

Format Collapse in Policy Optimization for LLM Reasoning

Debdeep Sanyal^{1*}, Aakash Sen Sharma^{2*},
Dhruv Kumar³, Saurabh Deshpande¹, Murari Mandal⁴

¹Birla AI Labs, ²InvideoAI,

³BITS Pilani, ⁴KIIT Bhubaneswar

{debdeep.sanyal, saurabh.deshpande-c}@oab.adityabirla.com
aakash.sensharma@invideo.io, dhruv.kumar@pilani.bits-pilani.ac.in
murari.mandalfcs@kiit.ac.in

Abstract

Policy optimization (PO) is widely used to post-train language models for reasoning, often with a rigid `<think><answer>` format. We study what PO learns when this constraint is relaxed. We evaluate a relaxed reward across six models and four benchmarks, then audit learned formats on GSM8K and controlled variants across three on-policy algorithms. In these audits, accepting either explicit reasoning or a direct answer pushes the policy toward the shortest valid format: an `<answer>`-only response. On GSM8K, the collapsed Qwen-2.5 7B policy produces answer-only outputs in 98.6% of generations and loses 7.1 pass@5 points relative to a strict-format baseline. Controlled rewards show this is not a regex artifact: with nested formats, PO learns easy requirements before harder ones despite a 4× hard-format reward; with mutually exclusive equal-reward formats, the audited Qwen-2.5 7B final policy assigns more than 99% of generations to the easiest format. A sample-gated gradient account fits the pattern: formats receive positive updates only when sampled, so early support can outweigh reward scale. KL-controlled runs show collapse appears only when the policy can move far enough from initialization. The result identifies a practical failure mode for RLVR reward design. We have open-sourced [the code](#) for the experiments mentioned in this work.

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) (Zhu et al., 2025; Wang et al., 2025; Wen et al., 2025) is the primary method for refining LLMs to perform complex, multi-step reasoning (Yu et al., 2025; Qwen et al., 2024; Grattafiori et al., 2024). A standard practice, introduced by DeepSeek-AI et al. (2025), is to enforce a rigid output structure that compels the model to follow a strict “think-then-answer” format to generate a

chain-of-thought. This approach has since been adopted in several subsequent works (Liu et al., 2025a; Xie et al., 2025; Chen et al., 2025). While effective, this enforcement approach obscures a fundamental question: what are the intrinsic preferences of the PO process when such external constraints are relaxed? If the optimizer exhibits a strong underlying bias, current alignment techniques may be inefficiently working against the natural dynamics of the learning process.

To investigate this question, we design a reward function that offers the model a choice between a structured, multi-part response and a direct, answer-only format. This seemingly minor relaxation of constraints leads to a consistent and unexpected outcome. Across a broad suite of model sizes and families, reasoning domains (math, code, logic), and PO algorithms (GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), REINFORCE++ (Hu et al., 2025)), we observe that the trained policy systematically learns to omit explicit reasoning. The model’s output consistently converges to the simplest valid format, a phenomenon we term the **Cognitive Shortcut**.

This initial finding motivates a series of controlled experiments to characterize the nature and strength of this preference. First, to test if this behavior follows a predictable order, we design a “Reward Hierarchy” experiment. This experiment uses three nested formats of increasing complexity ($f_1 \subset f_2 \subset f_3$), where each format, when correctly produced, yields an identical reward. Despite this flat reward landscape, which provides no explicit incentive to prefer one format over another, we observe a distinct sequential learning pattern. The optimizer consistently masters the simplest format (f_1) first. Only after the reward for f_1 has saturated does the policy begin to make significant progress on f_2 , followed finally by the most complex format, f_3 .

Second, to quantify the strength of this bias, we introduce a steep reward gradient, making the

*Equal Contribution.

