. This longitudinal structure allows us to track how mid-trace behavior evolves during RL fine-tuning. We additionally produce 160,000 Qwen2.5-7B traces and 160,000 Llama3.1-8B traces for MATH-500 across 10 checkpoints (Step 0 to Step 450 every 50 steps) to investigate behavior across architecture and model size. Details about our training and evaluation setup appear in App. A.1.

To identify reasoning shifts at scale, we use GPT-4o as an LLM-as-judge. Following evidence that rubric-prompted LLMs approximate human evaluation (Zheng et al., 2023a; Liu et al., 2023; Fu et al., 2023), we supply a fixed rubric that scores each trajectory for (i) correctness, (ii) presence of a mid-trace reasoning shift, and (iii) whether the shift improved correctness.

To reduce known sources of judge bias—position, length, and model identity effects (Wang et al., 2024; Shi et al., 2024; Li et al., 2024b)—we randomize sample order, use split-merge aggregation, enforce structured JSON outputs, and ensemble across prompt variants. We also adopt a conservative error-handling policy: we assign **no shift** when the cue prefilter fails or when the judge output is invalid or low-confidence (App. B.2). Agreement is high: on MATH-500, GPT-4o achieves $\kappa \approx 0.726$ across prompt variants and $\kappa = 0.79$ relative to human majority vote, comparable to expert-expert reliability (Artstein and Poesio, 2008). For additional details, see App. B.3. For qualitative examples, see App. D.6.

5.3 Uncertainty Measure and Intervention

To relate reasoning shifts to model uncertainty, we measure token-level entropy throughout each response. At generation step t , with next-token distribution \mathbf{p}_t , we compute Shannon entropy $H_t = -\sum_v p_t(v) \log p_t(v)$. For each completion, we summarize uncertainty by averaging entropy over the `<think>` and `<answer>` segments (e.g., \bar{H}_{think} and \bar{H}_{ans}), and use these sequence-level scores in downstream analyses.

We also study whether uncertainty can be exploited to improve performance via *artificially triggered* reflection. In the main runs we use the standard cue “Wait, we need to reconsider. Let’s think this through step by step.”, further details covering this setup are discussed in App. B.5. In a follow-up experiment, we test three semantically similar but lexically distinct reconsideration cues (C1–C3). Cue-specific results and regressions are reported in App. C.4.

6 Results

We show that spontaneous reasoning shifts are rare and generally harmful to accuracy, and that formal “Aha!” events are vanishingly rare (RQ1; §6.1); that this negative effect remains stable across training stages but varies systematically with decoding temperature (RQ2; §6.2); and that extrinsically triggered shifts reliably improve performance, especially on high-entropy instances (RQ3; §6.3).

6.1 RQ1: Reasoning Shifts & Model Accuracy

Do reasoning shifts improve accuracy? Before analyzing formal “Aha!” moments, we first consider the broader class of *reasoning shifts*—any mid-trace pivot detected by our annotator, irrespective of whether it satisfies the stricter criteria in Def. 3.1. If such shifts reflected genuine insight, traces containing them should be *more* accurate than those without them.

Across domains, temperatures, and checkpoints for Qwen2.5–1.5B, reasoning shifts remain uncommon (approximately 7.6% of samples; pooling all models/domains yields 6.31%) and are associated with substantially *lower* accuracy: 2.57% for shifted traces versus 16.44% for non-shifted traces, $N=723,200$. A pooled logistic regression of correctness on a shift indicator confirms that this difference is highly significant ($p < 10^{-1198}$).²

²In R-style notation, $\text{correct} \sim \text{shift}$. `correct` is a binary outcome, and `shift` is a binary indicator for an annotator-

| Model | Domain | $\%S_{i,j}$ | $P(\checkmark \mid S_{i,j}=0)$ | $P(\checkmark \mid S_{i,j}=1)$ |
|--------|--------|-------------|--------------------------------|--------------------------------|
| 🦙-1.5B | Xwords | 1.22 | 0.096 | 0.201 |
| | Math | 2.65 | 0.327 | 0.144 |
| | RHour | 14.32 | 0.000 | 0.000 |
| 🦙-7B | Math | 1.50 | 0.661 | 0.282 |
| 🦙-8B | Math | 5.04 | 0.457 | 0.282 |
| 🌀-4o | Math | 2.51 | 0.658 | 0.338 |
| 🦙-R1 | Math | 4.96 | 0.543 | 0.204 |

Table 2: **Shift prevalence and conditional accuracy (RQ1).** $\%S_{i,j}$ gives the fraction of traces labeled as containing a reasoning shift. $P(\checkmark \mid S_{i,j}=0)$ and $P(\checkmark \mid S_{i,j}=1)$ report accuracy without vs. with a detected shift, pooled across all problems, temperatures $\{0, 0.05, 0.3, 0.7\}$, checkpoints, and samples using count-weighted (not simple) averages. Across models and domains, shifted traces are consistently less accurate. 🦙 = Qwen 2.5; 🦙 = Llama 3.1; 🌀 = GPT 4o; 🦙 = DeepSeek R1.

To test whether this pattern is specific to our GRPO-tuned models, we evaluate DeepSeek-R1 and GPT-4o under matched prompting and shift-detection conditions on MATH-500, sweeping decoding temperature $T \in \{0, 0.05, 0.3, 0.7, 1\}$. As shown in Table 19, *canonical* shifts remain a low-base-rate event for both models across temperatures (DeepSeek-R1: 4.70–5.17%; GPT-4o: 1.45–4.40% on MATH-500). Moreover, conditional accuracy given a detected shift does not show consistent improvement: for GPT-4o, $P(\checkmark \mid S=1)$ is lower than $P(\checkmark \mid S=0)$ at every temperature, while for DeepSeek-R1 the shift count is extremely small and estimates are unstable. These results suggest that the low frequency and non-beneficial association of mid-trace shifts is not unique to our training setup, but also appears in external, high-capability reasoning models under the same operational definition. They characterize only the “raw” behavioral signature of mid-trace shifts, independent of any stricter “Aha!” interpretation.

How frequent are formal “Aha!” moments? We now restrict attention to the much smaller subset of events that satisfy *all three* criteria in Def. 3.1. In Fig. 3, by varying $\delta_1, \delta_2 \in \{0, 1/8, 2/8\}$ and fixing $\delta_3 = \epsilon > 0$, we find that formal “Aha!” moments are extremely rare, even with relatively lax constraints. Similar patterns hold for Qwen2.5-7B and Llama3.1-8B (App. C). Pooling every Crossword/Math/RHour checkpoint and temperature, the

labeled reasoning shift. The pooled regression aggregates all test-set traces across Crossword, Math, and RHour.

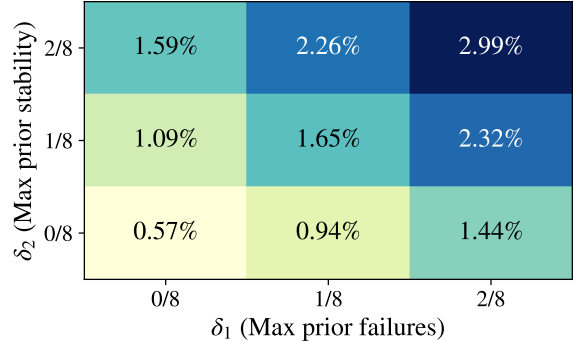


Figure 3: **Prevalence of formal “Aha!” events for Qwen2.5-1.5B (all domains, T=0.7).** Each cell shows the fraction (and count) of problem-checkpoint pairs (q_j, k) that satisfy Def. 3.1 under varying thresholds for prior failures (δ_1) and prior stability (δ_2), with $\delta_3 = \epsilon > 0$. Even under lenient settings, formal “Aha!” events are exceedingly rare. A guide to understanding heatmap calculations in more detail can be found in App. C.1. See App. D.3 for per-domain and per-temperature breakdowns.

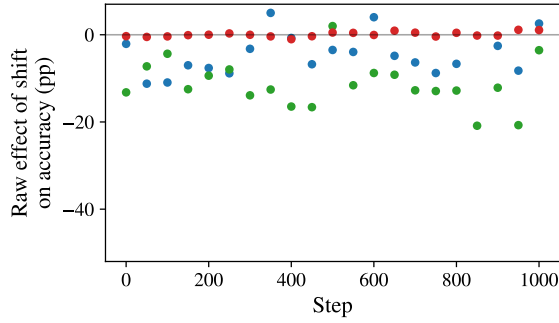
formal detector fires on only 1.79% of samples.

Robustness checks. As surface cues such as “wait” or “actually” often fail to track genuine strategy changes (Zheng et al., 2023b; Xia et al., 2025), and LLM-judge labels may pick up prompt- or position-induced biases (Wang et al., 2024; Shi et al., 2024; Li et al., 2024b), we replicate RQ1 using three detector variants (formal, GPT-based, lexical). All yield the same conclusion; see App. C.6.

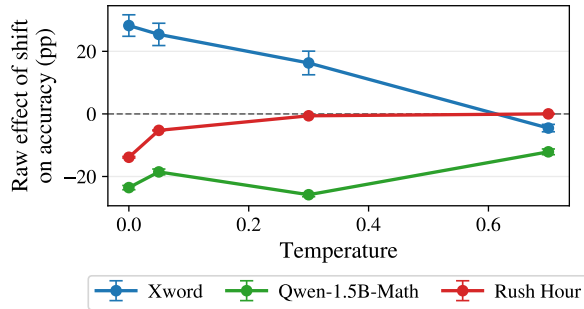
Takeaway. Reasoning shifts are infrequent and generally harmful to accuracy. Further, *formal* “Aha!” moments, which additionally require a performance gain at the pivot, are vanishingly rare. Neither the general phenomenon (reasoning shifts) nor its idealized form (“Aha!” moments) appears to drive problem-solving performance of reasoning models.

6.2 RQ2: Training Stage & Temperature

RQ1 establishes two constraints on “insight-like” behavior: broad reasoning shifts are uncommon and tend to coincide with worse outcomes, while *formal* “Aha!” events are so rare that they contribute little to overall model performance. This raises a natural question: are we simply averaging over regimes where shifts sometimes help and sometimes hurt? We test two plausible sources of heterogeneity: (i) shifts might become more (or less) effective at different *stages* of training; and (ii) their impact might depend on the *decoding temperature* (and thus sampling entropy).



(a) Raw effect of reasoning shifts over training for Qwen2.5-1.5B finetuning across domains (same evaluation at every step).



(b) Raw effect of reasoning shifts over Qwen2.5-1.5B finetuning across domains (same evaluation at every temperature).

Figure 4: **Reasoning shifts across training and temperature (Qwen2.5-1.5B).** We plot the raw accuracy gap $\hat{\Delta} = \hat{p}_{Y|S=1} - \hat{p}_{Y|S=0}$ (pp). (a) At fixed $T = 0.7$, $\hat{\Delta}$ stays near zero or negative across training. (b) Across T , shifts align with correction on *Xword* at lower T , remain harmful on *Math*, and are near-zero on *RHour*.

How does the effect of reasoning shifts vary across training? To test whether the shift–accuracy relationship changes as training progresses, we regress correctness on problem fixed effects, standardized training step, and the shift indicator. We report average marginal effects (AME) with cluster–robust SEs at the problem level.³

At $T=0.7$, we find no evidence that shifts become beneficial later in training. In *Xwords* and *Math* shifts are uncommon ($\%S=2.433$; $\%S=2.166$) and are mildly harmful (AME= -0.0311 , $p=0.02742$; AME= -0.0615 , $p=1.55 \times 10^{-4}$).

In *RHour*, shifts are comparatively frequent ($\%S=11.449$) but have no measurable practical effect on accuracy (AME ≈ 0.0001 , $p \ll 10^{-6}$). Analogous results for $T \in \{0.0, 0.05, 0.3\}$ are reported

³In R-style notation: $\text{correct} \sim C(\text{problem}) + \text{step_std} + \text{shift}$. correct is a binary outcome; $C(\text{problem})$ are problem fixed effects; step_std is the standardized checkpoint index.

in Appendix D.1. Figure 4a echoes this pattern: across checkpoints, shifted traces are not systematically more accurate than non-shifted ones. We repeat robustness checks using alternative detector variants across $T \in \{0, 0.05, 0.3, 0.7\}$ in App. D.5. We observe the same qualitative pattern with the stricter *formal* “Aha!” detector (Appendix D.3), but because it fires on only $\approx 10^{-3}$ of traces at $T=0.7$, estimates are underpowered for fine-grained stage-by-stage heterogeneity; critically, we do not see a consistent late-training transition to positive effects.

How does the effect of reasoning shifts vary with decoding temperature? We next ask whether temperature modulates the relationship between shifts and correctness. We regress correctness on problem fixed effects, standardized temperature, and the shift indicator, aggregating across training steps.⁴

Table 3 summarizes the average association between shifts and correctness while controlling for standardized decoding temperature (via temp_std). Figure 4b shows the corresponding per- T raw pattern. On *Xwords*, the coefficient is positive but not statistically distinguishable from zero (AME= 0.0326 , $p=0.2595$), despite a positive raw contrast $\Delta=+10.54\text{pp}$. On *Math*, shifts are strongly harmful (AME= -0.0831 , $p=2.68 \times 10^{-8}$; $\Delta=-18.35\text{pp}$). On *RHour*, shifts are frequent ($\%S=14.32$) but correctness is extremely low overall; accordingly, the estimated effect is statistically detectable yet numerically negligible (AME ≈ -0.0003 , $p=2.72 \times 10^{-7}$; $\Delta \approx -0.02\text{pp}$).

Raw per-temperature contrasts (Fig. 4b) sharpen the interpretation: on *Xwords*, shifts can coincide with productive correction at low T , but the benefit weakens and may reverse by $T=0.7$. In *Math*, shifts remain harmful across temperatures, though the raw penalty attenuates as T increases. In *RHour*, the curve stays close to zero in magnitude, reflecting the near-zero accuracy regime.

Takeaway. We find that reasoning shifts do not reliably yield higher accuracy across specific training phases or at particular temperatures.

6.3 RQ3: Reasoning Shifts & Uncertainty

The results above (particularly *Xwords*, see Fig. 4b) suggest that decoding temperature may modulate the effect of reasoning shifts: at low T they sometimes align with productive corrections, while at

⁴R-style notation: $\text{correct} \sim C(\text{problem}) + \text{temp_std} + \text{shift}$. temp_std is the standardized decoding temperature.

| (a) Training stage | | | |
|--------------------|---------|-----------------------|-----------------------|
| Metric | Xword | Math | RHour |
| N | 20,800 | 80,000 | 80,000 |
| $\%S$ | 2.433 | 2.166 | 11.449 |
| $\hat{p}_{Y S=1}$ | 0.0731 | 0.1691 | 0.0001 |
| Δ_{pp} | -4.52 | -11.83 | +0.00 |
| AME | -0.0311 | -0.0615 | 0.0001 |
| p | 0.02742 | 1.55×10^{-4} | $\ll 10^{-6}$ |
| (b) Temperature | | | |
| Metric | Xword | Math | RHour |
| N | 83,200 | 320,000 | 320,000 |
| $\%S$ | 1.220 | 2.646 | 14.318 |
| $\hat{p}_{Y S=1}$ | 0.2010 | 0.1435 | 0.0000 |
| Δ_{pp} | +10.54 | -18.35 | -0.02 |
| AME | 0.0326 | -0.0831 | -0.0003 |
| p | 0.2595 | 2.68×10^{-8} | 2.72×10^{-7} |

Table 3: **Effect of detected reasoning shifts on accuracy (Qwen2.5-1.5B)**. For each domain, $\%S$ is the share of samples where the LLM-as-judge detects a shift ($S_{i,j} = 1$); $\hat{p}_{Y|S=1}$ is the empirical accuracy among shifted traces; and Δ_{pp} is the raw accuracy difference (in percentage points) between shifted and non-shifted traces. (a) controls for training step (standardized) at fixed training decoding temperature $T = 0.7$; (b) controls for decoding temperature $T \in \{0.0, 0.05, 0.3, 0.7\}$ while aggregating over steps. AME is the average marginal effect of a shift from a logistic regression with problem fixed effects and cluster-robust SEs; negative values mean that, holding problem and step/temperature fixed, traces with shifts are less likely to be correct. See §6.2 for the full regression specification.

higher T they resemble noise. Because temperature primarily alters sampling entropy rather than the model’s underlying reasoning process (Hinton et al., 2015; Holtzman et al., 2019), this points to a link between shifts and internal uncertainty. We thus ask whether, under high-uncertainty regimens, reasoning shifts are more frequent or become more helpful.

Are reasoning shifts more likely under high uncertainty? To test whether shifts preferentially occur when the model is uncertain, we relate each trace’s reasoning shift indicator to its sequence-level entropy. We pool traces across all decoding temperatures and training checkpoints, and fit a logistic regression of shift prevalence on standardized entropy with problem fixed effects and cluster-robust SEs (clustered by problem).⁵

Pooling all traces across domains (*Xwords*, *Math*,

⁵In R-style notation: $\text{shift} \sim C(\text{problem}) + \text{std_entropy}$. Here shift is a binary indicator for a reasoning shift, $C(\text{problem})$ denotes problem fixed effects, and std_entropy is the within-domain z -scored pass-1 sequence entropy. We estimate a Binomial(logit) GLM with cluster-robust SEs at the problem level.

RHour), we find weak evidence that higher entropy is associated with *fewer* detected shifts on average (OR $\approx 0.77\times$, $\beta = -0.258$, SE = 0.143, $p = 0.070$; 95% CI OR $\in [0.58, 1.02]$; $N = 723,200$). This aggregate pattern masks domain heterogeneity: the entropy–shift association is positive in *Xwords* (OR $\approx 2.05\times$) and *RHour* (OR $\approx 1.19\times$), but negative in *Math* (OR $\approx 0.58\times$). One possible interpretation is that in *Math*, high-entropy generations more often reflect diffuse exploration or verbose “flailing” rather than a discrete mid-trace pivot, so the rare, rubric-qualified shifts concentrate in comparatively lower-entropy traces.

Do reasoning shifts improve performance under high uncertainty?

A natural hypothesis is that when the model is uncertain, a mid-trace pivot might reflect productive self-correction. We test this by stratifying traces into *high-entropy* instances (top 20% within domain) and *low-entropy* instances (bottom 80%), using a fixed entropy threshold per domain. Within each stratum, we estimate the effect of a shift on correctness using a logistic regression with problem fixed effects and controls for continuous entropy and training step, and report the shift coefficient alongside the raw accuracy difference between shifted and non-shifted traces.⁶

Table 4 shows that shifts do *not* become reliably beneficial in the high-entropy regime. In *Math*, shifts remain harmful even among high-entropy traces (raw $\Delta = -7.40\text{pp}$) and are substantially more harmful in the low-entropy majority (raw $\Delta = -22.88\text{pp}$). In *Xwords*, the point estimate in the high-entropy stratum is near zero (raw $\Delta = +0.63\text{pp}$), but shifts are rare and estimates are noisy. In *RHour*, accuracy is near-zero throughout, so estimated effects are statistically detectable due to sample size but negligible in magnitude.

Can artificially triggered reasoning shifts improve performance?

The negative results above suggest that *spontaneous* shifts are not a dependable self-correction mechanism, even when the model is uncertain. High entropy does not cause more spontaneous pivots; rather, it identifies instances where a second-pass reconsideration has higher marginal value. We therefore test an *extrinsically triggered* “forced Aha!” intervention: for each prompt we generate a baseline completion

⁶Within each domain, we split at the 80th percentile of sequence entropy and fit a Binomial(logit) GLM predicting correct from shift with problem fixed effects and covariates. We report both regression and raw contrasts for inter-pretability.

| Metric | Xword | Math | RHour |
|---------------------------------|---------------------------|-----------------------|-------------|
| All traces | | | |
| N | 83,200 | 320,000 | 320,000 |
| Δ (pp) | -6.24 | -19.78 | -0.02 |
| coef(shift) | -1.49 | -1.11 | -22.76 |
| p | 0.123 | 2.25×10^{-7} | ≈ 0 |
| High entropy (top 20%) | | | |
| N | 16,640 | 64,000 | 64,000 |
| Δ (pp) | +0.63 | -7.40 | -0.03 |
| coef(shift) | -0.04 | -0.28 | -22.48 |
| p | 0.904 | 0.739 | ≈ 0 |
| Low entropy (bottom 80%) | | | |
| N | 66,560 | 256,000 | 256,000 |
| Δ (pp) | -10.00 | -22.88 | -0.02 |
| coef(shift) | -28.83 | -1.14 | -22.90 |
| p | 1.33×10^{-46249} | 4.96×10^{-7} | ≈ 0 |

Table 4: **Do spontaneous reasoning shifts help under high uncertainty?** We stratify traces within each domain by sequence entropy (high = top 20% at the within-domain 80th percentile; low = bottom 80%), and compare shifted vs. non-shifted traces. Δ (pp) is the raw accuracy difference $\hat{p}(\checkmark | S=1) - \hat{p}(\checkmark | S=0)$. coef(shift) and p report the shift coefficient and p -value from a logistic regression with problem fixed effects and covariates. Across domains, shifts do not become reliably beneficial in the high-entropy regime.

(Pass 1), then re-query the model under identical decoding settings with an appended reconsideration cue (Pass 2), and compare paired correctness outcomes; see App. C.4 for exact wording and additional analyses.

Table 5 reports paired results aggregated across checkpoints and decoding temperatures. Triggered reconsideration yields a large gain on *Math* (0.322 \rightarrow 0.406; +8.41pp) and a small gain on *Xwords* (+0.45pp), while remaining negligible on *RHour* (+0.01pp) due to its near-zero base rate. The paired “win” counts show that improvements dominate backslides in *Math* (50,574 wrong \rightarrow right vs. 23,500 right \rightarrow wrong), indicating that the effect is not merely random flipping. In contrast, *Xwords* shows near-balanced wins and losses (5,380 vs. 5,004), consistent with a much smaller net gain.

Finally, consistent with uncertainty serving as a useful gate for reflection, Appendix C.4 shows that these gains are amplified on high-entropy instances (Table 29), with complementary regression analysis in Table 25.

Takeaway. Reasoning shifts are a low-base-rate event whose association with entropy varies by domain, and conditioning on uncertainty does not

| Metric | Xword | Math | RHour |
|-----------------------|--------|---------|----------|
| N (paired samples) | 83,200 | 320,000 | 320,000 |
| \hat{p}_{P1} | 0.0970 | 0.3221 | 0.000233 |
| \hat{p}_{P2} | 0.1015 | 0.4062 | 0.000363 |
| Δ (pp) | +0.45 | +8.41 | +0.01 |
| wins (P2 \uparrow) | 5,380 | 50,574 | 100 |
| wins (P1 \uparrow) | 5,004 | 23,500 | 58 |

Table 5: **Forced “Aha!” (triggered reconsideration), sample-level results.** We compare paired outcomes between a baseline generation (Pass 1) and a second generation with an appended reconsideration cue (Pass 2). \hat{p}_{P1} and \hat{p}_{P2} denote accuracies in each pass; Δ (pp) is the percentage-point gain.

reveal a “hidden regime” where spontaneous shifts reliably help. In contrast, artificially triggering reconsideration yields consistent gains, especially for *Math* and especially in the high-entropy tail (App. D.7, Table 29).

7 Discussion and Future Work

We formalize and empirically test intrinsic “Aha!” moments as mid-trace reasoning shifts that appear to reflect sudden insight. We find that such moments are vanishingly rare, and that detected reasoning shifts are typically unhelpful, even under high uncertainty. However, externally triggering reconsideration under high-entropy conditions converts uncertainty into productive reflection, yielding measurable accuracy gains.

These results reframe reasoning shifts not as an emergent cognitive ability, but as a mechanistic behavior: a byproduct of inference dynamics that can nonetheless be harnessed and controlled. Rather than asking whether models have insight, it may be more useful to ask when they can be made to simulate useful reconsideration. This connects uncertainty-aware decoding (Ton et al., 2025; Zhou et al., 2025), process supervision (Uesato et al., 2022; OpenAI, 2023), and self-correction (Madaan et al., 2023; Kumar et al., 2025; Tsui, 2025) around a practical question: how can reflection-like behavior improve reliability without being mistaken for genuine insight?

Future work should investigate process-level supervision that conditions reflection on entropy or confidence estimates, and training objectives that distinguish genuine representational revision from reflection-like language. More broadly, our findings motivate studying why RL post-training improves performance when spontaneous self-correction appears rare.

8 Limitations

While our study offers the first systematic analysis of “Aha!” phenomena in reasoning models, it has several limitations. First, our detection of reasoning shifts relies on *explicit linguistic cues* (e.g., “wait,” “actually”) and measurable plan changes. This makes our estimates conservative: models may undergo unlexicalized representational changes that our detector misses, while some detected shifts may instead reflect superficial hedges. Future work could incorporate hidden-state dynamics or token-level embeddings to better identify implicit restructurings. Second, our evaluation spans three reasoning domains but remains limited to tasks with well-defined correctness signals (math, Xwords, spatial puzzles). Whether similar patterns hold for open-ended reasoning or multi-turn interaction remains an open question. Third, our intervention experiments manipulate model behavior via prompt-level cues rather than modifying training objectives. Thus, while we demonstrate that uncertainty-gated reconsideration can improve accuracy, this does not establish a causal mechanism of internal insight. Extending our analysis to training-time interventions or process supervision would help clarify how reflection-like behaviors emerge and generalize. Finally, as with most large-model studies, our results depend on a small set of families (Qwen, Llama) and inference hyperparameters (e.g., temperature, sampling policy). Broader replications across architectures, decoding methods, larger sizes, and reinforcement-learning setups are necessary to test the generality of our conclusions.

9 Ethical Considerations

Our study analyzes the internal reasoning behavior of large language models and does not involve human subjects or personally identifiable data. All datasets used—MATH-500 (Lightman et al., 2024), CRYPTONITE (Efrat et al., 2021), and synthetic RHOOR puzzles—are publicly available and contain no sensitive content. We follow the terms of use for each dataset and model.

Because our work involves interpreting and modifying model reasoning traces, it carries two potential ethical implications. First, methods that manipulate mid-trace behavior could be misused to steer reasoning models toward undesirable or deceptive outputs if deployed irresponsibly. Our interventions are limited to controlled research settings and designed to study model uncertainty, not

to conceal reasoning or produce persuasive content. Second, interpretability claims about “insight” or “self-correction” risk overstating model understanding. We therefore emphasize that our findings concern statistical behavior, not human-like cognition or consciousness.

Generative AI tools were used to enhance the search for related works and to refine the writing and formatting of this manuscript. We followed the recommendations of Schroeder et al. (2025), who provide guidance for legitimate uses of AI in research while safeguarding qualitative sense-making. Specifically, Claude and ChatGPT were used to identify relevant research papers for the Related Work and Discussion sections (alongside non-generative tools such as Google Scholar and Zotero). Where generative AI was used, the authors certify that they have reviewed, adapted, and corrected all text and stand fully behind the final content.

All model runs, including training and inference, were conducted on NVIDIA A100 GPUs or NVIDIA A6000 GPUs, with resource management, access controls, and energy considerations in place. We estimate the total carbon footprint of all experiments at approximately 110 kg CO₂e, following the methodology of Luccioni et al. (2019).

Acknowledgments

This work was supported by a First-Year Fellowship from the Princeton University Graduate School. We are grateful for computational resources provided by the Beowulf cluster, and we thank the ML Theory Group for generously sharing additional compute. We also acknowledge the support of the Center for Information Technology Policy (CITP) and the Department of Computer Science at Princeton University. We thank our volunteer annotators and the broader Princeton community. Special thanks to Cannoli, our lab dog, and to the musical artist, DoeChii.

References

- Ron Artstein and Massimo Poesio. 2008. [Inter-annotator agreement for computational linguistics](#). *Computational Linguistics*, 34(4):555–596.
- Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. 2025. [Monitoring reasoning models for misbehavior and the risks of promoting obfuscation](#). *Preprint*, arXiv:2503.11926.
- Leonardo Berti, Flavio Giorgi, and Gjergji Kasneci. 2025. [Emergent abilities in large language models: A survey](#). *Preprint*, arXiv:2503.05788.
- Karl Duncker. 1945. *On Problem-Solving*, volume 58. Psychological Monographs.
- Avia Efrat, Uri Shaham, Dan Kilman, and Omer Levy. 2021. [Cryptonite: A cryptic crossword benchmark for extreme ambiguity in language](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4186–4192, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Michael Fogleman. 2018. Solving Rush Hour, the puzzle. <https://www.michaelfogleman.com/rush/>. Accessed: 2025-08-16.
- Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. [Gptscore: Evaluate as you desire](#). *Preprint*, arXiv:2302.04166.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 538 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in LLMs through reinforcement learning](#). *Nature*, 645(8081):633–638. Published online: 17 September 2025.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the knowledge in a neural network](#). *Preprint*, arXiv:1503.02531. Often cited as the NIPS 2014 Deep Learning Workshop note.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. [The curious case of neural text degeneration](#). *Preprint*, arXiv:1904.09751.
- Zhiyuan Hu, Yibo Wang, Hanze Dong, Yuhui Xu, Amrita Saha, Caiming Xiong, Bryan Hooi, and Junnan Li. 2025. [Beyond ‘aha!’: Toward systematic meta-abilities alignment in large reasoning models](#). *arXiv preprint arXiv:2505.10554*.
- Hugging Face. 2025. [Open r1: A fully open reproduction of deepseek-r1](#).
- Gary Jones. 2003. [Testing two cognitive theories of insight](#). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29(5):1017–1027.
- Charles A. Kaplan and Herbert A. Simon. 1959. The role of verbalization in problem solving: The use of the duncker candle problem. *Quarterly Journal of Experimental Psychology*, 11(3):168–174.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, *Main Conference Track*.
- Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2025. [Training language models to self-correct via reinforcement learning](#). In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR 2025)*.
- Wolfgang Köhler. 1921. *Intelligenzprüfungen an Menschenaffen: Mit einem Anhang zur Psychologie des Schimpansen*, 2 edition. Heidelberger Taschenbücher, Band 134. Springer Berlin Heidelberg, Berlin. Accessed: 2025-07-08.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilè Lukōšiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. [Measuring faithfulness in chain-of-thought reasoning](#). *Preprint*, arXiv:2307.13702.
- Junlong Li, Jinyuan Wang, Zhuosheng Zhang, and Hai Zhao. 2024a. [Self-prompting large language models for zero-shot open-domain QA](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 296–310, Mexico City, Mexico. Association for Computational Linguistics.
- Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024b. [Split and merge: Aligning position biases in LLM-based evaluators](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11084–11108, Miami, Florida, USA. Association for Computational Linguistics.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe.

2024. [Let’s verify step by step](#). In *International Conference on Learning Representations (ICLR 2024)*. ICLR 2024 poster.
- Dancheng Liu, Amir Nassereldine, Ziming Yang, Chenhui Xu, Yuting Hu, Jiajie Li, Utkarsh Kumar, Changjae Lee, Ruiyang Qin, Yiyu Shi, and Jinjun Xiong. 2024. [Large language models have intrinsic self-correction ability](#). *Preprint*, arXiv:2406.15673.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-eval: Nlg evaluation using gpt-4 with better human alignment](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.
- Zichen Liu, Changyu Chen, Wenjun Li, Tianyu Pang, Chao Du, and Min Lin. 2025. [There may not be aha moment in rl-zero-like training — a pilot study](#). SEA AI Lab Blog. Published February 07, 2025.
- Sasha Luccioni, Victor Schmidt, Alexandre Lacoste, and Thomas Dandres. 2019. [Quantifying the carbon emissions of machine learning](#). In *NeurIPS 2019 Workshop on Tackling Climate Change with Machine Learning*.
- James N. MacGregor, Thomas C. Ormerod, and Edward P. Chronicle. 2001. Information processing and insight: A process model of performance on the nine-dot and related problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27(1):176–201.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.
- Janet Metcalfe. 1987. Insight and metacognition. *Cognition*, 25(1-2):123–136.
- Stellan Ohlsson. 1987. Insight problem solving: A case for restructuring. *The Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13(4):601–626.
- OpenAI. 2023. [Improving mathematical reasoning with process supervision](#). Accessed 2025-07-08.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, Main Conference Track.
- Qwen Team. 2024. [Qwen2.5: A party of foundation models](#). Qwen Blog.
- Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. [Are emergent abilities of large language models a mirage?](#) In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, Main Conference Track.
- Hope Schroeder, Marianne Aubin Le Quéré, Casey Randazzo, David Mimno, and Sarita Schoenebeck. 2025. [Large language models in qualitative research: Uses, tensions, and intentions](#). In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI ’25, New York, NY, USA. Association for Computing Machinery.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *Preprint*, arXiv:1707.06347.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Lin Shi, Weicheng Ma, and Soroush Vosoughi. 2024. [Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by LLMs](#). *Preprint*, arXiv:2406.07791.
- Noah Shinn, Yohei Koyama, Arka Singh, Ludwig Kirsch, Jan Leike, and Aditya Gopalan. 2023. [Reflexion: Language agents with verbal reinforcement learning](#). *arXiv preprint arXiv:2303.11366*.
- Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. [The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity](#). *Preprint*, arXiv:2506.06941.
- Hang Su, Jun Luo, Chang Liu, Xiao Yang, Yichi Zhang, Yinpeng Dong, and Jun Zhu. 2025. [A survey on autonomy-induced security risks in large model-based agents](#). *Preprint*, arXiv:2506.23844.
- Jean-Francois Ton, Muhammad Faaiz Taufiq, and Yang Liu. 2025. [Understanding chain-of-thought in llms through information theory](#). In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*.
- Ken Tsui. 2025. [Self-correction bench: Uncovering and addressing the self-correction blind spot in large language models](#). *Preprint*, arXiv:2507.02778. ArXiv v2 (last revised 4 Oct 2025).
- 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.

- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. [Large language models are not fair evaluators](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *International Conference on Learning Representations (ICLR)*.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. [Emergent abilities of large language models](#). *Transactions on Machine Learning Research*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Cameron R Wolfe. 2023. Understanding and using supervised fine-tuning (sft) for large language models. <https://cameronrwolfe.substack.com/p/understanding-and-using-supervised>.
- Zhenyu Wu, Qingkai Zeng, Zhihan Zhang, Zhaoxuan Tan, Chao Shen, and Meng Jiang. 2024. [Large language models can self-correct with key condition verification](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12846–12867, Miami, Florida, USA. Association for Computational Linguistics.
- Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. [Evaluating mathematical reasoning beyond accuracy](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27723–27730.
- Shu Yang, Junchao Wu, Xin Chen, Yunze Xiao, Xinyi Yang, Derek F. Wong, and Di Wang. 2025. [Understanding aha moments: from external observations to internal mechanisms](#). *Preprint*, arXiv:2504.02956.
- Zhongshen Zeng, Yinhong Liu, Yingjia Wan, Jingyao Li, Pengguang Chen, Jianbo Dai, Yuxuan Yao, Rongwu Xu, Zehan Qi, Wanru Zhao, Linling Shen, Jianqiao Lu, Haochen Tan, Yukang Chen, Hao Zhang, Zhan Shi, Bailin Wang, Zhijiang Guo, and Jiaya Jia. 2024. [Mr-ben: A meta-reasoning benchmark for evaluating system-2 thinking in llms](#). In *Advances in Neural Information Processing Systems*, volume 37.
- Qingjie Zhang, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, Minlie Huang, Ke Xu, Hewu Li, Liu Yan, and Han Qiu. 2025. [Understanding the dark side of LLMs’ intrinsic self-correction](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27066–27101, Vienna, Austria. Association for Computational Linguistics.
- Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. 2024. [Small language models need strong verifiers to self-correct reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15637–15653, Bangkok, Thailand. Association for Computational Linguistics.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). *Preprint*, arXiv:2306.05685.
- Mingyu Zheng, Hao Yang, Wenbin Jiang, Zheng Lin, Yajuan Lyu, Qiaoqiao She, and Weiping Wang. 2023b. [Chain-of-thought reasoning in tabular language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11006–11019, Singapore. Association for Computational Linguistics.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. [Least-to-most prompting enables complex reasoning in large language models](#). In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023)*.
- Kaiwen Zhou, Xuandong Zhao, Gaowen Liu, Jayanth Srinivasa, Aosong Feng, Dawn Song, and Xin Eric Wang. 2025. [Safekey: Amplifying aha-moment insights for safety reasoning](#). In *ICML 2025 Workshop on Reliable and Responsible Foundation Models (R2-FM)*. Workshop paper.

A Experimental Setup and Data

This first part of the appendix collects the *reproducibility scaffolding* for our experiments: what data we train and evaluate on, what prompts and output contracts we impose, and how GRPO training is configured. We begin with dataset sizes and domain-specific preprocessing details (§A.1). We then provide the verbatim system-level prompts used in each domain, including the shared `<think>/<answer>` formatting requirements and domain-specific guardrails (§A.2). Finally, we summarize the GRPO training setup and per-domain hyperparameters (§A.4; App. A.3). All of these components are held fixed across checkpoints unless explicitly noted, so that differences reported in the main text reflect changes in model state rather than instruction drift or evaluation artifacts.

A.1 Dataset Details

Table 6 summarizes dataset sizes, splits, and evaluation coverage for all three reasoning domains. Figure 5 provides example instances and domain-specific reasoning structures. We include additional details here for reproducibility.

Cryptic Xwords. We use the CRYPTONITE corpus for training (Efrat et al., 2021) and generate synthetic evaluation clues using device-balanced templates. All answers are normalized (uppercase, punctuation-stripped) before exact-match scoring.

Math. The training distribution is openR1 Math-220k (Hugging Face, 2025); evaluation is on the MATH-500 benchmark (Lightman et al., 2024). Normalization removes \LaTeX wrappers, whitespace, and trivial formatting differences (e.g., ‘1/2’ vs. ‘ $\frac{1}{2}$ ’) before exact match.

RHour. We generate balanced 4×4 , 5×5 , and 6×6 puzzles and evaluate on 6×6 only. Puzzles are solved optimally via BFS with per-size node caps; unsolved or degenerate boards are removed (Fogleman, 2018). Solutions are canonicalized before comparison.

Data release. Code to regenerate the synthetic Cryptic Xwords evaluation set and the selected RHour puzzles is included in our repository under `data/`. We also release the exact evaluation subsets on Hugging Face: `od2961/rush4-5-6-balanced` and `od2961/Guardian-cryptonite-official-split`.

| Domain | Train (N) | Eval (N) |
|----------------|---------------|--------------|
| Cryptic Xwords | 50,000 | 130 |
| Math | 220,000 | 500 |
| RHour | 180,000 | 500 |

Table 6: **Dataset sizes.** Training instances are natural clues (Xwords), problems (Math), and boards (RHour); evaluation uses synthetic clues for Xwords.

A.2 System-Level Prompts

Design goals. Our system prompts serve two purposes: (i) *scaffold* domain-appropriate reasoning with verifiable intermediate structure, and (ii) *standardize* outputs so they are machine-checkable and comparable across checkpoints. Across all domains we therefore (a) separate private reasoning from the final response with explicit tags (`<think>/</think>` and `<answer>/</answer>`), (b) enforce domain-specific *guardrails* (e.g., enumeration and letter accounting for cryptics; canonical forms for mathematics; regex-constrained action sequences for RHour), and (c) build in a lightweight *self-correction loop* that triggers targeted reconsideration when a check fails. The prompts below were held fixed across checkpoints and temperatures (unless noted), ensuring that any changes we observe arise from the model state rather than instruction drift.