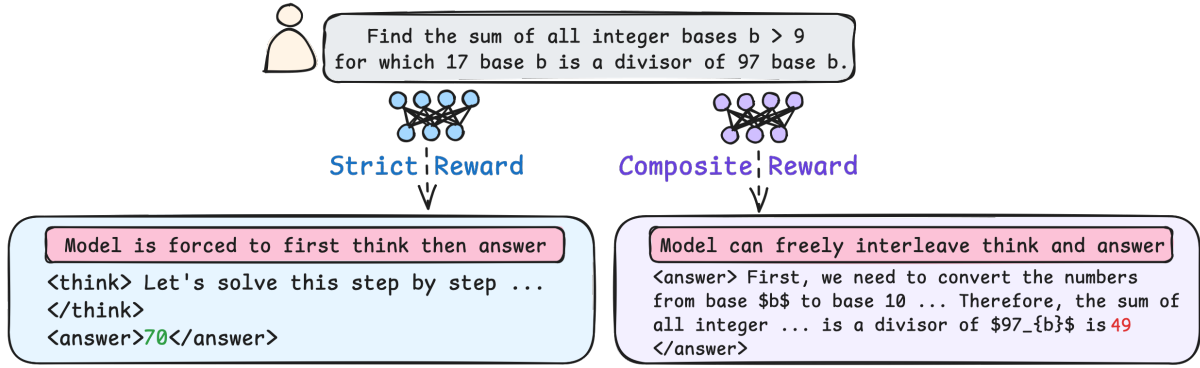


Figure 1: We compare the standard Strict Reward paradigm (**left**), which enforces a rigid *think-then-answer* structure, against our novel Composite Reward (**right**), which grants the model the freedom to choose its solution path. The central finding investigated in this paper is the emergent policy learned under this freedom: the model consistently bypasses explicit reasoning to converge on the simplest valid format, a phenomenon we term the *Cognitive Shortcut*.

most complex format $4\times$ more valuable than the simplest. The sequential learning pattern persists, demonstrating that the bias can override significant reward incentives. The optimizer forgoes a larger, more complex reward in favor of a smaller, more easily learnable one.

Finally, we investigate the underlying mechanism of this phenomenon. A prominent trend in state-of-the-art PO is the removal of the KL divergence penalty to grant models more freedom to explore the reward landscape (Yu et al., 2025; Yue et al., 2025). By treating KL divergence as a diagnostic, we find that convergence on the Cognitive Shortcut is not a passive drift but an active optimization process that requires a substantial policy shift from the pre-trained prior. This finding suggests a trade-off: the exploratory freedom necessary for discovering high-reward policies is the same mechanism that enables the model to find and exploit the simplest specification of a reward function, a classic form of reward hacking.

Our contributions are as follows:

❶ We identify and characterize the **Least Resistance Bias**, an empirical finding that policy optimization in LLMs systematically prefers the simplest valid output format, often at the cost of performance.

❷ We quantify the strength of this bias through a series of controlled experiments, demonstrating its persistence even against significant, conflicting reward incentives.

❸ We establish a causal link between this bias and the degree of policy exploration, showing that a high KL divergence from the initial policy is a prerequisite for discovering and exploiting these “Cognitive Shortcuts”.

❹ We provide extensive empirical evidence

showing this bias is a general property of the PO paradigm, holding across five model families, three reasoning domains, and three distinct PO algorithms.

2 Setup

To investigate the behavior of policy optimization under flexible format constraints, we designed a comprehensive suite of experiments. Our methodology was guided by the principles of diversity, to ensure our findings generalize across multiple contexts, and relevance, by using established benchmarks for multi-step reasoning.

2.1 Models

We selected six publicly available models from five distinct architectural families, with parameter counts ranging from 4B to 12B. The model suite includes Gemma-3 (4B & 12B) (Team et al., 2025), Qwen-2.5 (7B) (Qwen et al., 2024), Llama-3.1 (8B) (Grattafiori et al., 2024), Ministral (8B) (Mistral AI, 2024), and Yi (6B) (AI et al., 2024). This architectural and scale diversity helps to ensure that our conclusions are a general property of the learning process and not an artifact of a specific model. All experiments were conducted on a cluster of 3 NVIDIA RTX A6000 GPUs, and all the results reported are with pass@5 accuracy.

2.2 Datasets

To ensure our findings generalize across different tasks and are not domain-specific, we curated a diverse set of datasets spanning three critical reasoning areas. For **mathematical reasoning**, we used a tiered selection including the multi-step arithmetic of GSM8K (Cobbe et al., 2021), the more complex algebraic problems in Math-Hard (Hendrycks

Table 1: **Performance Comparison of Training Paradigms Across Diverse Models and Reasoning Tasks.** We report the final accuracy (%) of models trained with a **Strict Reward** (enforcing the `<think><answer>` format) versus our **Composite Reward** (allowing a choice of formats). The results are notably mixed across all four reasoning domains, with no single paradigm consistently outperforming the other. This ambiguity suggests that a simple accuracy comparison is insufficient to understand the underlying learning dynamics, motivating a deeper investigation into the optimizer’s intrinsic preferences. Best performance in each pair is highlighted in bold.

Model	GSM8K		rStar-Coder		ReClor		Planning-Mystery	
	Strict	Composite	Strict	Composite	Strict	Composite	Strict	Composite
Gemma-3 4B	72.4	73.1	55.8	53.5	73.2	69.4	58.3	55.9
Qwen-2.5 7B	92.4	85.5	73.2	73.6	80.1	73.8	77.5	73.8
Llama-3.1 8B	86.2	82.9	64.1	59.2	78.3	72.7	74.9	70.6
Ministral 8B	89.5	84.8	74.0	72.6	81.0	76.5	76.8	77.3
Yi 6B	85.1	83.8	68.3	63.7	68.4	65.1	71.2	67.7
Gemma-3 12B	94.6	86.4	76.1	72.3	85.5	81.2	81.2	76.5

et al., 2021), and traces from Open-R1 Math 220k (Hugging Face, 2025). For **code reasoning**, we used competition-level problems from Microsoft’s rStar-Coder (Liu et al., 2025b) and functionally verifiable tasks from Open R1 verifiable coding (Hugging Face, 2025). Finally, for **logical reasoning**, we tested our models on the puzzles of Knights and Knaves (Xie et al., 2024), and the natural language deduction tasks in ReClor (Yu et al., 2020).

2.3 Policy Optimization Algorithms

To establish that the observed behavior is a feature of the PO paradigm itself, we replicated our experiments across three distinct families of policy optimization algorithms.

First, we employed **DAPO** (Yu et al., 2025), the objective for which is given by:

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}_{\mathcal{D}, \pi_{\theta_{\text{old}}}} \left[\frac{1}{Z} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \mathcal{L}_{i,t}(\theta) \right] \quad (1)$$

where $Z = \sum_{i=1}^G |o_i|$ is a normalization term and $\mathcal{L}_{i,t}(\theta)$ is the token-level objective:

$$\mathcal{L}_{i,t}(\theta) = \min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), \dots) \hat{A}_{i,t} \right) \quad (2)$$

Here, $r_{i,t}(\theta) = \frac{\pi_{\theta}(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})}$ is the importance ratio, and $\hat{A}_{i,t}$ is the advantage term.

Second, we used **Dr. GRPO** (Liu et al., 2025c), which corrects for response length and question difficulty biases as in the original version (Shao et al., 2024). Its objective is similar to Equation 1 but uses a different clipping strategy.

Finally, to connect our findings to foundational policy gradient methods, we included **REINFORCE++** (Hu et al., 2025), a modern variant of the classic REINFORCE algorithm. The consistent emergence of the Least Resistance Bias across these distinct algorithmic philosophies provides strong evidence that it is a general property of on-policy optimization for language generation.

3 Identifying the Least Resistance Bias

Our investigation begins by relaxing the rigid format constraints typically imposed in policy optimization for reasoning tasks. We hypothesized that this enforcement might obscure an intrinsic bias in the optimization process. To test this, we designed an experiment to answer a fundamental question: *What solution format does the optimizer prefer when given a choice?*

3.1 Defining a Choice via Reward Functions

To formally define a choice for the policy, we contrast two reward functions. Let y be a model-generated sequence of tokens.