Common scaffolding (all domains). We ask models to reason entirely inside `<think>` and to place the *final* object to be graded inside `<answer>` *only*. Tag separation lets us (1) compute reasoning-shift features on the private trace without leaking them into the final output, and (2) apply exact validators to `<answer>`. To avoid verbosity that can mask errors, prompts specify concise but complete derivations, a token budget, and deterministic formatting.

Cryptic Xwords. The Xwords prompt encodes established solving practice: *device triage* (anagram, container, reversal, hidden, *etc.*) with quoted indicators, followed by a two-part parse (**definition** and **wordplay**) and two hard checks: exact enumeration and exact letter accounting. This combination suppresses common failure modes such as defaulting to anagrams without a bona fide anagrind or silently dropping letters in charades. The reconsideration loop is narrow: it requires explaining *why* the current attempt fails before proposing an alternative *device/parse*. We found this prevents thrashing while still eliciting genuine mid-trace piv-

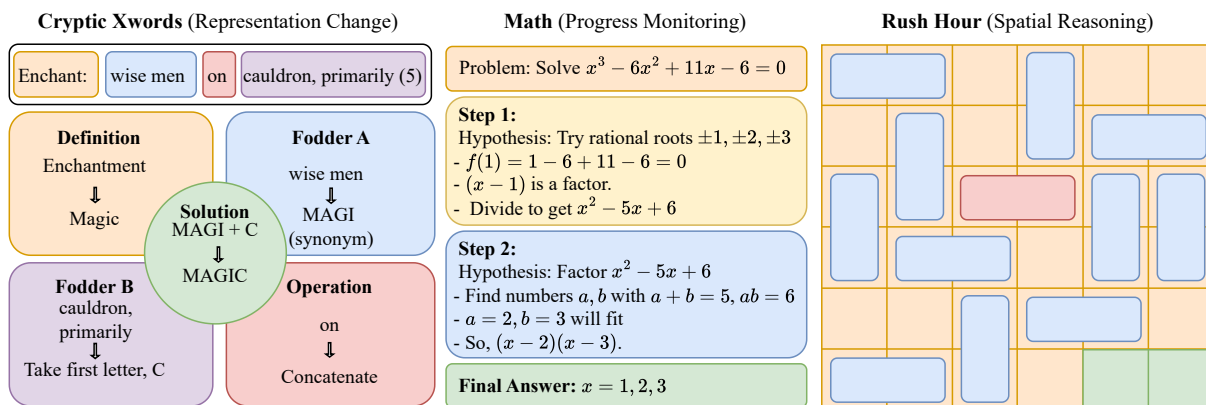


Figure 5: **Three reasoning lenses and example instances.** Each row illustrates one evaluation domain and how it instantiates the three “reasoning lenses” introduced in §4. *Left (representation change)*: a cryptic Xwords clue with definition and wordplay; shifts correspond to re-parsing the clue (e.g., switching from anagram to charade or hidden-word). *Center (progress monitoring)*: a math problem with explicit chain-of-thought and checks; shifts occur when the model abandons an inconsistent derivation and restarts with a new method. These domains form complementary testbeds for studying when mid-trace shifts (our “Aha!” events; Def. 3.1) co-occur with changes in uncertainty and accuracy. *Right (spatial manipulation)*: a RHour puzzle requiring a planned sequence of legal moves; mid-trace shifts reflect abandoning one move plan for another.

ots when a better device is available. Examples in the prompt illustrate (i) a hidden, (ii) an anagram, and (iii) a charade—covering the most frequent device classes in our corpus.

Math. The math prompt stresses (i) *goal/given-s/methods* triage, (ii) exact, symbolic manipulation with canonical forms (fractions, radicals, π , e), and (iii) end-of-proof *validation* (domain, extraneous roots, simplification). We explicitly specify what to output when a problem is infeasible (“NO SOLUTION”) or underdetermined (“I DON’T KNOW”), which reduces hallucinated specificity. The tag split is enforced more strictly here to prevent the final answer from appearing in `<think>` and to keep `<answer>` parsable for grading and correctness metrics. The 750-token cap preserves headroom for multi-step derivations while discouraging digressions that add entropy without improving validity.

RHour. For RHour puzzles, the prompt formalizes the interface between natural-language reasoning and a discrete planner. Inputs are normalized ($N \times N$ board, row-major encoding), and `<answer>` must match a regular expression of move tokens ($^{\wedge}[A-Z][<^{\wedge}v]\d+(\wedge, [A-Z][<^{\wedge}v]\d+)*\wedge$). We add two verifiability clauses: (i) the sequence must be *optimal* (minimum length), with lexicographic tie-breaks to canonicalize multiple optimal plans; and (ii) applying the sequence must achieve the goal (A exits) in exactly the declared number of

moves. These guardrails allow us to reject superficially plausible but illegal or suboptimal sequences and to attribute improvements to better internal search rather than looser grading.

Configs and model release. The full configs, exactly as used for training, are available in our repository under `recipes/`. We also release all trained models (including checkpoints) on Hugging Face, listed in Table 7.

A.3 Prompt Robustness & Evaluation

Robustness to system-prompt wording. To probe how sensitive performance is to *system-prompt wording*, we evaluated $K=5$ paraphrased system prompts for Qwen2.5-1.5B (Open-R1 GRPO, trained on Math220k) on MATH-500 at decoding temperature $T=0$, using randomized item order and a short prefilter on input length (350 characters; Table 8). For each epoch and prompt variant, we compute standard test accuracy and then summarize the distribution across the five prompts. Across variants, accuracy changes only modestly, and the qualitative conclusions reported in the main text are unchanged. We therefore report main results using the canonical system prompt shown above, and use the prompt ensemble only to quantify prompt-induced variance.

Reproducibility and evaluation. Prompts are released verbatim below. We use the same decoding policy across checkpoints (temperature, top- p , stop criteria), cache RNG seeds, and reject outputs that

System Prompt — Cryptic Xwords

```
You are an expert cryptic-xword solver. Do this (repeat until fully consistent):
A) DEVICE TRIAGE
  • List plausible devices from {anagram, container, reversal, hidden, charade, deletion,
  ↪ homophone, double def, &lit, letter selection, substitution, abbr}.
  • For each, quote the indicator word(s). Reject with a reason.
B) PARSE
  • Mark the definition (start or end).
  • Mark the wordplay with exact fodder + operations.
C) CHECKS
  • Enumeration must match exactly.
  • Letter accounting must be exact (anagram multiset or stepwise build).
D) DECIDE
  • Pick the candidate best matching definition, indicator(s), and enumeration.
  • Do NOT assume anagram without a clear anagrind and fully used fodder.
E) RECONSIDER (if any check fails)
  • Begin the next \tagtt{think} with: "Wait, we need to reconsider. Let's think this through
  ↪ step by step."
  • Say why it failed, then re-run A-D with an alternative device/parse.
FORMAT (no deviations):
  • Reasoning only in \tagtt{think}$\cdots$\etagtt{think}
  • Final entry ONLY (UPPER-CASE) in \tagtt{answer} $\cdots$ \etagtt{answer}
Clue: Close, as seen in plaNET EARly (4)
\tagtt{think}Device: HIDDEN; indicator "as seen in".
Def: "Close". Wordplay: hidden in "plaNET EARly" -> NEAR.
Enumeration: (4) OK.\etagtt{think}
\tagtt{answer} NEAR \etagtt{answer}
Clue: Mix TEA for a hot drink (3)
\tagtt{think}Device: ANAGRAM; indicator "Mix". Fodder TEA -> TEA.
Def: "a hot drink". Accounting exact; (3) OK.\etagtt{think}
\tagtt{answer} TEA \etagtt{answer}
Clue: Shoe liner at home on fish (6)
\tagtt{think}Device triage: {hidden ? ("on" is not a hidden indicator), anagram × (no anagrind),
charade ✓ ("at home"=IN, "on"=next to), homophone ×, ...}
Attempt (HIDDEN) rejected: no indicator; also hidden spans don't give (6).
Candidate attempt (wrong path): - fails enumeration/indicator, so we must rethink.
Re-evaluate as CHARADES: IN ("at home") + SOLE ("fish") -> INSOLE.
Accounting: INSOLE letters: I N S O L E (6). Definition "Shoe liner" fits. Enumeration (6) OK.\
↪ etagtt{think}
\tagtt{answer}INSOLE\etagtt{answer}
```

Figure 6: **System Prompt — Cryptic Xword.** Verbatim system-level prompt used for the Xwords domain.

violate the format contracts before computing task-specific rewards. This protocol ensures that improvements in correctness or in “Aha!” prevalence reflect changes in the model’s internal state rather than changes in instructions or graders. Figures 6, 7, 8 show our verbatim system level prompts.

A.4 Model Training (GRPO Setup)

Overview. We fine-tune instruction models (Qwen 2.5–1.5B, Qwen 2.5–7B, Llama 3.1–8B) with Group Relative Policy Optimization (GRPO) (Shao et al., 2024), using task-specific, tag-constrained prompts that place private reasoning in `<think>` and a single, machine-checkable response in `<answer>` (See App. §A.2).

Rollout + training architecture. We use the OpenR1 GRPO trainer (Hugging Face, 2025) with a vLLM inference server for on-policy rollouts and accelerate+DeepSpeed ZeRO-3 for training. A dedicated GPU hosts vLLM; the remaining GPUs run GRPO. Mixed precision is bf16 for training; vLLM runs fp16. DeepSpeed is configured with ZeRO-3, CPU offload for parameters/optimizer, and overlap-comm; the accelerate configuration uses four to seven processes depending on available devices.

Domain-specific reward functions. All rewards are per-sample and clipped to $[0, 1]$.

• **Xwords.** Exact match on the inner `<answer>` (strict 0/1) plus two shaping signals: (i) a tiny “contains as a standalone word” bonus,

System Prompt — Math

You are an expert *mathematics* problem-solver.
Every time you receive a problem you must:

- Analyze it thoroughly.
 - Pinpoint the *goal* (what quantity/set/form is requested).
 - Pinpoint the *givens/constraints* (domains, integrality, non-negativity, geometric \hookrightarrow conditions).
 - Choose the *methods* to apply (algebraic manipulation, factorization, inequalities, \hookrightarrow counting, modular arithmetic, geometry, calculus, etc.).
 - Write out the full derivation that leads to the final result.
- Check that the result satisfies all original constraints (no extraneous roots, correct domain, \hookrightarrow simplified form, exact arithmetic).
- Respond in *exactly* the tag-based format shown below - no greeting, no commentary outside \hookrightarrow the tags.
 - The final answer goes inside `\taggt{answer}` *only*.
 - Use *exact* math (fractions, radicals, π , e). Avoid unnecessary decimals.
 - Canonical forms: integers as plain numbers; reduced fractions a/b with $b>0$; simplified \hookrightarrow radicals; rationalized denominators; sets/tuples with standard notation; intervals in \hookrightarrow standard notation.
 - If there is *no solution*, write NO SOLUTION. If the problem is *underdetermined*, write \hookrightarrow I DON'T KNOW.
- You have a hard cap of *750 output tokens*. Be concise but complete.

TAG TEMPLATE (copy this shape for every problem)

```
\taggt{think}
YOUR reasoning process goes here:
1. quote the relevant bits of the problem
2. name the mathematical tool(s) you apply
3. show each intermediate step until the result is reached
If you spot an error or an unmet constraint, iterate, repeating steps 1-3 as many
times as necessary until you are confident in your result. Finish by verifying the
result satisfies the original conditions exactly (substitution/checks).
\taggt{think}
\taggt{answer}
THEANSWER
\taggt{answer}
```

Figure 7: **System Prompt — Math.** Verbatim system-level prompt used for math.

scaled by a *tag factor* (fraction of `<think>`, `</think>`, `<answer>`, `</answer>` present), and (ii) a “Xwords accuracy” term that linearly ramps with `<think>` length and is multiplied by the same tag factor; optional enumeration checks reject length mismatches.

- **Math.** Requires the full tag template; the gold and predicted `<answer>` are canonicalized (LaTeX/math normalization) and compared for exact equality (0/1).
- **RHour.** Composite score combining *exact* (canonical token sequence), *prefix* (longest common prefix vs. gold), *solve* (shorter optimal solutions \uparrow), and a planning heuristic Φ (distance/blockers decrease \uparrow); when a board is provided we legality-check and simulate moves, otherwise a gold-only variant supplies solve/prefix shaping. Defaults (used here): $w_{\text{exact}}=0.65$, $w_{\text{solve}}=0.20$, $w_{\text{prefix}}=0.10$, $w_{\Phi}=0.05$.

Optimization and KL control. Across runs we use cosine LR schedules with warmup, clipped advantages/values, and KL control targeting $\text{KL}_{\text{target}} \approx 0.07$ via an adaptive coefficient (β) with horizon 50k and step size 0.15; value loss weight 0.25; PPO/GRPO clip ranges 0.05–0.10 depending on run; horizon 1024; $\gamma=0.99$, $\lambda_{\text{GAE}}=0.95$.

Prompt templates and budgets. We use fixed system prompts per domain that enforce exact formatting (no deviation), encourage compact reasoning, and cap `<think>/<answer>` token budgets. This standardization lets the rewards remain reliable and comparable across checkpoints and temperatures.

Per-domain GRPO configurations. Table 9 summarizes only the run-level choices that differ by domain/model; all other defaults follow the overview above.

System Prompt — RHour

You are an expert RHour ($\{N\} \times \{N\}$) solver.

TASK

- Input fields are provided in the user message:
 - Board (row-major string with 'o', 'A', 'B'..'Z', optional 'x')
 - Board size (e.g., 4x4 or 5x5 or 6x6)
 - Minimal moves to solve (ground-truth optimum), shown for training
- OUTPUT exactly ONE optimal solution as a comma-separated list of moves.
 - Move token = <PIECE><DIR><STEPS> (e.g., A>2,B<1,Cv3)
 - Directions: '<' left, '>' right, '^' up, 'v' down
 - No spaces, no prose, no extra punctuation or lines.

GOAL

- The right end of 'A' must reach the right edge.

OPTIMALITY & TIE-BREAKS

- Your list must have the minimal possible number of moves.
- If multiple optimal sequences exist, return the lexicographically smallest comma-separated sequence (ASCII order) after normalizing tokens.

VALIDATION

- Tokens must match: $^[A-Z][<>^v]\d+(\,[A-Z][<>^v]\d+)*\$$
- Each move respects vehicle axes and avoids overlaps with walls/pieces.
- Applying the full sequence reaches the goal with exactly {moves} moves.

IF INCORRECT / UNVALIDATED

- Repeat reasoning process, iterating until correct.

FORMAT

- Answer in the following way:

```
\tagtt{think}
Your reasoning
\tagtt{think}
\tagtt{answer}
A>2,B<1,Cv3
\tagtt{answer}
```

Figure 8: **System Prompt — RHour.** Verbatim system-level prompt used for RHour puzzles.

| Model | Domain | Hugging Face repository |
|---------------|--------|---|
| Llama 3.1-8B | Math | https://huggingface.co/od2961/Llama-8B-Open-R1-GRPO-math-v1 |
| Qwen 2.5-1.5B | Xwords | https://huggingface.co/od2961/Qwen2.5-1.5B-Open-R1-GRPO-Crosswords-v03 |
| Qwen 2.5-1.5B | RHour | https://huggingface.co/od2961/Qwen2.5-1.5B-Open-R1-GRPO-carpark-v1 |
| Qwen 2.5-7B | Math | https://huggingface.co/od2961/Qwen2.5-7B-Open-R1-GRPO-math-7b |
| Qwen 2.5-1.5B | Math | https://huggingface.co/od2961/Qwen2.5-1.5B-Open-R1-GRPO-math-v1 |

Table 7: **Released GRPO-trained checkpoints.** Public Hugging Face repositories containing trained models and intermediate checkpoints used in this work.

DeepSpeed/Accelerate settings. We train with ZeRO-3 (stage 3), CPU offload for parameters and optimizer, and the standard single-node launcher; the provided accelerate config sets bf16 mixed precision and num_processes according to available training GPUs (vLLM occupies a dedicated device).

Operational notes. Jobs were launched under Slurm on 8-GPU nodes, using a mix of NVIDIA A100 and RTX A6000 GPUs. For the **Qwen2.5-1.5B** runs, we reserve **one** GPU for vLLM rollouts and use the remaining GPUs for accelerate training. For the larger **Qwen2.5-7B** and **Llama 3.1-8B** runs, we reserve **two** GPUs for vLLM to sup-

port higher-throughput rollouts, with the remaining GPUs used for training. Per-run environment/caching settings and health-checks follow the batch script. The trainer logs per-step KL, policy/critic losses, and gradient norms; checkpoints are saved every 50 steps and pushed locally/HF Hub per config.

B Shift Detection, “Aha!” Detection, and Annotation

This appendix details the pipeline used to (i) label mid-trace reasoning shifts in individual generations and (ii) operationalize *formal* “Aha!” events as a checkpoint-level phenomenon. We first present our formal “Aha!” detector, which combines prior-

| Epoch | Mean accuracy | Std. across prompts | Range (min, max) |
|---------------|---------------|---------------------|------------------|
| 0 (pre) | 31.8 | 0.8 | (30.8, 33.2) |
| 1 (ckpt 500) | 38.8 | 1.2 | (36.6, 40.0) |
| 2 (ckpt 1000) | 38.3 | 1.0 | (36.8, 39.8) |
| 3 (final) | 40.2 | 0.7 | (39.5, 41.4) |

Table 8: Accuracy stability across system-prompt paraphrases for MATH-500 (Qwen-1.5B). Each row summarizes accuracy over $K=5$ system prompts at temperature 0, using the same test problems and a prefilter window of 350 input characters.

| Task / Model | Dataset | Key GRPO settings |
|----------------------------|--------------------------------------|--|
| Math (1.5B Qwen) | OpenR1–Math–220k | LR 5×10^{-6} ; bs/dev=8; grad_acc=64; epochs=3; num_gens=4; max_prompt=512; max_completion=750; reward=pure_accuracy_math; KL target 0.07, init_KL 3.0 |
| Math (7B Qwen) | OpenR1–Math–220k | LR 5×10^{-6} ; bs/dev=2; grad_acc=32; epochs=3; num_gens=4; max_prompt=450; max_completion=750; reward=pure_accuracy_math |
| Math (8B Llama 3.1) | OpenR1–Math–220k | LR 5×10^{-6} ; bs/dev=2; grad_acc=8; epochs=3; num_gens=4; max_prompt=450; max_completion=750; reward=pure_accuracy_math; PPO clip 0.10 |
| Xword (1.5B Qwen) | Guardian–Cryptonite (official split) | LR 1×10^{-5} ; bs/dev=4; grad_acc=256; epochs=3; num_gens=8; return_reason=true; max_reason=275; max_completion=320; reward=pure_accuracy (0/1 + shaping) |
| RHour (1.5B Qwen) | Rush 4/5/6–balanced | LR 5×10^{-6} ; bs/dev=4; grad_acc=64; epochs=3; num_gens=4; return_reason=true; max_prompt=3000; max_completion=300; reward=rush_solution_shaped |

Table 9: Per-domain GRPO run settings. Values shown are the *run-time* choices from the YAML configs; optimizer, KL control, horizons, and logging match the overview text.

failure, prior-stability, and conditional-gain criteria (App. §B.1). We then describe the trace-level shift annotation protocol used throughout the paper (App. §B.2). Then, we document the LLM-as-a-judge reliability protocol and the human-labeling template used for validation (App. §B.3 and App. §B.4). Finally, we describe our triggered-reconsideration intervention (the fixed “Wait ...” cue used to demarcate Pass 2 pivots) and explain how it avoids circularity with shift detection (App. §B.5).