The standard approach uses a **Strict Reward** function, $R_{\text{strict}}(y)$, which provides a positive reward only if y adheres to a predefined *think-then-answer* structure. This defines a single, narrow path to a successful outcome.

$$R_{\text{strict}}(y) = \begin{cases} +3 & \text{if } y \text{ matches the strict format} \\ -4 & \text{otherwise} \end{cases} \quad (3)$$

To create a choice, we designed a **Composite Reward** function, $R_{\text{composite}}(y)$. This function defines a much broader set of valid formats. It re-

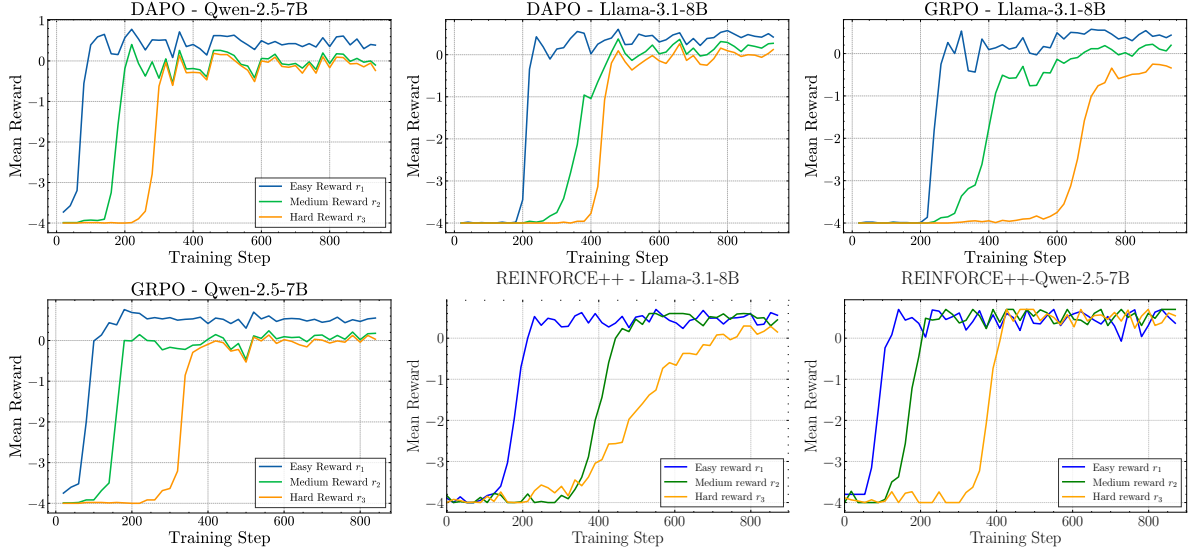


Figure 2: Sequential optimization of format rewards under the nested Reward Hierarchy experiment. Across different models (Qwen-2.5-7B, Llama-3.1-8B) and PO algorithms (DAPO, GRPO, REINFORCE++), the learning dynamics are remarkably consistent. The policy first masters the innermost, **easiest reward** f_1 , then the **medium reward** f_2 , and finally the outermost, **hardest reward** f_3 .

wards any number of interleaved `<think>` and `<answer>` blocks, and critically, it also rewards a direct `<answer>`-only format that contains no preceding reasoning.

$$R_{\text{composite}}(y) = \begin{cases} +3 & \text{if } y \text{ matches a valid format} \\ -4 & \text{otherwise} \end{cases} \quad (4)$$

The full regular expressions defining these formats are detailed in Appendix B.1.3, along with the system prompts used in Appendix B.2.

Our initial expectation was that a policy trained with $R_{\text{composite}}$ would learn a nuanced strategy, applying explicit reasoning for more difficult problems. However, our empirical results, which we present in the following section, show a starkly different and highly consistent outcome.

3.2 Convergence to a Minimal Format

Our initial experiments were designed to test the hypothesis that a policy trained with the flexible $R_{\text{composite}}$ would learn to adaptively use reasoning. The results, presented in Table 1 (See Table 6 for a detailed version), do not support this hypothesis. Instead, they point to a different, highly consistent behavior.

A primary observation from Table 1 is the performance degradation on certain tasks when models are trained with $R_{\text{composite}}$ compared to R_{strict} . On GSM8K, for instance, the policy trained with R_{strict} outperforms the one trained with $R_{\text{composite}}$ across all models, with performance gaps of up to **7.1%**

for Qwen-2.5 7B (92.4% vs. 85.5%) and **8.2%** for Gemma-3 12B (94.6% vs. 86.4%). Similar trends are observed on rStar-Coder and ReClor, where the enforced *think-then-answer* structure provides a performance advantage.