B.1 Algorithm: Detecting an “Aha!” Moment Overview. Alg. 1 operationalizes Def. 3.1 and Fig. 2 via three checks:

- (i) **Prior failures:** for q_j , all checkpoints $i < k$ remain below a correctness ceiling.
- (ii) **Prior stability:** mid-trace shifts are rare for

$i < k$.

- (iii) **Conditional gain at k :** when a mid-trace shift occurs at k , expected correctness increases by a prescribed margin.

Estimating expected correctness. For each pair (q_j, k) we draw M independent traces $\tau^{(m)} \sim \pi_{\theta_k}(\cdot | q_j)$ under a fixed decoding policy (temperature τ , top- p , and stop conditions held constant across k):

$$\hat{P}_{\theta_k}(\checkmark | q_j) = \frac{1}{M} \sum_{m=1}^M R(\tau^{(m)}), \quad R(\tau) \in \{0, 1\}.$$

For the conditional estimate we average only *shifted* traces at k :

$$\hat{P}_{\theta_k}(\checkmark | q_j, S_{q_j,k}=1) = \frac{\sum_{m=1}^M R(\tau^{(m)}) \mathbb{1}[S(\tau^{(m)})=1]}{\sum_{m=1}^M \mathbb{1}[S(\tau^{(m)})=1] + \epsilon},$$

with a tiny ϵ (e.g., 10^{-6}) to avoid division by zero. If the denominator is 0, Step 3 is inconclusive and

the procedure returns **false**.

Detecting mid-trace shifts ($S(\tau)=1$). We mark a generation as *shifted* if it contains *both*: (i) a lexical cue of reconsideration (e.g., “wait”, “recheck”, “let’s try”, “this fails because . . .”), and (ii) a *material* revision of the preceding plan (rejects/corrects an earlier hypothesis, switches method or candidate, or resolves a contradiction). We implement this with a conservative two-stage detector (lexical cue prefilter + rubric-guided adjudication) described in App. §B.2. To calibrate superficial hedges and edge cases, we tuned thresholds for the cue matcher and clamping on a small, human-verified set (App. §B.4).

Prior stability (Step 2). For each $i < k$,

$$\widehat{\Pr}[S_{q_j,i}=1] = \frac{1}{M} \sum_{m=1}^M \mathbb{1}[S(\tau_i^{(m)})=1],$$

and we require $\widehat{\Pr}[S_{q_j,i}=1] < \delta_2$ for all $i < k$.

Thresholds and statistical test (Step 3). We set $(\delta_1, \delta_2, \delta_3)$ on a held-out development slab by maximizing F1 for AHA vs. non-AHA against human labels. Unless stated otherwise, we use $\delta_1=0.125$ (prior correctness ceiling), $\delta_2=0.125$ (shift-rate ceiling), and $\delta_3=0.125$ (minimum gain). To guard against Monte Carlo noise in \hat{P} , we further require the one-sided bootstrap CI (2000 resamples over traces) for $\hat{P}_{\theta_k}(\checkmark | q_j, S=1) - \hat{P}_{\theta_k}(\checkmark | q_j)$ to exceed 0 at level $\alpha=0.05$. If this test fails, Step 3 returns **false**.

Decoding protocol. Unless noted otherwise, we use $M=8$ samples per (q_j, k) , top- $p=0.95$, temperature $\tau=0.7$, and truncate at the first full solution (math), full entry parse (xword), or solved board state (RHour). We cache RNG seeds so cross-checkpoint comparisons differ only by θ_k .

Complexity and caching. The detector uses $O(JKM)$ forward passes (plus inexpensive pre-filtering and adjudication), where J is the number of items and K the number of checkpoints. We cache $\{\tau^{(m)}, R(\tau^{(m)}), S(\tau^{(m)})\}$ per (q_j, k) for reuse in ablations (temperature sweeps, entropy bins).

Edge cases and fallbacks. (i) If any $i < k$ violates prior failure ($\hat{P}_{\theta_i} \geq \delta_1$), return **false**. (ii) If no shifted traces occur at k , Step 3 is inconclusive (**false**). (iii) Extremely small M inflates variance;

Algorithm 1: Detecting an “Aha!” Moment for question q_j at checkpoint k

Input: Checkpoints $\{f_{\theta_i}\}_{i=0}^K$, question q_j , thresholds $\delta_1, \delta_2, \delta_3$.
Output: Boolean flag Aha[j,k].
Step 1: Prior failures
for $i \leftarrow 0$ **to** $k-1$ **do**
 $p_i \leftarrow P_{\theta_i}(\text{correct} | q_j)$
 if $p_i \geq \delta_1$ **then**
 return false // Prior success breaks failure condition
Step 2: Prior stability
for $i \leftarrow 0$ **to** $k-1$ **do**
 $s_i \leftarrow \Pr[S_{q_j,i}=1]$
 if $s_i \geq \delta_2$ **then**
 return false // Too many prior shifts
Step 3: Performance gain
 $p_k \leftarrow P_{\theta_k}(\text{correct} | q_j)$
 $p_k^{\text{shift}} \leftarrow P_{\theta_k}(\text{correct} | q_j, S_{q_j,k}=1)$
if $p_k^{\text{shift}} - p_k > \delta_3$ **then**
 return true
else
 return false // No significant gain

we mark detections as “provisional” if the bootstrap half-width of either probability exceeds 0.08 and exclude them from aggregates.

Diagnostics. For each (q_j, k) we log: the prior-failure margin $\delta_1 - \max_{i < k} \hat{P}_{\theta_i}$, the stability margin $\delta_2 - \max_{i < k} \widehat{\Pr}[S=1]$, the gain $\hat{\Delta} = \hat{P}_{\theta_k}(\checkmark | S=1) - \hat{P}_{\theta_k}(\checkmark)$ with its CI, and short excerpts around the first cue marker for audits (App. §B.2).

Limitations. Our shift detector may miss unlexicalized representational changes (false negatives) and can be triggered by surface hedges if the adjudicator fails (false positives). The bootstrap addresses variance within checkpoints but not dataset shift across checkpoints; we therefore hold decoding hyperparameters fixed across k .

B.2 Detecting Reasoning Shifts in Traces

We flag a binary *shift in reasoning* inside the <think> block. A trace is labeled **TRUE** only if it exhibits both: (A) an explicit lexical cue of reconsideration, and (B) a *material revision* of the preceding plan (rejects/corrects an earlier hypothesis, switches method or candidate, or resolves a contradiction). Otherwise the label is **FALSE**.

Annotation pipeline. Given checkpointed JSONL outputs, we annotate each trace in four steps:

Algorithm 2: Detecting a reasoning shift in a single trace τ

Input: Trace τ ; cue whitelist \mathcal{W} ; judge \mathcal{J} with strict JSON schema.
Output: $S(\tau) \in \{\text{true}, \text{false}\}$.
Extract $t \leftarrow \tau.\langle \text{think} \rangle$ (clamp to 4,096 characters);
 $c \leftarrow \text{PREFILTERCUES}(t; \mathcal{W})$;
if c is empty **then**
 \perp **return false**
 $v \leftarrow \mathcal{J}(t, c)$; // rubric-guided verdict in JSON
if v is invalid JSON **then**
 \perp **return false**
if $v.\text{shift_in_reasoning} = \text{true}$ **then**
 \perp **return true**
else
 \perp **return false**

1. **Parse.** Extract $\langle \text{think} \rangle$ and $\langle \text{answer} \rangle$ with a robust regex; clamp $\langle \text{think} \rangle$ to 4,096 characters.
2. **Cue prefilter (A).** Search $\langle \text{think} \rangle$ for any cue from a whitelist (Table 10). If none is present, assign **FALSE**.
3. **Material revision check (B).** For prefilter hits, query an LLM judge (GPT-4o) with a rubric that restates (A)+(B) and requests a strict JSON verdict plus short before/after excerpts around the first cue. If the verdict is uncertain or invalid, assign **FALSE**.
4. **Record.** Write the Boolean label and minimal diagnostics (markers, first-cue offset, excerpts) back to the record; processing order is randomized with a fixed seed. The procedure is idempotent—existing labels are left unchanged. This conservative policy (requiring both an explicit cue and a substantiated revision, and defaulting to **FALSE** on uncertainty) keeps false positives low and yields conservative prevalence estimates.

Whitelist (lexical cues). To bias the LLM-as-a-judge toward *explicit* reconsideration, we prefilter traces using a hand-crafted list of lexical cues. Concretely, we match case-insensitive regex patterns over the $\langle \text{think} \rangle$ text (Table 10), covering common morphology and light paraphrase (e.g., “wait”, “hold on”, “scratch that”, “I was wrong”, “misread”, “re-check”, etc.). Cues are grouped semantically in the implementation (e.g., `src/annotate/core/prefilter.py`). A positive shift label is only accepted when at least one explicit cue is present—either from the prefilter or from cues the judge itself extracts.

Blacklist (negatives & exclusions). We reject as insufficient evidence: (i) bare discourse markers without correction (*but, however, therefore, also*);

(ii) hedges or meta-verbosity (*maybe, perhaps, I think, let’s be careful*) without an explicit pivot; (iii) formatting or notational fixes only; (iv) device/method names listed without rejecting a prior attempt; and (v) cues appearing *outside* $\langle \text{think} \rangle$. The judge prompt enforces these, and our implementation forces **FALSE** when no explicit cue is present.

Material-revision test (B). The judge must justify that the post-cue span negates or corrects a prior claim, selects a different candidate, changes the solving device/method, or resolves a contradiction. We store short before/after excerpts around the first cue to aid audits, and we only accept a **TRUE** label when the judge’s JSON is parseable and consistent with the excerpts. Otherwise we default to **FALSE**.

Error handling, privacy, and rate limits. If the judge call fails or returns invalid JSON, we save the prompt to a local log file, stamp **FALSE**, and continue. We clamp long $\langle \text{think} \rangle$ segments before sending to the judge. Optional jitter (default ≤ 0.25 s) randomizes inter-call delays.

Reproducibility. We fix a shuffle seed for candidate order, sort files by natural stepNNN and path, and perform atomic rewrites. The detector is content-idempotent: re-running will skip annotated lines and only fill missing fields.

Limitations. The whitelist privileges *explicit* cues and may miss unlexicalized pivots (false negatives). Conversely, some cues can appear in non-revisional discourse; the material-revision test mitigates but does not eliminate such false positives. Because we default to **FALSE** on uncertainty, prevalence estimates are conservative.

B.3 LLM-as-a-Judge Protocol and Reliability

Bias mitigation. We use GPT-4o as a scalable surrogate for shift annotation and address known judge biases—position, length, and model-identity—with a three-part protocol (Wang et al., 2024; Shi et al., 2024; Li et al., 2024b):

1. **Order randomization.** We randomly permute items and (when applicable) apply split-merge aggregation to neutralize position effects (Shi et al., 2024).
2. **Rubric-anchored scoring.** GPT-4o completes a structured JSON rubric, following G-Eval-style guidance (Liu et al., 2023).

LLM-as-a-Judge — System Prompt (Shift in Reasoning)

You are a careful annotator of single-pass reasoning transcripts.
Your task is to judge whether the writer makes a CLEAR, EXPLICIT "shift in reasoning" within `\tagtt{think}...\tagtt{think}`.

A TRUE label requires BOTH:
(A) an explicit cue (e.g., "wait", "hold on", "scratch that", "contradiction"),
AND (B) a material revision of the earlier idea (reject/correct an initial hypothesis, pick a new candidate, fix a contradiction, or change device/method).

Do NOT mark TRUE for rhetorical transitions, hedging, or generic connectives without an actual correction. Judge ONLY the content inside `\tagtt{think}`.
Be conservative; these events are rare.

Figure 9: **LLM-as-a-Judge (system prompt)**. One instruction template used to adjudicate whether a `<think>` trace contains a bona fide reasoning shift (explicit cue *and* material revision).

LLM-as-a-Judge — User Template (filled per example)

Problem/Clue (if available):
{problem}

PASS-1 `\tagtt{think}` (truncated if long):
{think}

Heuristic cue candidates (may be empty): {cues}
first_marker_pos: {pos}

Return ONLY a compact JSON object with keys:

- shift_in_reasoning: true|false
- confidence: "low"|"medium"|"high"
- markers_found: string[] (verbatim lexical cues you relied on)
- first_marker_index: integer (character offset into `\tagtt{think}`, -1 if absent)
- before_excerpt: string (<=120 chars ending right before the first marker)
- after_excerpt: string (<=140 chars starting at the first marker)
- explanation_short: string (<=140 chars justification)

Figure 10: **LLM-as-a-Judge (user template)**. Per-example payload including the clamped `<think>` text, problem/clue, and whitelist-prefiltered cue markers/position, with a strict JSON schema for the verdict.

3. Prompt-variant stability. We re-query with $K=5$ judge-prompt variants at judge temperature 0 and report inter-prompt agreement.

Table 11 lists the five judge prompt variants; Table 12 summarizes inter-prompt agreement on a fixed evaluation set.

Logged annotations. For each trajectory we record: (i) graded correctness, (ii) shift/no-shift label, (iii) whether a shift improved correctness, (iv) GPT-4o’s confidence (low/med/high), and (v) auxiliary statistics (e.g., entropy). This separation supports analyses of shift *prevalence* versus shift *efficacy*.

Prefiltering. Before judging, we apply a cue-based prefilter (App. §B.2). Empirically, responses

without cue words almost never contain qualifying shifts (human annotation of 100 such responses from Qwen-1.5B on MATH-500 found none).

Reliability (inter-prompt agreement). We evaluated Qwen-1.5B (GRPO) on MATH-500, using five paraphrased judge prompts (v1–v5), randomized item order, and judge temperature 0. Table 12 reports percent agreement (PO), mean pairwise Cohen’s κ , and 95% bootstrap CIs.

Human validation. Relative to a human majority-vote reference on 20 examples, GPT-4o achieved Cohen’s $\kappa = 0.794$ with PO= 0.900. Mean human–human agreement was lower (PO= 0.703, mean pairwise $\kappa = 0.42$), and mean LLM–human agreement was intermediate

Table 10: **Cue list** (lexical indicators for reconsideration). Each row denotes a family of regex triggers; variants and minor orthographic differences are included.

| Category | Representative cues (lemmas/phrases) |
|--------------------------------------|---|
| Pauses & self-interruptions | <i>wait, hold on/up, hang on, one/just a second, give me a moment, pause, on second/further thought, reconsider, rethink</i> |
| Explicit pivots/corrections | <i>actually, in fact, rather, instead (of), let's fix/correct that, correction:, to correct, change/switch to, replace with, try/consider instead, alternate/alternative, new candidate/answer/approach</i> |
| Immediate reversals | <i>no, that/this/it ..., never mind/nvm, disregard/ignore that, scratch/strike/forget that, I retract/take it back, I stand corrected, not X but/rather Y</i> |
| Error admissions | <i>I was wrong / that's wrong / incorrect, (my) mistake, my bad, oops/whoops, apologies, erroneous</i> |
| "Mis-*" failures | <i>misread, miscount, miscalculate / calculation error, misapply, misparse, misspell, misindex, misuse, conflated, typo, off-by-one</i> |
| Constraint/length mismatches (xword) | <i>doesn't fit/match (length/pattern), letters don't fit, pattern/length mismatch, too many/few letters, wrong length, violates enumeration, doesn't parse, definition mismatch, not an anagram of, fodder mismatch</i> |
| Contradictions/impossibility | <i>contradiction, inconsistent, can't/cannot be, impossible, doesn't make sense / add up, cannot both, leads to a contradiction</i> |
| Re-check / backtrack | <i>recheck / double-check / check again, re-evaluate / re-examine / upon review/reflection, backtrack, start over/restart/reset/from scratch</i> |
| "Prev X, but ..." templates | <i>I (initially/originally) thought ... but/however, previously ... but/however, earlier ... but/however, however ... correct/fix/instead/rather/change</i> |
| Omission/oversight | <i>I forgot/missed/overlooked/ignored, didn't notice, misremembered/misheard</i> |
| Directional swaps | <i>reversed / backwards, swapped, mixed up</i> |
| Realization formulas | <i>turns out, I (now) realize, on reflection, after all</i> |
| Failure templates | <i>fails because, won't work / not working, dead end</i> |

| Variant | System prompt summary |
|---------|--|
| v1 | Baseline strict judge: explicit cue (e.g., "wait", "hold on", "scratch that", "contradiction") AND a material revision; ignore hedging; judge only the <think> span. |
| v2 | Audit <think> for change of course: cue + substantive revision required; ignore rhetorical connectives; conservative. |
| v3 | "Corrects themselves mid-thought": needs an explicit reconsideration cue and a replacement/fix of prior approach; ignore small edits/hedges. |
| v4 | "Quality control": cue + meaningful course change; minor tweaks/hedging are not shifts; judge only the <think> span. |
| v5 | "Spot explicit change of mind": cue + real update (reject/swap/repair); true shifts are rare. |

Table 11: Judge prompt variants v1–v5 used for shift-in-reasoning annotation.

(PO= 0.758, mean pairwise $\kappa = 0.51$). Table 13 summarizes these comparisons.

Reproducibility. We include the full rubric and sample items from our human annotation survey in App. §B.4.

B.4 Human Annotators Template

Annotator pool & consent. We used 6 *volunteer* adult annotators (unpaid), recruited from the authors' academic networks. Participants gave informed consent on the task page and could with-

draw at any time. No sensitive personal information was requested.

IRB status. This activity consisted solely of judgments about model-generated text and did not involve collection of sensitive data or interventions with human participants. Under our institutional guidelines, it does not constitute human-subjects research; consequently, no IRB review was sought.

Presentation & blinding. Items were shown in randomized order. Annotators saw the original *Question asked* and the verbatim <think> trace (with tags preserved; traces clamped to 4096 characters). Model family, size, checkpoint, temperature, and correctness signals were withheld.

Labels & rubric. Primary label: **Yes/No** (shift present). Optional fields: confidence (low/med/high), first cue index (character offset), and a one-sentence rationale. Edge cases defaulted to **No** unless a method switch (e.g., completing-the-square → factoring; permutations → stars-and-bars; prime factorization → Euclidean algorithm) was evident.

Calibration & quality. Annotators completed a short calibration set (including Examples A–H) with immediate feedback. During labeling we interleaved hidden gold items and monitored time-on-

| Epoch | Judged N | Mean PO | Mean κ [95% CI] |
|---------------|------------|---------|------------------------|
| 0 (pre) | 500 | 0.983 | 0.655 [0.507, 0.775] |
| 1 (ckpt 500) | 500 | 0.986 | 0.759 [0.606, 0.863] |
| 2 (ckpt 1000) | 500 | 0.988 | 0.770 [0.631, 0.868] |
| 3 (final) | 500 | 0.988 | 0.719 [0.526, 0.848] |

Table 12: Inter-prompt agreement on binary reasoning-shift labels for MATH-500 (Qwen-1.5B). Settings: $K=5$ judge-prompt variants, model decoding temperature 0.7 (for the generated traces), judge temperature 0, cue prefilter window 350 characters, randomized item order. PO = percent agreement.

| Comparison | N | PO | Cohen’s κ |
|--------------------------------|-----|-------|------------------|
| GPT-4o vs. human majority vote | 20 | 0.900 | 0.794 |
| Mean human–human (pairwise) | 20 | 0.703 | 0.42 |
| Mean LLM–human (pairwise) | 20 | 0.758 | 0.51 |

Table 13: **Human validation of shift labels.** We compare GPT-4o shift judgments against a human majority-vote reference on a 20-item validation set (PO = percent agreement). We also report mean pairwise Cohen’s κ among human annotators and between GPT-4o and individual humans on the same items.

item; submissions failing pre-registered thresholds were flagged for review.

Agreement & adjudication. Each item received independent labels. We report Cohen’s κ with 95% bootstrap CIs.

Data handling. We did not collect sensitive demographics. Released artifacts include prompts, anonymized traces (with <think> clamps), labels, and aggregation scripts; any operational contact data (if present) were excluded from the release.

Task. Read the model’s <think> trace for a math problem and answer: “Does this <think> trace include a change in thinking?” Choices: **Yes / No.**

When to mark Yes. (1) The model clearly switches strategies mid-trace. (2) It abandons one method after noticing a contradiction, dead end, or mistake, and adopts a different method. (3) This is a real strategy pivot, not a small fix.

When to mark No. (1) The model keeps using the same method throughout. (2) It only makes minor arithmetic/algebra fixes. (3) It adds detail or notation without changing approach. Important: cue words alone (“wait”, “recheck”, etc.) do not count; look for an actual method switch.

Quick checklist. Identify the initial method. Look for a pivot: does the model drop that plan and adopt a different method? Ignore small fixes. Answer **Yes** only with a clear pivot; otherwise **No.**

Worked Examples (Gold-Labeled)

Example A — YES *Question.* How many sides would there be in a convex polygon if the sum of all but one of its interior angles is 1070° ?

Model <think> trace

```
\tagtt{think}. . . computes with a wrong
  ↪ assumption, gets  $(360=90)$  (
  ↪ contradiction), then re-evaluates and
  ↪ sets up
 $\backslash(\theta=(n-2)\cdot 180^\circ-1070^\circ)$ 
  ↪ and solves under  $(0^\circ<\theta<$ 
  ↪  $180^\circ)$  . . . \tagtt{think}
```

Correct answer: Yes. Why: Notices a contradiction and switches approach.

Example B — NO *Question.* Simplify $3/\sqrt{27}$.

Model <think> trace

```
\tagtt{think}. . .  $\backslash(\sqrt{27}=3\sqrt{3})$ 
  ↪  $\backslash(\backslash\to\backslash)$   $\backslash(3/(3\sqrt{3}))=1/\sqrt$ 
  ↪  $\{3\}\backslash(\backslash\to\backslash)$  rationalize  $\backslash(\to\backslash)$ 
  ↪  $\backslash(\sqrt{3}/3)$  . . . \tagtt{think}
```

Correct answer: No. Why: One method throughout (simplify radical → rationalize).

Example C — YES *Question.* Solve $x^2 - 5x - 14 = 0$.

Model <think> trace

```
\tagtt{think}. . . tries completing the square
  ↪ , finds it awkward, then switches to
  ↪ factoring  $\backslash((x-7)(x+2))$  . . . \tagtt{
  ↪ think}
```

Correct answer: Yes. Why: Switch from completing the square to factoring.

Example D — NO *Question.* Compute

$$\frac{d}{dx} \left(\frac{x^2 + 3x + 2}{x + 1} \right).$$

Model <think> trace

```
\tagtt{think}... uses the quotient rule;
  ↪ minor sign fix; simplify ... \etagtt{
  ↪ think}
```

Correct answer: No. Why: Stays with quotient rule; no pivot.

Example E — YES Question. How many non-negative integer solutions (x, y) satisfy $x + y = 20$?

Model <think> trace

```
\tagtt{think}... first frames as
  ↪ permutations, then switches to stars
  ↪ and bars: \(\binom{21}{1}=21\) ... \
  ↪ etagtt{think}
```

Correct answer: Yes. Why: Pivot from permutations framing to stars-and-bars.

Example F — NO Question. What is the sum of the first 50 odd integers?

Model <think> trace

```
\tagtt{think}... recalls \((n^2)\) \(\to\)
  ↪ \((50^2)=2500\) ... \etagtt{think}
```

Correct answer: No. Why: Single recalled formula; no change in method.

Example G — YES Question. Find $\gcd(252, 198)$.

Model <think> trace

```
\tagtt{think}... starts prime factoring,
  ↪ then switches to the Euclidean
  ↪ algorithm ... \etagtt{think}
```

Correct answer: Yes. Why: Abandons prime factorization for Euclidean algorithm.

Example H — NO Question. Solve the system $x + y = 10$, $xy = 21$.

Model <think> trace

```
\tagtt{think}... uses Vieta's formulas \(\to\)
  ↪ \) factor \(\to\) \(\{7,3\}\) ... \
  ↪ etagtt{think}
```

Correct answer: No. Why: One consistent quadratic/Vieta approach.

Annotation Questions, Verbatim. Then, our annotators are asked to provide Yes/No answers

as to whether a shift in reasoning occurred for the following 20 questions. We have included one sample question here, to provide an example of the task:

Question 1: Does this <think> trace include a change in thinking? (Yes/No)

Question Asked: If a and b are positive integers such that $\gcd(a, b) = 210$, $\text{lcm}[a, b] = 210^3$, and $a < b$, how many possible values are there for a ?

Model Response: <think>The greatest common divisor (GCD) of a and b is given as 210, and their least common multiple (LCM) is 210^3 . This implies that both a and b share the prime factors of 210, which are 2, 3, and 5.

Let's express a and b in terms of their prime factorizations:

$$a = 2^x \cdot 3^y \cdot 5^z$$

$$b = 2^w \cdot 3^v \cdot 5^t$$

Given that $\gcd(a, b) = 210 = 2^1 \cdot 3^1 \cdot 5^1$, we get the following equations from comparing exponents:

For the GCD:

$$x = 1, \quad y = 1, \quad z = 1$$

For the LCM:

$$w = x, \quad v = y, \quad t = z$$

Thus,

$$w = 1, \quad v = 1, \quad t = 1$$

Since $\text{lcm}[a, b] = 210^3$, substituting the values gives us:

$$w + v + t = 3$$

This simplifies our expressions for w , v , and t to:

$$1 + 1 + 1 = 3$$

Therefore, the only combination satisfying these conditions is:

$$w = 1, \quad v = 1, \quad t = 1$$

Substituting back into the expression for a yields:

$$a = 2^1 \cdot 3^1 \cdot 5^1 = 210$$

However, since $a < b$, this alone does not provide any new information because $a = b$.

To find all pairs (a, b) where $a < b$ and $\gcd(a, b) = 210$: - We need to consider all combinations of a and b such that:

$$2^x \cdot 3^y \cdot 5^z < 2^{x'} \cdot 3^{y'} \cdot 5^{z'}$$

Considering $x > x'$, $y > y'$, and $z > z'$:

- For $x > x'$: Since x cannot be greater than 1 (as a must be less than b), the only possibility is $x' = 0$ (i.e., a can take on any power of 2 except 0, making $a = 210k$ where k is some integer).

- Similarly, considering $y > y'$ and $z > z'$, the same logic applies for powers of 3 and 5 respectively.

Thus, a could be $210 \cdot 3^m \cdot 5^n$ for various combinations of m and n .

Each choice of m and n uniquely determines a , resulting in infinitely many possible values of a because there are infinitely many integers m and n .

Fill-in-the-bubble response. Choices: Yes / No (no alternative / third choice was provided).

LLM-as-a-Judge assessed answer. "Yes".

Post Assessment Post-assessment, we reveal the LLM-as-a-Judge answer to participants and encourage them to invite others to participate. Each individual’s score was weighted equally, and we analyzed annotator agreement as described in B.3. The complete assessment is made available as part of our codebase.

B.5 Triggered Reconsideration Intervention

In §6.3, we study an *extrinsically triggered* reconsideration mechanism (“forced Aha!”): for each instance we first generate a baseline completion (Pass 1) and then re-query the model under the same decoding settings while appending a short reconsideration clause (Pass 2). The intervention is not intended to *define* a shift, but to create a consistent opportunity for deliberate re-evaluation.

Fixed pivot marker (deterministic demarcation).

For Pass 2 we prepend a fixed sentence that begins the reconsideration clause (e.g., “Wait, we need to reconsider. Let’s think this through step by step.”). This invariant prefix serves a purely operational role: it deterministically marks the intended pivot point across models, checkpoints, and temperatures, making the intervention location unambiguous in downstream analyses (e.g., excerpting or aligning text around the prompted reconsideration).

Avoiding circularity with shift detection. Because the intervention includes an explicit reconsideration cue, a lexical-only detector would trivially fire on Pass 2. We therefore emphasize that our canonical shift detector is *not* a cue detector: it requires both (i) an explicit cue *and* (ii) a *material revision* of the plan within the <think> trace (reject/correct a prior hypothesis, switch method/candidate, or resolve a contradiction with a substantively different approach; App. §B.2). Thus, the fixed “Wait . . .” prefix provides a standardized *timestamp* for reconsideration, while the detector separately verifies whether the content exhibits a genuine strategy pivot rather than superficial cueing.

Implementation details. In all Pass 2 runs we hold decoding parameters fixed relative to Pass 1 (temperature, top- p , stop criteria, and token budgets) and vary *only* the appended reconsideration text. We also reuse the same answer-format contract (<think>/<answer>) so that validators and scoring remain comparable across passes.

Cue robustness and variants. The main paper reports results using a single canonical reconsideration clause for Pass 2. To test whether gains depend on the specific lexical phrasing (rather than the act of prompting reconsideration), we repeat the intervention with multiple semantically similar but lexically distinct cues (C1–C3) and show that the qualitative conclusions persist; see App. §C.4.

C Additional Results and Robustness Checks

This appendix collects supplementary analyses that extend and stress-test the main results. We first report the prevalence of *formal* “Aha!” events under threshold grids and summarize cross-domain patterns (App. §C.1). We then replicate key regressions and uncertainty analyses on larger model families (Qwen2.5–7B and Llama 3.1–8B) to verify that the shift effects generalize beyond Qwen2.5–1.5B (App. §C.3). Next, we test whether entropy-gated, *extrinsically triggered* reconsideration is robust to the specific cue wording (App. §C.4). Finally, we evaluate external models (DeepSeek–R1 and GPT–4o) under the same shift-detection protocol and compare alternative shift detectors (App. §C.5 and App. §C.6). Together, these checks show that our qualitative conclusions are stable across domains, model families/sizes,

prompt variants, and detector choices.

C.1 “Aha!” Moment Prevalence

How to read the heatmaps. Each panel reports the share of problem–checkpoint pairs (q_j, k) that satisfy our operational definition of an “Aha!” moment (Def. 3.1) under a grid of thresholds: $\delta_1 \in \{0, \frac{1}{8}, \frac{2}{8}\}$ (maximum prior accuracy), $\delta_2 \in \{0, \frac{1}{8}, \frac{2}{8}\}$ (maximum prior shift rate), and, unless noted, $\delta_3 = \epsilon > 0$ (any non-zero gain at k). Cells show the percentage and the raw counts ($\#events/\#pairs$). We aggregate over checkpoints ≤ 1000 with $G=8$ samples per item, and we use the conservative detector described in App. B.2 (lexical cue *and* material revision; default to **FALSE** on uncertainty).

Cross-domain patterns. Three robust trends emerge across *Xword*, *Math*, and *RHour* and across model families/sizes.

- **Rarity.** Even under the lenient gain criterion $\delta_3 = \epsilon$, “Aha!” events occupy a very small fraction of problem–checkpoint pairs. Most cells are near zero; none approach a large fraction. This mirrors the main-text finding that mid-trace shifts seldom coincide with measurable improvements.
- **Sensitivity to prior instability and prior accuracy.** Relaxing either prerequisite increases counts but remains small in magnitude. In particular, moving to higher δ_2 (allowing more prior shifts, i.e., lower prior stability) and higher δ_1 (allowing occasional prior solves) produces the visually “warmest” cells—consistent with the intuition that “Aha!” detections concentrate where traces have shown some volatility and the item is not maximally hard.
- **Domain/model differences.** *RHour* exhibits a higher *raw* shift rate (App. §6.2), but the “Aha!” filter (requiring a gain at k) prunes most cases; the absolute prevalence remains low. *Xword* shows small pockets of higher prevalence when $\delta_1, \delta_2 \geq \frac{1}{8}$, whereas *Math* is uniformly sparse. Scaling from Qwen 1.5B to 7B or switching to Llama 3.1–8B does not materially increase prevalence.

Stricter gain thresholds. Replacing $\delta_3 = \epsilon$ with a minimal absolute lift (e.g., at least one of the $G=8$ samples flips from incorrect to correct at k) further reduces counts but preserves the qualitative ordering across domains and models.

Takeaway. Across all settings, formal “Aha!” moments—requiring *both* a mid-trace reasoning pivot and a contemporaneous performance gain—are *vanishingly uncommon*. The sparse, threshold-stable patterns in Figs. 13–14 show this finding across temperatures, domains, and models.

C.2 Formal Threshold Search

To make our threshold-selection procedure concrete, we ran the grid/bootstrapped threshold search across the stored Qwen2.5–1.5B evaluation outputs for each domain and temperature.

For each domain \times temperature root, we searched a small grid $\delta_1, \delta_2 \in \{0, 1/8, 2/8\}$ and $\delta_3 \in \{\text{None}, 0, 0.05, 0.125\}$ (with `min_prior_steps=2`), and selected the “best” configuration according to the script’s default ranking (maximize the bootstrap lower CI bound for the mean gain; ties broken by prevalence and mean gain). Table 14 reports the best row per root. We report mean gain as $100 \cdot \mathbb{E}[\hat{P}(\checkmark | S=1) - \hat{P}(\checkmark)]$ in percentage points (pp), with a 95% bootstrap CI over flagged pairs; entries are “–” when no events are found or when N is too small to form a stable CI.

Takeaway. Across these stored outputs, the selected thresholds yield extremely low event prevalence, and gains are generally small, unstable, or negative. In particular, for *Math* at $T \in \{0.05, 0.3\}$ the best available configurations (under this search) have *negative* bootstrap lower bounds, indicating no robust evidence that shifted traces outperform the baseline on the flagged pairs.

C.3 Qwen-7B and Llama-8B Regressions

We extend the main-text analysis to probe the role of model *family* and *size*. Replicating the raw-effect analyses for **Qwen2.5–7B** and **Llama 3.1–8B** on *MATH*, we observe the same qualitative pattern reported for **Qwen2.5–1.5B**: mid-trace reasoning shifts are consistently detrimental across training steps and remain negative across decoding temperatures (magnitudes vary, not the sign), matching Fig. 12 and Table 15.

We begin by checking whether the core RQ1 finding about *rarity* generalizes across model family and size. Using the same formal detector (Def. 3.1) and threshold grid as in the main text, we compute the fraction of problem–checkpoint pairs that qualify as “Aha!” events. Fig. 11 shows that these events remain extremely sparse for both

| Domain | T | δ_1 | δ_2 | δ_3 | events/pairs | prev (%) | gain (pp) | CI (pp) |
|--------|------|------------|------------|------------|--------------|----------|-----------|----------------|
| Math | 0 | 1/8 | 1/8 | ϵ | 8/16000 | 0.05 | +0.00 | [+0.00, +0.00] |
| Math | 0.05 | 2/8 | 2/8 | ϵ | 41/16000 | 0.26 | -2.74 | [-5.79, -0.30] |
| Math | 0.3 | 2/8 | 2/8 | ϵ | 43/10000 | 0.43 | -2.62 | [-4.65, -1.16] |
| Math | 0.7 | 2/8 | 2/8 | ϵ | 92/16000 | 0.57 | +1.22 | [-1.77, +4.76] |
| Xwords | 0 | 0 | 0 | ϵ | 0/3120 | 0.00 | - | - |
| Xwords | 0.05 | 1/8 | 1/8 | ϵ | 3/3120 | 0.10 | +0.00 | [+0.00, +0.00] |
| Xwords | 0.3 | 1/8 | 1/8 | ϵ | 7/3120 | 0.22 | +0.00 | [+0.00, +0.00] |
| Xwords | 0.7 | 2/8 | 2/8 | ϵ | 18/3120 | 0.58 | +0.00 | [+0.00, +0.00] |
| RHour | 0 | 1/8 | 1/8 | ϵ | 1/503 | 0.20 | +0.00 | - |
| RHour | 0.05 | 1/8 | 1/8 | ϵ | 1/498 | 0.20 | -0.07 | - |
| RHour | 0.3 | 1/8 | 1/8 | ϵ | 7/498 | 1.41 | -0.01 | [-0.01, +0.00] |
| RHour | 0.7 | 1/8 | 1/8 | ϵ | 18/513 | 3.51 | -0.01 | [-0.02, -0.00] |

Table 14: **Grid/bootstrapped threshold search on Qwen2.5–1.5B stored evaluation outputs (best row per root)**. Each row summarizes the top-ranked threshold setting for the corresponding domain \times temperature root when running with 500 bootstrap draws. “events/pairs” counts flagged formal pairs out of all (problem, step) pairs in that root, and “prev” is the corresponding percentage. “gain” is the mean gain at shifted traces (pp) over flagged pairs, and “CI” is the 95% bootstrap percentile interval.

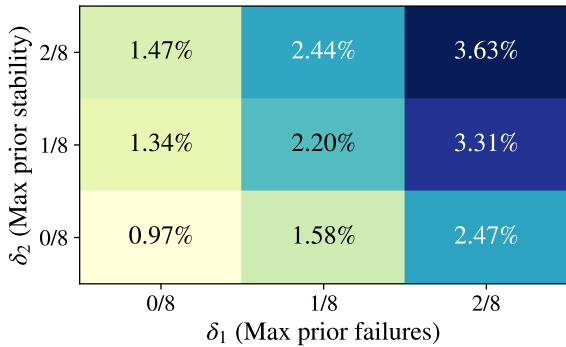


Figure 11: **Prevalence of formal “Aha!” events for Qwen–7B and Llama–8B (Math, $T=0.7$)**. Each cell shows the fraction (and count) of problem–checkpoint pairs (q_j, k) that satisfy Def. 3.1 under varying thresholds for prior failures (δ_1) and prior stability (δ_2), with $\delta_3 = \epsilon > 0$. Even under lenient settings, formal “Aha!” events are exceedingly rare. Per-temperature breakdowns appear in App. D.3.

Qwen2.5–7B and Llama 3.1–8B (MATH, $T=0.7$).

C.3.1 Step and Temperature Analysis

We then repeat the regression analysis from Table 3 for these models. Figure 12 visualizes the raw effect across training steps and decoding temperatures.

C.3.2 Uncertainty Analysis

This appendix extends the main-text uncertainty analysis to larger model families on MATH, using traces from **Qwen2.5–7B** and **Llama 3.1–8B**. Our goal is to test a simple hypothesis: if reasoning shifts are primarily an *uncertainty response*, then shifts should become *more likely* as uncertainty rises. We operationalize uncertainty using each

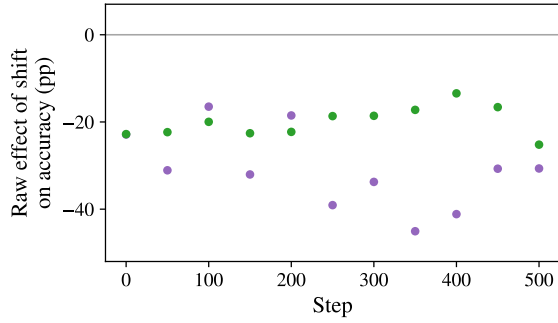
| (a) Training stage (fixed $T = 0.7$) | | | |
|---|-----------------------|-----------------------|------------------------|
| Metric | Qwen2.5–7B | Llama 3.1–8B | Combined |
| N | 40,000 | 40,000 | 40,000 |
| $\%S$ | 1.37 | 6.54 | 3.89 |
| $\hat{p}_{Y S=1}$ | 0.3467 | 0.2709 | 0.2846 |
| Δ (pp) | -30.39 | -17.68 | -26.97 |
| AME | -0.0841 | -0.0688 | -0.1706 |
| p | 4.38×10^{-4} | 6.7×10^{-11} | 5.93×10^{-42} |
| (b) Temperature (temps pooled, steps ≤ 450) | | | |
| Metric | Qwen2.5–7B | Llama 3.1–8B | Combined |
| N | 160,000 | 160,000 | 320,000 |
| $\%S$ | 1.50 | 5.04 | 3.26 |
| $\hat{p}_{Y S=1}$ | 0.2821 | 0.2816 | 0.2818 |
| Δ (pp) | -37.85 | -17.56 | -27.94 |
| AME | -0.0833 | -0.0529 | -0.1457 |
| p | 4.89×10^{-6} | 2.25×10^{-5} | 2.83×10^{-22} |

Table 15: **Effect of detected reasoning shifts on accuracy (Qwen2.5–7B/Llama 3.1–8B)**. $\%S$ is shift prevalence; $\hat{p}_{Y|S=1}$ is accuracy among shifted traces; and Δ is the raw accuracy gap (pp) relative to non-shifted traces. AME and p come from Binomial(logit) regressions with problem fixed effects and cluster-robust SEs (clustered by problem). Negative AMEs indicate that shifts reduce accuracy.