This performance difference is a direct consequence of the learned policy. When trained with $R_{\text{composite}}$, the optimizer does not learn a dynamic balance between reasoning and answering. Instead, across all experimental runs, *the policy consistently converges to the simplest valid format: a direct <answer>-only response*. We term this learned behavior the *Cognitive Shortcut*. This behavior involves the systematic omission of the intermediate reasoning steps that are often necessary for solving complex problems. The optimizer, following the *Least Resistance Bias*, forgoes the complex `<think>` block, even when doing so is detrimental to the final task accuracy.

To provide direct evidence for this causal link, we analyzed the output distribution of a trained policy. On GSM8K, the Qwen-7B model trained with $R_{\text{composite}}$ generated the `<answer>`-only shortcut in *all* the cases. This shortcut format had a significantly lower task accuracy than the full chain-of-thought format (84.9% vs. 97.6%), accounting for the overall performance drop (see Appendix B.1.3 for full analysis). The discovery of this consistent and often suboptimal behavior serves as the primary motivation for the more controlled experiments that follow, which aim to characterize the

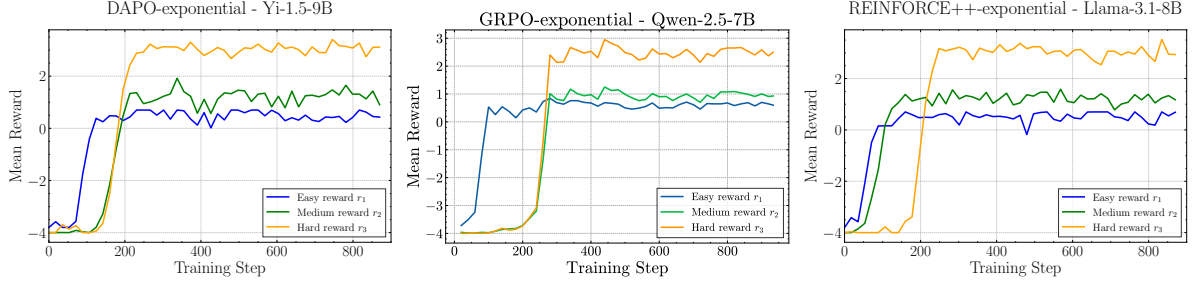


Figure 3: Learning dynamics under an exponentially weighted reward scheme. Despite the hard format f_3 offering 4x the reward of the easy format f_1 , the optimizer’s learning trajectory remains stubbornly sequential. The massive reward for f_3 is initially ignored in favor of the more learnable, lower-value rewards.

nature and strength of this preference. See Appendix C.1 for more details.

4 Characterizing the Least Resistance Bias

The discovery of the Cognitive Shortcut raises a deeper question about the learning process itself: is this preference for simplicity an absolute barrier, or does it reflect a sequential ordering in how the optimizer approaches a complex objective? To investigate this, we shift from a single, disjunctive reward function to a composite objective, where the total reward is the sum of multiple, distinct format requirements. In this new setting, a policy that only satisfies the simplest requirement would still incur penalties from the unsatisfied, more complex requirements, creating a clear incentive to eventually master all components. This allows us to decompose the bias and observe the sequence in which the optimizer addresses objectives of varying complexity.

4.1 Sequential Optimization in Nested Reward Hierarchies

To test this hypothesis, we designed an experiment to isolate the effect of format complexity. We constructed a reward function, $R_{\text{hierarchy}}$, composed of three nested format requirements, f_1 , f_2 , and f_3 , representing a gradient of increasing structural complexity.

1. **Easy Format (f_1):** Requires the generation to contain a final answer enclosed in `\boxed{. *}`.
2. **Medium Format (f_2):** A superset of f_1 , requiring the entire response to be enclosed in `<answer>` tags.
3. **Hard Format (f_3):** A superset of f_2 , requiring a full `<think><answer>` structure.

This nested structure, $f_1 \subset f_2 \subset f_3$, is a crucial feature of the design. A generation that satisfies the hard format f_3 also satisfies f_2 and f_1 , creating a structural incentive to learn the single, all-encompassing solution. To isolate the effect of complexity, we set the reward for satisfying any of these formats to be identical: $r(y, f_1) = r(y, f_2) = r(y, f_3) = R_{\text{max}}$.