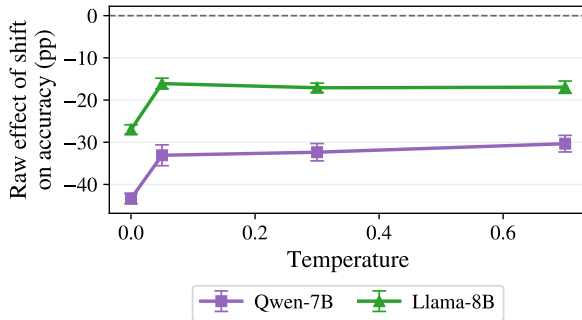
trace’s *sequence-level entropy* and use the same GPT-derived binary shift indicator as in the main text.

Shift prevalence vs. entropy. For each decoding temperature T , we regress the shift indicator on standardized sequence entropy with problem fixed effects and cluster-robust standard errors clustered by problem:

$$\text{shift} \sim C(\text{problem}) + \text{std_entropy}.$$



(a) Raw effect by training step at $T=0.7$ (MATH).



(b) Raw effect vs. decoding temperature (MATH).

Figure 12: **Qwen2.5-7B vs. Llama 3.1-8B on MATH.** Raw accuracy difference $\Delta = \hat{p}_{Y|S=1} - \hat{p}_{Y|S=0}$. (a) Across training steps at $T=0.7$, the effect is stable and negative for both models. (b) Across temperatures $T \in \{0.0, 0.05, 0.3, 0.7\}$ the effect remains negative; Llama 3.1-8B exhibits a smaller penalty than Qwen2.5-7B.

Across both model families, we again find a *non-positive* association between entropy and shift prevalence. In particular, at $T=0.05$ and $T=0.7$, a 1 SD increase in entropy significantly *reduces* the odds of a detected shift ($OR_{1\sigma}=0.63$, $p=0.001294$; $OR_{1\sigma}=0.67$, $p=0.002396$), while the estimates at $T=0$ and $T=0.3$ are not distinguishable from zero. This mirrors the smaller Qwen2.5-1.5B MATH models: shifts are not more common in high-entropy regimes, and when a dependence is detectable, it points in the opposite direction.

Entropy-stratified shift effects on accuracy. To complement the prevalence analysis, Table 16 stratifies the *raw* shift effect on correctness by entropy (high = top 20%, low = bottom 80%), pooling temperatures and restricting to early training steps (steps ≤ 450). The qualitative picture is consistent across strata: shifts are associated with *lower* accuracy even within the high-entropy slice.

Forced reconsideration as a separate mechanism. Finally, Table 17 reports paired sample-

| Metric | Qwen2.5-7B | Llama 3.1-8B | Combined |
|---|-----------------------|--------------|----------|
| All traces (temps pooled, steps ≤ 450) | | | |
| N | 160,000 | 160,000 | 320,000 |
| Δ (pp) | -44.43 | -14.83 | -33.69 |
| p | 1.32×10^{-4} | 0.6973 | 0.001725 |
| High entropy (top 20%) | | | |
| N | 32,000 | 31,757 | 63,763 |
| Δ (pp) | -22.03 | -8.93 | -10.30 |
| p | 0.06963 | 0.7834 | 0.001017 |
| Low entropy (bottom 80%) | | | |
| N | 128,000 | 127,027 | 255,021 |
| Δ (pp) | -48.87 | -14.23 | -38.86 |
| p | 1.44×10^{-4} | 0.7221 | 0.01824 |

Table 16: Entropy-stratified shift effects (MATH, steps ≤ 450 , temps pooled). Δ (pp) is the raw accuracy gap $\hat{p}(\checkmark | S=1) - \hat{p}(\checkmark | S=0)$. p is from $\text{logit}(\text{correct} \sim \text{shift} + \text{problem FEs})$ within each stratum.

| Metric | Qwen2.5-7B | Llama 3.1-8B |
|-----------------------|------------|--------------|
| N | 14,176 | 222,658 |
| \hat{p}_{P1} | 0.5509 | 0.4416 |
| \hat{p}_{P2} | 0.6107 | 0.3997 |
| Δ (pp) | +5.97 | -4.19 |
| wins (P2 \uparrow) | 2,156 | 27,106 |
| wins (P1 \uparrow) | 1,309 | 36,439 |

Table 17: **Forced “Aha!” (triggered reconsideration), sample-level results on MATH.** \hat{p}_{P1} and \hat{p}_{P2} are accuracies in baseline vs. forced pass; Δ is the percentage-point gain; “wins” count paired samples where one pass is correct and the other is not.

level results for *triggered reconsideration* (Pass 2). This manipulation differs from spontaneous shifts: it explicitly prompts the model to re-evaluate. On MATH, forced reconsideration yields a positive gain for **Qwen2.5-7B** (+5.97pp) but a negative gain for **Llama 3.1-8B** (-4.19pp) in this evaluation slice. We tested only on a subset given the high compute cost.

C.4 Entropy-Gated Interventions with Multiple Cues

To test whether the effect of artificially triggered reflection depends on the specific reconsideration cue used, we evaluate three semantically similar but lexically distinct prompts:

- **C1:** “Hold on, this reasoning might be wrong. Let’s go back and check each step carefully.”
- **C2:** “Actually, this approach doesn’t look correct. Let’s restart and work through the solution more systematically.”

| Cue | β (std. ent.) | OR | OR _{1σ} |
|-----|----------------------|----------------------|------------------------------------|
| C1 | 0.79 [0.59, 1.00] | 3.64 [2.60, 5.09] | 2.21 [1.80, 2.72] |
| C2 | 0.86 [0.65, 1.07] | 4.32 [3.03, 6.17] | 2.36 [1.92, 2.91] |
| C3 | 0.91 [0.71, 1.12] | 4.09 [2.98, 5.62] | 2.49 [2.03, 3.06] |

Table 18: **Entropy-gated improvement under three reconsideration cues.** β is the coefficient on standardized entropy from a logistic regression controlling for baseline correctness and problem fixed effects; brackets give 95% CIs. OR is the unit odds ratio (raw entropy), and OR_{1 σ} is the odds ratio for a 1 SD increase in entropy.

- **C3:** “Wait, something is not right; we need to reconsider. Let’s think this through step by step.”

For each cue, we re-run 8×500 Math problems (Qwen2.5–1.5B, final checkpoint) with 1-shot decoding at $T=0.1$, obtaining 500 paired baseline and cued completions per cue. We then fit a logistic regression for each cue, controlling for baseline correctness and problem identity.⁷

Across all cues, higher entropy is strongly associated with improved post-intervention accuracy. Table 18 reports standardized entropy coefficients, unit odds ratios (raw entropy), and odds ratios for a 1 SD increase in entropy.

All three cues show the same qualitative pattern: a one-standard deviation increase in entropy substantially increases the odds of correctness after the reconsideration cue ($2.2\times$ – $2.5\times$ across cues). C2 yields the strongest effect, but the differences are modest, indicating that the intervention’s success is tied to *uncertainty* rather than to any particular lexical phrasing.

C.5 Reasoning Shifts at Scale

To verify that our findings are not an artifact of the GRPO-tuned models studied in the main paper, we evaluate two widely discussed reasoning models—DeepSeek–R1 and GPT–4o—under our shift-detection protocol. These models have been cited as exhibiting frequent “Aha!” moments or dramatic mid-trace realizations (Guo et al., 2025), making them a natural stress test for our methodology.

⁷In R-style notation: $\text{correct} \sim \text{entropy_std} + \text{baseline_correct} + C(\text{problem})$. Here entropy_std is the within-domain standardized sequence-level entropy defined in §5.3.

Experimental setup. We evaluate both models with:

- group sampling with $G=8$ independent traces per instance (fixed decoding settings per group),
- decoding temps $T \in \{0, 0.05, 0.3, 0.7, 1\}$,
- identical prompting format (with `<think>` and `<answer>` tags),
- no system-level alterations or heuristics.

On MATH–500 and RHOOR, each model produces $G=8$ samples for each of 500 instances, yielding $N=4,000$ traces per model per temperature. On XWORDS, we evaluate 130 instances under the same $G=8$ protocol, yielding $N=1,040$ traces per model per temperature (reflecting completed pass-1 outputs).

Shift detection. We use the same annotation protocol as in §5.2 and App. B.2:

1. **Cue prefilter:** at least one explicit lexical cue of reconsideration (e.g., “wait”, “actually”, “hold on”), using the whitelist in Table 10.
2. **Material revision:** GPT–4o judges whether the post-cue reasoning constitutes a genuine plan pivot (rejecting a candidate, switching method, resolving a contradiction), returning a strict JSON verdict.
3. Cases lacking either (A) lexical cue or (B) structural revision are labeled as **no shift**.

Results. Table 19 reports reasoning shift prevalence and conditional accuracy by decoding temperature across three domains. Under our strict definition (lexical cue + material plan revision), shifts are *non-trivial but still minority* events across both external models. On MATH–500, DeepSeek–R1 shifts occur in roughly 4.70–5.17% of traces, while GPT–4o ranges from 1.45–4.40% depending on T . On XWORDS, canonical shift rates are lower for DeepSeek–R1 (0.96–1.83%) and moderate for GPT–4o (2.98–7.31%). On RHOOR, GPT–4o exhibits 1.45–3.12% shift rates across temperatures; DeepSeek–R1 RHOOR results are not yet available.

Crucially, shifts do not yield a consistent accuracy benefit: for most model–domain–temperature settings, $P(\checkmark \mid S=1)$ is below $P(\checkmark \mid S=0)$. Even in the closest-to-parity case (GPT–4o on MATH–500 at $T=0.3$), shifted traces remain slightly less accurate ($P(\checkmark \mid S=1)=0.632$ vs. $P(\checkmark \mid S=0)=0.649$), and the direction of the gap

is not stable across temperatures. Overall, we do not observe a temperature regime in which shifts reliably improve correctness.

Interpretation. These results reinforce two conclusions:

1. **Reasoning shifts remain a minority behavior even in heavily discussed “reasoning” models.** Across three domains and a five-temperature sweep, only a small fraction of traces satisfy our strict definition (lexical cue + material plan revision), with rates ranging from roughly $\sim 1\%$ on XWORDS (DeepSeek-R1) to a few percent on MATH-500 and RHOURL.
2. **Reasoning shifts do not reliably improve accuracy.** Conditional accuracy under a shift is typically *lower* than the non-shift baseline across domains and temperatures, and we do not observe a stable regime in which shifts consistently improve correctness.

Data release. We release the external-model outputs and shift annotations used in this analysis on Hugging Face, mirroring the per-temperature splits used throughout this appendix (Table 20). DeepSeek-R1 RHOURL runs are not yet included.

C.6 Alternate Shift Detectors

Prior work shows that superficial linguistic markers of hesitation—such as “wait,” “hold on,” or “actually”—are unreliable indicators of genuine cognitive shifts. Keyword-based detectors misclassify such cues at high rates, often interpreting hedges or verbosity as insight-like events (Zheng et al., 2023b; Xia et al., 2025). Recent analyses of “Aha!”-style behavior in LLMs similarly report that many mid-trace cues reflect shallow self-correction or filler language rather than substantive plan changes (Yang et al., 2025).

In parallel, LLM-as-a-judge evaluations are known to exhibit position, ordering, and verbosity biases unless structured and controlled (Wang et al., 2024; Shi et al., 2024; Li et al., 2024b). Because our primary shift detector uses an LLM-as-judge, it is important to verify that conclusions do not depend on the specific annotation mechanism.

Detector variants. We replicate the full RQ1 analysis using three detectors: (i) a strict formal “Aha!” criterion (Def. 3.1), (ii) our rubric-guided GPT-based shift detector used in the main text, and (iii) a permissive lexical-only detector that flags

any cue-phrase occurrence. Table 21 summarizes results for Qwen2.5-1.5B at $T = 0.7$.

1. **Formal Aha (formal).** The strict criterion in Def. 3.1, which requires (i) prior failure, (ii) prior stability, and (iii) a performance gain on traces with a detected shift.
2. **GPT-based shifts (gpt).** GPT-4o marks a shift when it observes an explicit cue of reconsideration together with a material change in reasoning strategy (App. B.2).
3. **Lexical-only shifts (words).** A looser detector that flags a shift whenever the <think> trace contains at least one cue phrase from our whitelist, regardless of whether the subsequent reasoning reflects a genuine plan pivot.

Metrics. For each detector, domain, and item we compute: (i) shift prevalence $\%S$, (ii) accuracies $\hat{p}_{Y|S=1}$ and $\hat{p}_{Y|S=0}$, (iii) the raw accuracy difference $\Delta\% = 100 \cdot (\hat{p}_{Y|S=1} - \hat{p}_{Y|S=0})$ (percentage points), and (iv) the average marginal effect (AME) of a shift from a logistic regression with problem fixed effects and cluster-robust SEs (shown with p -value).

Takeaways. Two patterns are consistent across domains:

1. **Shifts are rare under every detector.** Even the most permissive lexical detector (words) identifies shifts in at most 1.2% of Math traces and 0.6% of RHour traces; the formal Aha criterion is stricter still.
2. **Shifts are non-beneficial to accuracy.** Raw differences $\Delta\%$ and AMEs are non-positive across domains and detectors, with the only exception being MATH under the strict formal detector, where the estimate is small and statistically indistinguishable from zero ($p=0.82$). In MATH, both the GPT-based and lexical detectors show statistically significant negative AMEs.

Overall, this robustness check confirms that our main RQ1 conclusion does not depend on the specific shift detector: whether we use the strict formal Aha definition, the rubric-guided GPT detector, or a lexical cue heuristic, mid-trace shifts are rare and generally *harm* correctness rather than help it.

D Supplementary Figures & Tables

Overview. This appendix collects supplementary tables and figures that expand the main-text analyses and document additional aggregations that are

| Domain | Model | T | # Problems | G | # Traces | % Shifts (count) | $P(\checkmark S=0)$ | $P(\checkmark S=1)$ |
|----------|-------------|------|------------|-----|----------|------------------|-----------------------|-----------------------|
| MATH-500 | | | | | | | | |
| | DeepSeek-R1 | 0 | 500 | 8 | 4,000 | 4.70% (188) | 0.559 | 0.245 |
| | DeepSeek-R1 | 0.05 | 500 | 8 | 4,000 | 5.17% (207) | 0.550 | 0.169 |
| | DeepSeek-R1 | 0.3 | 500 | 8 | 4,000 | 5.10% (204) | 0.534 | 0.211 |
| | DeepSeek-R1 | 0.7 | 500 | 8 | 4,000 | 4.85% (194) | 0.529 | 0.196 |
| | DeepSeek-R1 | 1 | 500 | 8 | 4,000 | 5.10% (204) | 0.532 | 0.172 |
| | GPT-4o | 0 | 500 | 8 | 4,000 | 4.40% (176) | 0.656 | 0.267 |
| | GPT-4o | 0.05 | 500 | 8 | 4,000 | 2.02% (81) | 0.688 | 0.210 |
| | GPT-4o | 0.3 | 500 | 8 | 4,000 | 2.17% (87) | 0.649 | 0.632 |
| | GPT-4o | 0.7 | 500 | 8 | 4,000 | 1.45% (58) | 0.639 | 0.293 |
| | GPT-4o | 1 | 500 | 8 | 4,000 | 3.45% (138) | 0.643 | 0.304 |
| XWORDS | | | | | | | | |
| | DeepSeek-R1 | 0 | 130 | 8 | 1,040 | 1.54% (16) | 0.677 | 0.688 |
| | DeepSeek-R1 | 0.05 | 130 | 8 | 1,040 | 1.63% (17) | 0.703 | 0.235 |
| | DeepSeek-R1 | 0.3 | 130 | 8 | 1,040 | 0.96% (10) | 0.799 | 0.400 |
| | DeepSeek-R1 | 0.7 | 130 | 8 | 1,040 | 1.06% (11) | 0.797 | 0.545 |
| | DeepSeek-R1 | 1 | 130 | 8 | 1,040 | 1.83% (19) | 0.793 | 0.526 |
| | GPT-4o | 0 | 130 | 8 | 1,040 | 6.63% (69) | 0.733 | 0.377 |
| | GPT-4o | 0.05 | 130 | 8 | 1,040 | 6.35% (66) | 0.716 | 0.439 |
| | GPT-4o | 0.3 | 130 | 8 | 1,040 | 6.44% (67) | 0.725 | 0.373 |
| | GPT-4o | 0.7 | 130 | 8 | 1,040 | 7.31% (76) | 0.698 | 0.421 |
| | GPT-4o | 1 | 130 | 8 | 1,040 | 2.98% (31) | 0.650 | 0.581 |
| RHOOR | | | | | | | | |
| | GPT-4o | 0 | 500 | 8 | 4,000 | 3.12% (125) | 0.004 | 0.000 |
| | GPT-4o | 0.05 | 500 | 8 | 4,000 | 1.45% (58) | 0.005 | 0.000 |
| | GPT-4o | 0.3 | 500 | 8 | 4,000 | 2.17% (87) | 0.008 | 0.000 |
| | GPT-4o | 0.7 | 500 | 8 | 4,000 | 2.40% (96) | 0.007 | 0.000 |
| | GPT-4o | 1 | 500 | 8 | 4,000 | 2.73% (109) | 0.005 | 0.000 |
| | DeepSeek-R1 | 0 | 500 | 8 | 4,000 | 11.45% (458) | 0.000 | 0.000 |
| | DeepSeek-R1 | 0.05 | 500 | 8 | 4,000 | 11.25% (450) | 0.000 | 0.000 |
| | DeepSeek-R1 | 0.3 | 500 | 8 | 4,000 | 10.53% (421) | 0.000 | 0.000 |
| | DeepSeek-R1 | 0.7 | 500 | 8 | 4,000 | 10.95% (438) | 0.000 | 0.000 |
| | DeepSeek-R1 | 1 | 500 | 8 | 4,000 | 10.82% (433) | 0.000 | 0.000 |

Table 19: **External models across domains.** Canonical shift rates and conditional accuracy by decoding temperature on MATH-500, XWORDS, and RHOOR. Each instance is sampled $G=8$ times per temperature; MATH-500 and RHOOR use 500 instances ($N=4,000$ traces per model per T), while XWORDS uses 130 instances ($N=1,040$ traces per model per T), matching completed pass-1 outputs. We report shift prevalence (% Shifts), accuracy on non-shifted traces $P(\checkmark | S=0)$, and accuracy on shifted traces $P(\checkmark | S=1)$.

referenced in our scripts but not surfaced elsewhere in the paper. We provide: (i) training-stage regressions at fixed decoding temperatures (beyond the $T=0.7$ slice in the main text), (ii) temperature sweeps for the stricter *formal* “Aha!” detector, (iii) analogous temperature/stage breakdowns for larger models (Qwen2.5-7B and Llama 3.1-8B) on MATH, and (iv) additional uncertainty-gated intervention summaries, including pooled Qwen-1.5B and 7B/8B entropy-regression results. All tables use the same conventions as the main text: % S is shift prevalence, Δ_{pp} denotes a raw accuracy difference in percentage points, and AMEs/coefficients come from Binomial(logit) models with problem fixed effects and cluster-robust SEs (clustered by problem).

D.1 Training-stage effects at other decoding temperatures

Table 22 replicates the training-stage analysis from Table 3, holding the decoding temperature fixed at $T \in \{0.0, 0.05, 0.3\}$. Across these settings, we again find no evidence that reasoning shifts become beneficial later in training. In *Math*, shifts are consistently harmful across all temperatures. In *RHour*, accuracies are near zero for both shifted and non-shifted traces, and the estimated effects are practically negligible.

D.2 Training-stage effects at other decoding temperatures (Qwen-7B and Llama-8B)

Table 23 provides the same fixed-temperature, training-stage analysis as Table 22, but for larger

| Domain | Model | Dataset (Hugging Face) |
|----------|-------------|--------------------------|
| MATH-500 | GPT-4o | gpt4o-math500-t0 |
| MATH-500 | GPT-4o | gpt4o-math500-t005 |
| MATH-500 | GPT-4o | gpt4o-math500-t03 |
| MATH-500 | GPT-4o | gpt4o-math500-t07 |
| MATH-500 | GPT-4o | gpt4o-math500-t1 |
| MATH-500 | DeepSeek-R1 | deepseek-r1-math500-t0 |
| MATH-500 | DeepSeek-R1 | deepseek-r1-math500-t005 |
| MATH-500 | DeepSeek-R1 | deepseek-r1-math500-t03 |
| MATH-500 | DeepSeek-R1 | deepseek-r1-math500-t07 |
| MATH-500 | DeepSeek-R1 | deepseek-r1-math500-t1 |
| XWORDS | GPT-4o | gpt4o-xwords-t0 |
| XWORDS | GPT-4o | gpt4o-xwords-t005 |
| XWORDS | GPT-4o | gpt4o-xwords-t03 |
| XWORDS | GPT-4o | gpt4o-xwords-t07 |
| XWORDS | GPT-4o | gpt4o-xwords-t1 |
| XWORDS | DeepSeek-R1 | deepseek-r1-xwords-t0 |
| XWORDS | DeepSeek-R1 | deepseek-r1-xwords-t005 |
| XWORDS | DeepSeek-R1 | deepseek-r1-xwords-t03 |
| XWORDS | DeepSeek-R1 | deepseek-r1-xwords-t07 |
| XWORDS | DeepSeek-R1 | deepseek-r1-xwords-t1 |
| RHOUR | GPT-4o | gpt4o-rhour-t0 |
| RHOUR | GPT-4o | gpt4o-rhour-t005 |
| RHOUR | GPT-4o | gpt4o-rhour-t03 |
| RHOUR | GPT-4o | gpt4o-rhour-t07 |
| RHOUR | GPT-4o | gpt4o-rhour-t1 |
| RHOUR | DeepSeek-R1 | deepseek-r1-rhour-t0 |
| RHOUR | DeepSeek-R1 | deepseek-r1-rhour-t005 |
| RHOUR | DeepSeek-R1 | deepseek-r1-rhour-t03 |
| RHOUR | DeepSeek-R1 | deepseek-r1-rhour-t07 |
| RHOUR | DeepSeek-R1 | deepseek-r1-rhour-t1 |

Table 20: **Released external-model outputs across domains.** Hugging Face datasets containing the traces used in App. §C.5, for $T \in \{0, 0.05, 0.3, 0.7, 1\}$. All dataset IDs are under the namespace od2961/.

models on MATH (Qwen2.5-7B and Llama 3.1-8B), evaluated over steps ≤ 450 . Across temperatures, shifts remain associated with lower accuracy; the magnitude of the raw penalty varies with T and model family, but does not reverse sign.

D.3 Formal “Aha!” moments across decoding temperatures

We repeat the temperature-sweep analysis using the stricter *formal* “Aha!” detector (Def. 3.1), which requires a mid-trace pivot *and* a contemporaneous performance gain at that checkpoint. For each decoding temperature $T \in \{0, 0.05, 0.3, 0.7\}$, we estimate the association between correctness and the formal-Aha indicator while controlling for problem fixed effects and training stage (standardized step), reporting average marginal effects (AME) with cluster-robust SEs. Because the formal detector is extremely sparse in several regimes (and never fires for RHour at $T \leq 0.3$), some conditional quantities are undefined; we denote these

with “-”.

D.4 Formal “Aha!” moments across decoding temperatures (Qwen-7B and Llama-8B)

Table 24 repeats the formal-detector temperature sweep for larger models on MATH (Qwen2.5-7B and Llama 3.1-8B), evaluated over steps ≤ 450 . As in the 1.5B setting, formal “Aha!” detections remain extremely sparse across temperatures, and conditional estimates can be unstable.

D.5 Additional Temperature Ablations

Fig. 13 provides additional temperature ablations across our suite of Qwen2.5-1.5B traces for the Xword, Math, and RHour datasets. We carry out the same analysis over our Qwen-7B and Llama-8B Math traces in Fig. 14.

D.6 Qualitative review of formal “Aha!” Moments

Below, we show a qualitative inspection of a small set of (Formal) “Aha!” detections from our stored Qwen2.5-1.5B evaluation outputs. For each domain we apply the Formal criteria at the problem-checkpoint level and then show representative shifted traces.

Math. We use ($\delta_1 = 0.250$, $\delta_2 = 0.250$, $\delta_3 = 0.000$) with `min_prior_steps=2`.

Xword. We use ($\delta_1 = 0.500$, $\delta_2 = 0.500$, $\delta_3 = 0.000$) with `min_prior_steps=2`.

RHour. We use ($\delta_1 = 0.250$, $\delta_2 = 0.250$, $\delta_3 = None$) with `min_prior_steps=2`. Because RHour accuracies are near zero in these stored outputs, we found too few events satisfying a positive gain constraint; we therefore omit the gain threshold for this qualitative inspection.

D.7 Triggered reconsideration under uncertainty

We extend §6.3 by analyzing when an *extrinsically triggered* reconsideration cue (Pass 2) is most effective. We report both a nonparametric entropy gate (top-20% vs. bottom-80% by pass-1 entropy) and a regression that treats entropy as a continuous predictor.

Entropy-gated gains (nonparametric stratification). For each domain, we bucket prompts by pass-1 sequence entropy using a fixed within-domain threshold at the 80th percentile (high = top 20%, low = bottom 80%). We report pass-1

| Domain | Detector | %S | $\hat{p}_{Y S=1}$ | $\hat{p}_{Y S=0}$ | $\Delta\%$ | AME (p) |
|--------------|----------|--------|-------------------|-------------------|------------|-----------------------------------|
| Xword | formal | 0.0008 | 0.0000 | 0.1181 | -11.81 | -0.1181 (0) |
| | gpt | 0.0010 | 0.0400 | 0.1181 | -7.81 | -0.0651 (0.05095) |
| | words | 0.0013 | 0.0312 | 0.1182 | -8.69 | -0.0712 (0.04761) |
| Math | formal | 0.0008 | 0.0215 | 0.3006 | -27.91 | +0.0275 (0.8201) |
| | gpt | 0.0030 | 0.1622 | 0.3008 | -13.87 | -0.1086 (7.80×10^{-6}) |
| | words | 0.0120 | 0.2606 | 0.3009 | -4.03 | -0.0469 (0.002153) |
| RHour | formal | 0.0023 | 0.0000 | 0.0001 | -0.01 | -0.0001 (0) |
| | gpt | 0.0026 | 0.0000 | 0.0001 | -0.01 | -0.0001 (0) |
| | words | 0.0060 | 0.0000 | 0.0001 | -0.01 | -0.0001 (0) |

Table 21: **Alternative shift detectors (Qwen2.5–1.5B, $T=0.7$)**. Across all three detectors, shifts are rare and do not yield higher accuracy.

and pass-2 accuracies and the paired gain Δ in percentage points. In addition to per-domain results, we include a pooled “ALL” row that aggregates Xword/Math/RHour (count-weighted).

Entropy as a continuous predictor (regression).

We regress pass-2 correctness on standardized pass-1 entropy, controlling for pass-1 correctness and problem fixed effects (cluster-robust SEs at the problem level). Table 25 reports the log-odds coefficient β_{ent} (per +1 SD entropy) and the corresponding odds ratio $\text{OR}_{1\sigma} = \exp(\beta_{\text{ent}})$.

Pass-2 entropy regression for larger models.

Table 26 reports the same regression for Qwen2.5–7B and Llama 3.1–8B on MATH. Here, entropy has a small and non-significant association for Qwen2.5–7B, while for Llama 3.1–8B the association is negative and statistically detectable.

D.8 Pass-2 accuracy conditional on detected shifts (additional summary)

Because our intervention defines a second pass (Pass 2), it is useful to verify that the negative association between *spontaneous* shifts and correctness is not an artifact of evaluating only the first-pass answer. Table 28 reports, for each setting, the Pass 2 accuracy among traces whose *Pass 1* reasoning was labeled as shifted vs. non-shifted, alongside the corresponding raw differences.

E Release and Artifacts

All artifact details (contents, structure, and reproduction steps) are described in the corresponding artifact appendix sections of this document. For convenience, we provide the single entry-point link here.

Repository. The full artifact bundle (evaluation pipeline, shift-detection code, configs, and sup-

porting documentation) can be found linked to our github repository.

Contact. For questions, bug reports, or replication issues, please use the GitHub issue tracker: <https://github.com/humans-and-machines/Illusion-of-Reasoning/issues>

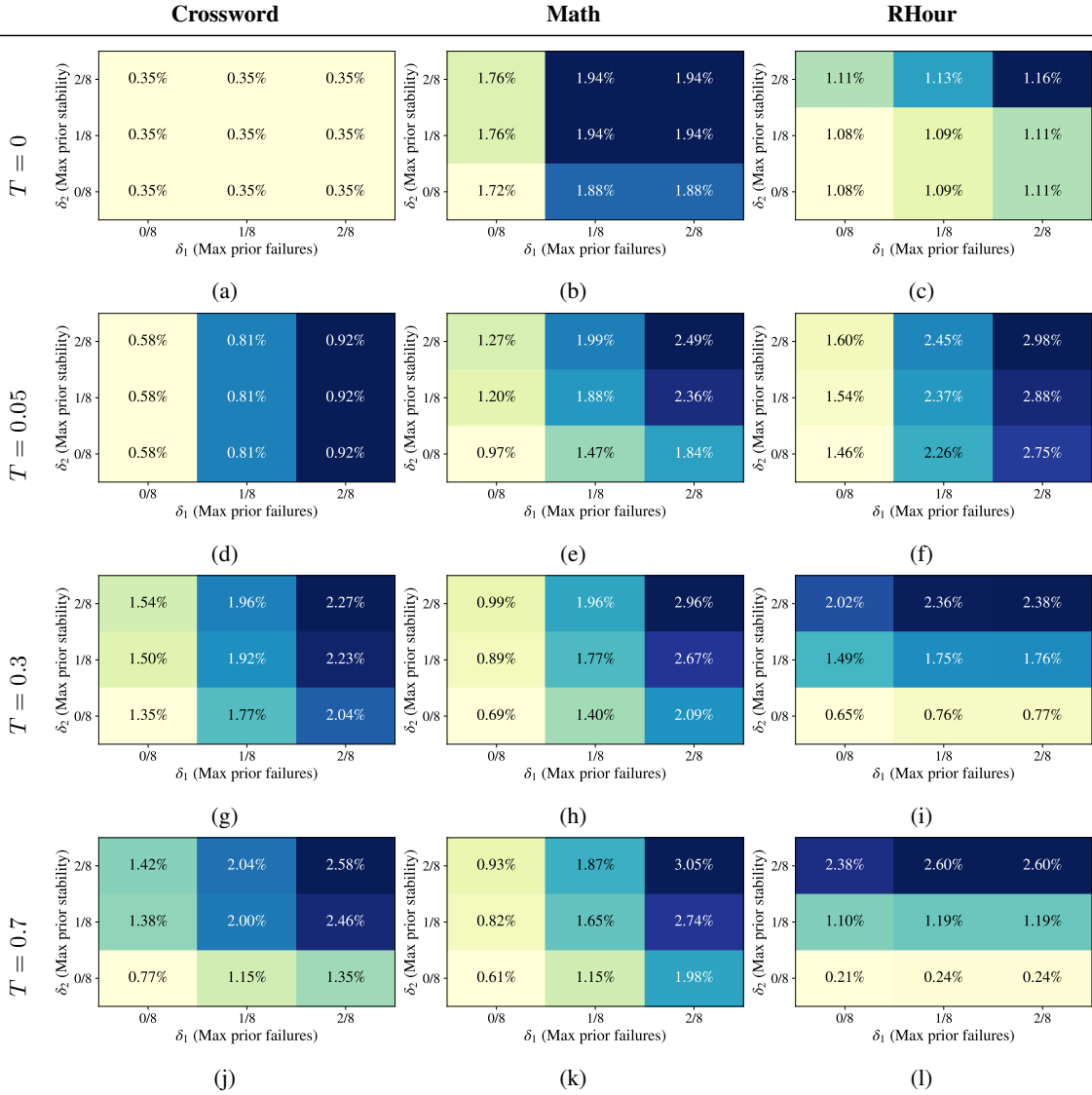


Figure 13: **Aha!** moment prevalence heatmaps (Qwen-1.5B) across decoding temperatures. Columns are domains; rows vary decoding temperature T . Cells show the share of (q_j, k) pairs meeting Def. 3.1 under the threshold grid; see App. B.2 for detection details.

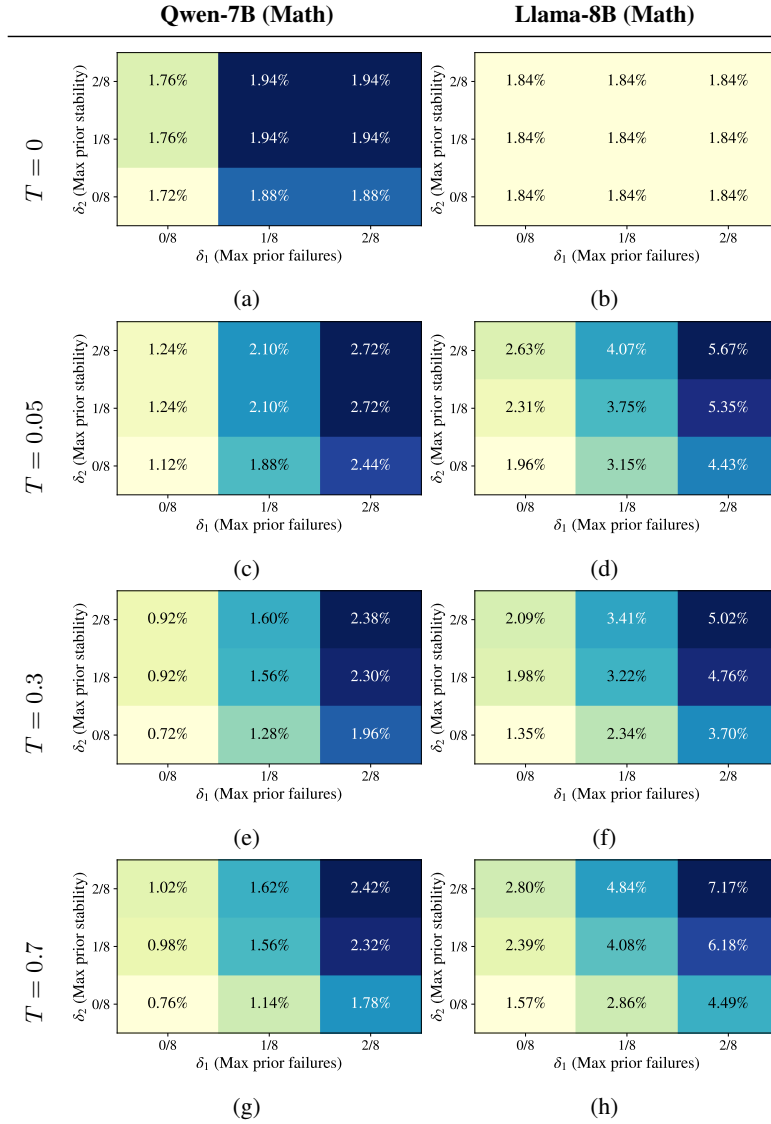


Figure 14: **Aha! moment prevalence heatmaps (Qwen-7B vs. Llama-8B; MATH-500) across decoding temperatures.** Columns are models; rows vary decoding temperature T . Cells show the share of (q_j, k) pairs meeting Def. 3.1 under the threshold grid; see App. B.2 for detection details.

| Training stage at fixed decoding temperature $T = 0.0$ | | | |
|---|------------------------|-------------------------|---------|
| Metric | Xword | Math | RHour |
| N | 20,800 | 80,000 | 80,000 |
| $\%S$ | 0.947 | 1.866 | 14.679 |
| $\hat{p}_{Y S=1}$ | 0.3655 | 0.0683 | 0.0000 |
| Δ_{pp} | +28.24 | -23.64 | -0.04 |
| AME | 0.0027 | -0.0044 | -0.0001 |
| p | 6.89×10^{-35} | 1.19×10^{-67} | 0.999 |
| Training stage at fixed decoding temperature $T = 0.05$ | | | |
| Metric | Xword | Math | RHour |
| N | 20,800 | 80,000 | 80,000 |
| $\%S$ | 0.851 | 1.854 | 15.386 |
| $\hat{p}_{Y S=1}$ | 0.3390 | 0.1382 | 0.0000 |
| Δ_{pp} | +25.41 | -18.56 | -0.05 |
| AME | 0.0022 | -0.0034 | -0.0001 |
| p | 1.94×10^{-26} | 2.0×10^{-47} | 0.999 |
| Training stage at fixed decoding temperature $T = 0.3$ | | | |
| Metric | Xword | Math | RHour |
| N | 20,800 | 80,000 | 80,000 |
| $\%S$ | 0.649 | 4.696 | 15.759 |
| $\hat{p}_{Y S=1}$ | 0.2593 | 0.1637 | 0.0000 |
| Δ_{pp} | +16.28 | -23.01 | -0.01 |
| AME | 0.0011 | -0.0108 | -0.0000 |
| p | 1.93×10^{-9} | 1.58×10^{-158} | 0.999 |

Table 22: **Effect of detected reasoning shifts on accuracy (Qwen2.5-1.5B): training-stage analysis at fixed temperature.** For each fixed decoding temperature $T \in \{0.0, 0.05, 0.3\}$, we report the share of traces with a detected shift ($\%S$), accuracy among shifted traces ($\hat{p}_{Y|S=1}$), the raw accuracy difference in percentage points (Δ_{pp}) between shifted and non-shifted traces, and the average marginal effect (AME) from a logistic regression with problem fixed effects, a standardized training-step control, and cluster-robust SEs (clustered by problem). Negative AME values indicate that shifted traces are less likely to be correct holding problem and training stage fixed.

| Training stage at fixed decoding temperature $T = 0$ | | |
|---|-----------------------|----------------------|
| Metric | Qwen2.5-7B | Llama3.1-8B |
| N | 40,000 | 40,000 |
| $\%S$ | 2.538 | 2.418 |
| $\hat{p}_{Y S=1}$ | 0.2039 | 0.1607 |
| Δ_{pp} | -45.10 | -27.18 |
| AME | -0.0659 | -0.0597 |
| p | 0.03314 | 0.04043 |
| Training stage at fixed decoding temperature $T = 0.05$ | | |
| Metric | Qwen2.5-7B | Llama3.1-8B |
| N | 40,000 | 40,208 |
| $\%S$ | 0.853 | 5.710 |
| $\hat{p}_{Y S=1}$ | 0.3284 | 0.3319 |
| Δ_{pp} | -34.06 | -14.82 |
| AME | -0.0401 | -0.0436 |
| p | 0.07879 | 0.007971 |
| Training stage at fixed decoding temperature $T = 0.3$ | | |
| Metric | Qwen2.5-7B | Llama3.1-8B |
| N | 40,000 | 40,192 |
| $\%S$ | 1.248 | 5.576 |
| $\hat{p}_{Y S=1}$ | 0.3387 | 0.2945 |
| Δ_{pp} | -32.91 | -17.44 |
| AME | -0.0788 | -0.0540 |
| p | 2.91×10^{-4} | 4.4×10^{-5} |

Table 23: **Effect of detected reasoning shifts on accuracy: training-stage analysis at fixed temperature.** For each fixed decoding temperature $T \in \{0.0, 0.05, 0.3\}$, we report shift prevalence ($\%S$), accuracy among shifted traces ($\hat{p}_{Y|S=1}$), the raw accuracy difference in percentage points (Δ_{pp}), and the average marginal effect (AME) from a logistic regression with problem fixed effects, a standardized training-step control, and cluster-robust SEs. Negative AME values indicate that shifted traces are less likely to be correct holding problem and training stage fixed.

| Metric | Qwen2.5-7B | Llama3.1-8B |
|-------------------|----------------------------|---------------------------|
| $T = 0$ | | |
| N | 40,000 | 40,000 |
| $\%S$ | 0.362 | 0.832 |
| $\hat{p}_{Y S=1}$ | 0.0621 | 0.0449 |
| Δ_{pp} | -58.35 | -38.42 |
| AME | +0.0541 | +0.0112 |
| p | 0.6627 | 0.815 |
| $T = 0.05$ | | |
| N | 40,000 | 40,000 |
| $\%S$ | 0.048 | 0.090 |
| $\hat{p}_{Y S=1}$ | 0.0000 | 0.0278 |
| Δ_{pp} | -66.64 | -44.42 |
| AME | -0.3735 | +0.1626 |
| p | 1.84×10^{-135441} | 0.2839 |
| $T = 0.3$ | | |
| N | 40,000 | 40,000 |
| $\%S$ | 0.045 | 0.109 |
| $\hat{p}_{Y S=1}$ | 0.0000 | 0.0000 |
| Δ_{pp} | -66.40 | -45.97 |
| AME | -0.4970 | -0.4114 |
| p | 8.64×10^{-65910} | 3.13×10^{-95446} |
| $T = 0.7$ | | |
| N | 40,000 | 40,000 |
| $\%S$ | 0.022 | 0.073 |
| $\hat{p}_{Y S=1}$ | 0.0000 | 0.0357 |
| Δ_{pp} | -64.66 | -40.08 |
| AME | -0.5572 | +0.1937 |
| p | 4.7×10^{-54940} | 0.1153 |

Table 24: **Formal “Aha!” detector (Def. 3.1): temperature sweep for Qwen2.5-7B/Llama3.1-8B.** For each decoding temperature, $\%S$ is the share of traces flagged by the formal detector; $\hat{p}_{Y|S=1}$ is empirical accuracy among flagged traces; and Δ_{pp} is the raw accuracy difference (percentage points) between flagged and non-flagged traces. AME is the average marginal effect from a logistic regression with problem fixed effects and a standardized training-step control.

| Domain | N | β_{ent} | $OR_{1\sigma}$ | p |
|--------|---------|---------------|----------------|-------------------------|
| Xword | 99,840 | -0.033 | 0.97 | 0.091 |
| Math | 464,000 | +0.019 | 1.02 | 0.146 |
| RHour | 331,120 | -0.407 | 0.67 | 2.36×10^{-119} |

Table 25: **Pass-2 accuracy vs. pass-1 entropy (Qwen2.5-1.5B).** We regress pass-2 correctness on standardized pass-1 entropy, controlling for pass-1 correctness and problem fixed effects. β_{ent} is the log-odds coefficient for a 1 SD entropy increase and $OR_{1\sigma} = \exp(\beta_{ent})$.

| Group | N | β_{ent} | $OR_{1\sigma}$ | p |
|-------------|---------|---------------|----------------|----------|
| Qwen2.5-7B | 63,404 | +0.012 | 1.01 | 0.7586 |
| Llama3.1-8B | 102,232 | -0.075 | 0.93 | 0.005146 |

Table 26: **Pass-2 accuracy vs. pass-1 entropy (Qwen2.5-7B/Llama3.1-8B).** We regress pass-2 correctness on standardized pass-1 entropy, controlling for pass-1 correctness and problem fixed effects (cluster-robust SEs).

| Metric | Crossword | Math | RHour |
|-------------------|-----------------------|------------------------|--------|
| $T = 0.0$ | | | |
| N | 20,800 | 80,000 | 80,000 |
| $\%S$ | 0.471 | 0.462 | 0.000 |
| $\hat{p}_{Y S=1}$ | 0.0816 | 0.0000 | - |
| Δ_{pp} | -0.42 | -30.17 | - |
| AME | -0.0000 | -0.0014 | - |
| p | 0.883 | 0.999 | - |
| $T = 0.05$ | | | |
| N | 20,800 | 80,000 | 80,000 |
| $\%S$ | 0.212 | 0.299 | 0.000 |
| $\hat{p}_{Y S=1}$ | 0.0000 | 0.0251 | - |
| Δ_{pp} | -8.72 | -29.62 | - |
| AME | -0.0002 | -0.0009 | - |
| p | 0.999 | 1.92×10^{-12} | - |
| $T = 0.3$ | | | |
| N | 20,800 | 80,000 | 80,000 |
| $\%S$ | 0.312 | 0.475 | 0.000 |
| $\hat{p}_{Y S=1}$ | 0.0000 | 0.0211 | - |
| Δ_{pp} | -9.78 | -36.36 | - |
| AME | -0.0003 | -0.0017 | - |
| p | 0.999 | 4.17×10^{-21} | - |
| $T = 0.7$ | | | |
| N | 20,800 | 80,000 | 80,000 |
| $\%S$ | 1.438 | 0.364 | 8.191 |
| $\hat{p}_{Y S=1}$ | 0.0067 | 0.0241 | 0.0002 |
| Δ_{pp} | -11.22 | -26.17 | +0.01 |
| AME | -0.0016 | -0.0010 | 0.0000 |
| p | 2.41×10^{-5} | 3.25×10^{-13} | 0.461 |

Table 27: **Formal “Aha!” detector (Def. 3.1): temperature sweep.** For each domain and decoding temperature, $\%S$ is the share of traces flagged by the formal detector; $\hat{p}_{Y|S=1}$ is empirical accuracy among flagged traces; and Δ_{pp} is the raw accuracy difference (percentage points) between flagged and non-flagged traces. AME is the average marginal effect of a formal-Aha flag from a logistic regression with problem fixed effects, a standardized training-step control, and cluster-robust SEs. Cells marked “-” indicate the detector never fired in that regime, making conditional quantities undefined.

| Experiment | T | N | $\%S$ | $P_2(\checkmark S=1)$ | $P_2(\checkmark S=0)$ | Δ_2 (pp) |
|----------------------------|------|---------|-------|-------------------------|-------------------------|-----------------|
| Qwen2.5-1.5B (all domains) | all | 723,200 | 7.65 | 3.95 | 20.42 | -16.47 |
| Qwen2.5-7B (Math) | 0.0 | 39,080 | 2.54 | 25.62 | 67.10 | -41.47 |
| Qwen2.5-7B (Math) | 0.05 | 2,768 | 0.85 | 13.64 | 64.20 | -50.57 |
| Qwen2.5-7B (Math) | 0.3 | 1,104 | 1.25 | 30.77 | 57.79 | -27.03 |
| Qwen2.5-7B (Math) | 0.7 | 20,180 | 1.37 | 45.07 | 64.73 | -19.66 |
| Llama 3.1-8B (Math) | 0.0 | 14,728 | 2.42 | 32.94 | 36.67 | -3.74 |
| Llama 3.1-8B (Math) | 0.05 | 28,808 | 5.71 | 24.67 | 37.20 | -12.53 |
| Llama 3.1-8B (Math) | 0.3 | 30,240 | 5.58 | 27.78 | 40.19 | -12.41 |
| Llama 3.1-8B (Math) | 0.7 | 28,376 | 6.54 | 28.01 | 42.36 | -14.35 |

Table 28: **Pass-2 accuracy conditional on Pass-1 shift labels.** $P_2(\checkmark | S=1)$ and $P_2(\checkmark | S=0)$ denote Pass 2 accuracies among traces whose Pass 1 reasoning was labeled as shifted vs. non-shifted, respectively, and Δ_2 is the raw percentage-point difference.

| Domain | Bucket | N | \hat{p}_{P1} (%) | \hat{p}_{P2} (%) | Δ (pp) |
|---------|--------|---------|--------------------|--------------------|---------------|
| Xword | all | 99,840 | 9.65 | 10.15 | +0.49 |
| Xword | high | 19,969 | 8.56 | 9.59 | +1.04 |
| Xword | low | 79,871 | 9.93 | 10.29 | +0.36 |
| Math | all | 464,000 | 32.70 | 40.43 | +7.74 |
| Math | high | 92,800 | 19.70 | 35.09 | +15.38 |
| Math | low | 371,200 | 35.94 | 41.77 | +5.82 |
| RHour | all | 331,120 | 0.023 | 0.036 | +0.013 |
| RHour | high | 66,224 | 0.027 | 0.023 | -0.005 |
| RHour | low | 264,896 | 0.022 | 0.039 | +0.017 |
| Overall | all | 894,960 | 18.04 | 22.11 | +4.07 |
| Overall | high | 178,993 | 11.18 | 19.27 | +8.09 |
| Overall | low | 715,967 | 19.75 | 22.82 | +3.07 |

Table 29: **Triggered reconsideration gains by pass-1 entropy.** We bucket instances by pass-1 sequence entropy within each domain (high = top 20%, low = bottom 80%). “Overall” aggregates across domains using count-weighted averages.

Math example 1 (T=0.05, step=100)**root:** GRPO-1.5B-math-temp-0.05**problem:** If $\arccos x + \arccos 2x + \arccos 3x = \pi$, then x satisfies a cubic polynomial of the form $ax^3 + bx^2 + cx + d = 0$, where a, b, c, d are integers and $a \neq 0$. Find the smallest possible value of $|a| + |b| + |c| + |d|$.**step:** 100 **n_samples:** 8 **shift_rate:** 0.125 **freq_correct:** 0.125 $p(\sqrt{\mid S=1})$: 1.000 **gain:** +87.50 pp**Shift cue:** contradiction **Strategy shift:** propose $x \rightarrow$ feasibility check \rightarrow backtrack / replan**Shift excerpt (PASS-1):** [SHIFT at "However, substituting $x = 0 \dots$ does not satisfy"]**markers:** contradiction

$\dots x = 0$. However, substituting $x = 0$ into the original equation does not satisfy it because it contradicts π . Therefore, let's consider other values of x that might work. We need to solve the system of equations formed

PASS-1 <answer>: $-1/27$ **Math example 2 (T=0.05, step=850)****root:** GRPO-1.5B-math-temp-0.05**problem:** Let a, b, c, d be positive real numbers such that $a + b + c + d = 1$. Find the minimum value of $\frac{1}{a} + \frac{1}{b} + \frac{4}{c} + \frac{16}{d}$.**step:** 850 **n_samples:** 8 **shift_rate:** 0.125 **freq_correct:** 0.125 $p(\sqrt{\mid S=1})$: 1.000 **gain:** +87.50 pp**Shift cue:** instead **Strategy shift:** dead-end/degenerate reasoning \rightarrow switch to AM-GM template**Shift excerpt (PASS-1):** [SHIFT at "Instead, let's consider \dots AM-GM"]**markers:** instead \dots we get:

$$\frac{1}{a} + \frac{1}{b} + \frac{4}{c} + \frac{16}{d} = \infty.$$

However, this approach does not yield a finite minimum. Instead, let's consider another strategy using the Arithmetic Mean-Geometric Mean Inequality (AM-GM): \dots

PASS-1 <answer>: 64**Math example 3 (T=0.05, step=400)****root:** GRPO-1.5B-math-temp-0.05**problem:** Let $x_1, x_2, x_3, y_1, y_2, y_3$ be real numbers such that

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 = 9,$$

$$(x_1 - x_3)^2 + (y_1 - y_3)^2 = 16,$$

$$(x_2 - x_3)^2 + (y_2 - y_3)^2 = 25.$$

Find $\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}^2$.**step:** 400 **n_samples:** 8 **shift_rate:** 0.125 **freq_correct:** 0.125 $p(\sqrt{\mid S=1})$: 1.000 **gain:** +87.50 pp**Shift cue:** instead **Strategy shift:** direct determinant manipulation \rightarrow switch to "consider determinant itself" (reframe)**Shift excerpt (PASS-1):** [SHIFT at "Instead, we will consider the determinant itself \dots "]**markers:** instead

\dots However, finding the determinant directly from these equations might not be straightforward. Instead, we will consider the determinant itself and see if we can simplify our approach. \dots

PASS-1 <answer>: 144**Math example 4 (T=0.7, step=100)****root:** GRPO-1.5B-math-temp-0.7**problem:** A hexagon is inscribed in a circle (diagram omitted). What is the measure of α , in degrees?**step:** 100 **n_samples:** 8 **shift_rate:** 0.125 **freq_correct:** 0.125 $p(\sqrt{\mid S=1})$: 1.000 **gain:** +87.50 pp**Shift cue:** however **Strategy shift:** algebra gives invalid angle \rightarrow constraint check (angle must be positive) \rightarrow sign correction**Shift excerpt (PASS-1):** [SHIFT at "However, since $\alpha \dots$ should be positive"]**markers:** however

\dots Simplifying, $\alpha + 110^\circ = 70^\circ$, leading to $\alpha = -40^\circ$. However, since α represents an angle within the circle, it should be positive. \dots

PASS-1 <answer>: 40

Math example 5 (T=0.7, step=900)

root: GRPO-1.5B-math-temp-0.7
problem: Compute $\sin^3 18^\circ + \sin^2 18^\circ$.
step: 900 n_samples: 8 shift_rate: 0.125
freq_correct: 0.125 $p(\checkmark | S=1)$: 1.000 gain: +87.50 pp
Shift cue: instead Strategy shift: failed analytic attempt → abandon/switch to numerical/other-method talk (search collapse)
Shift excerpt (PASS-1): [SHIFT at "Instead, let's try solving numerically ..."]
markers: instead
... Since $\sqrt{5} > 3$, $y = \frac{1}{2}$ does not satisfy the equation. Instead, let's try solving numerically or by another method. ...
PASS-1 <answer>: $-1/8$

Xwords example 1 (T=0.7, step=1000)

root: GRPO-1.5B-xword-temp-0.7
problem: Front of ship; bend (3)
step: 1000 n_samples: 8 shift_rate: 0.125
freq_correct: 0.125 $p(\checkmark | S=1)$: 1.000 gain: +87.50 pp
Shift cue: doesn't fit Strategy shift: definition mismatch → reinterpret "front of ship" → BOW
Shift excerpt (PASS-1): [SHIFT at "However, considering ... ship's front ..."]
markers: doesn't fit
... the definition does not fit directly. However, considering the context of a ship's front, ...
PASS-1 <answer>: BOW

Xwords example 2 (T=0.7, step=300)

root: GRPO-1.5B-xword-temp-0.7
problem: Clue: Celebrity from stare losing tail (4)\nGold: STAR
step: 300 n_samples: 8 shift_rate: 0.125
freq_correct: 0.125 $p_{\text{correct_given_shift}}$: 1.000 gain: +87.50 pp
Shift excerpt (PASS-1): [SHIFT: doesn't fit]
Strategy shift: length mismatch → simplify transformation
markers: doesn't fit
... Attempt with "TILL": Definition: "Celebrity" could be fitting given the context, but the length is still too much. This seems too long and doesn't fit easily into the 4-letter constraint. Wordplay: STARE loses its TAIL → STAR. PASS-1 <answer>: STAR

Xwords example 3 (T=0.3, step=550)

root: GRPO-1.5B-xword-temp-0.3
problem: Clue: Shoe liner at home on fish (6)\nGold: INSOLE
step: 550 n_samples: 8 shift_rate: 0.250
freq_correct: 0.250 $p_{\text{correct_given_shift}}$: 0.500 gain: +25.00 pp
Shift excerpt (PASS-1): [SHIFT: doesn't fit / re-evaluate] Strategy shift: reject homophone → switch to charade
markers: doesn't fit, re-evaluate
... Homophone? ("shoe" sounds like "soe") - Doesn't fit here
Candidate attempt (WRONG PATH): - HIDDEN: No hidden indicator present - WRONG PATH: Consider other indicators
Re-evaluate as CHARADES: IN ("at home") + SOLE ("fish") → INSOLE. Accounting: INSOLE letters: I N S O L E (6). Definition "Shoe liner" fits. Enumeration (6) OK.
Final answer: INSOLE
PASS-1 <answer>: INSOLE

Xwords example 4 (T=0.05, step=900)

root: GRPO-1.5B-xword-temp-0.05
problem: Clue: Frighten when energy enters scar (5)\nGold: SCARE
step: 900 n_samples: 8 shift_rate: 0.250
freq_correct: 0.125 $p_{\text{correct_given_shift}}$: 0.500 gain: +37.50 pp
Shift excerpt (PASS-1): [SHIFT: rethink / re-evaluate] Strategy shift: failed parse → substitution re-parse
markers: rethink, re-evaluate
... Device matches, but there's no indication that it should be considered an abbreviation. Candidate attempt (wrong path): - fails definition, so we must rethink. Re-evaluate as SUBSTITUTION: SCARE → SCARE. Accounting: SCARE letters: S C A R E (5). Definition "Frighten" fits. Enumeration (5) OK.
PASS-1 <answer>: SCARE

RHour example 1 (T=0, step=950)

root: GRPO-1.5B-carpark-temp-0
problem: example_id: idx_101\nGold: ['Cv2', 'A>1']
step: 950 n_samples: 8 shift_rate: 1.000
freq_correct: 0.000 $p(\checkmark | S=1)$: 0.000 gain: +0.00 pp
Shift cue: hang on Strategy shift: attempt → detect blocker → replan with enabling move
Shift excerpt (PASS-1): [SHIFT at "Hang on..."]
markers: hang on
... S1: E>1 → GOAL× (blocked by D). Hang on ...
S2: Fv1,E>1 → GOAL✓ (all valid sequences pass this check).
PASS-1 <answer>: Fv1,E>1

RHour example 2 (T=0, step=600)

```
root: GRPO-1.5B-carpark-temp-0
problem: example_id: idx_109\nGold: ['Cv1',
'Cv1', 'A>2']
step: 600 n_samples: 8 shift_rate: 0.375
freq_correct: 0.000  $p(\sqrt{|S|=1})$ : 0.000 gain:
+0.00 pp
Shift cue: check again Strategy shift: local
fix → explicit verification pass → continue
search
Shift excerpt (PASS-1): [SHIFT at “Check again
...”]
markers: check again
... Move B >1 to avoid collision with D. Check
again for collisions after moving B: No more
collisions detected...
PASS-1 <answer>: C>1,E>1
```

F License

MIT License

Copyright (c) 2025 *Liv d'Aliberti, Manoel Ribeiro*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.