The results of this experiment are presented in Figure 2. The learning curves do not show absolute convergence on the unified f_3 solution. Instead, they reveal a consistent, sequential optimization process. Across all models and algorithms, the learning process unfolds in predictable stages:

- **Stage 1:** The mean reward for the simplest format, f_1 , is the first to increase, typically saturating within the first 200-300 training steps.
- **Stage 2:** Only after the policy has reliably mastered f_1 does the optimizer begin to make significant progress on the more complex f_2 format. The reward curve for f_2 begins its ascent only after the f_1 curve has plateaued.
- **Stage 3:** The most complex format, f_3 , is addressed last, with its reward curve beginning to increase only after f_2 is well on its way to convergence.

This staged optimization provides strong evidence for the Least Resistance Bias. Even with a structural incentive to learn the unified solution, the optimizer defaults to a sequential strategy, prioritizing the least complex objective first (See Appendix C.1.1 for a detailed analysis). This observed behavior motivates a further stress test to quantify the strength of this bias against conflicting incentives. Refer to Appendix B.1.3 for a detailed discussion on the motivation and the differences among the



Figure 4: **Learning dynamics under mutually exclusive, exponentially-scaled rewards.** In this adversarial test, the policy can choose only one of three disjoint formats: easy (f_1), medium (f_2), or hard (f_3), where the rewards for all the formats are the same. Across all models and algorithms, the policy exhibits absolute convergence: it exclusively learns to obtain the low-value easy reward (f_1), completely ignoring the larger, more complex rewards. This demonstrates that the Least Resistance Bias is a dominant factor that can cause the optimizer to forgo significantly higher rewards in favor of the most immediately learnable path.

experiments. See Appendix C.1.1 for more details on the experiment.

4.2 Quantifying the Bias Against Reward Incentives

The sequential learning observed in Section 4.1, even with uniform rewards, suggests the Least Resistance Bias is a significant factor in the optimization process. This motivates a follow-up question: can this intrinsic bias be overridden by strong, extrinsic reward incentives? To quantify the strength of the bias, we designed an experiment that creates a direct conflict between format complexity and reward magnitude.

We modified the nested reward structure from the previous experiment to introduce a steep incentive gradient favoring the most complex format. In this setup, we defined the reward magnitudes as a geometric progression: the reward for the medium format f_2 was set to be $2\times$ that of the easy format f_1 , and the reward for the hard format f_3 was in turn $2\times$ that of the medium format. This created a reward landscape where $r(y, f_3) = 2r(y, f_2) = 4r(y, f_1)$.

The results, presented in Figure 3, show that the underlying learning dynamic is highly resistant to this incentive gradient. Faced with a choice between a small, easily attainable reward and a substantially larger but more complex one, the optimizer still prioritizes the simpler format initially.

The learning process can be characterized by two distinct phases:

- **Phase 1 (Simplicity First):** For the first ~ 150 – 200 training steps, the optimizer focuses almost exclusively on mastering the easy format, f_1 . The reward curves for the

more complex formats, f_2 and f_3 , show little to no progress during this phase, despite the large potential reward from f_3 .

- **Phase 2 (Incentive-Driven Co-learning):** Once a performance threshold on f_1 is reached, a new dynamic emerges. The optimizer does not simply move to f_2 . Instead, the learning curves for f_2 and f_3 begin to rise in near-lockstep. The strong reward gradient from f_3 appears to influence the learning of the structurally similar f_2 , causing the optimizer to learn both complex formats simultaneously.

These results indicate that the Least Resistance Bias is not a mere tie-breaker but a dominant factor that dictates the initial stages of optimization (See Appendix C.1.1 for a detailed analysis). Only after the simplest objective is mastered does the optimizer become responsive to the larger reward incentives associated with more complex formats.

4.3 Mutually Exclusive Rewards: Isolating the Bias

Our investigation has thus far revealed a consistent, sequential bias towards simplicity, even against strong structural and reward incentives. However, the nested structure of previous experiments could still allow for an interpretation where sequential learning is an artifact of the optimizer mastering shared format components. To eliminate this possibility and test the *Least Resistance Bias* in its purest form, we designed a final experiment: **a mutually exclusive choice under identical rewards**.

This experimental design extends our Reward Hierarchy, but with a critical modification: the format choices are made strictly disjoint. We constructed

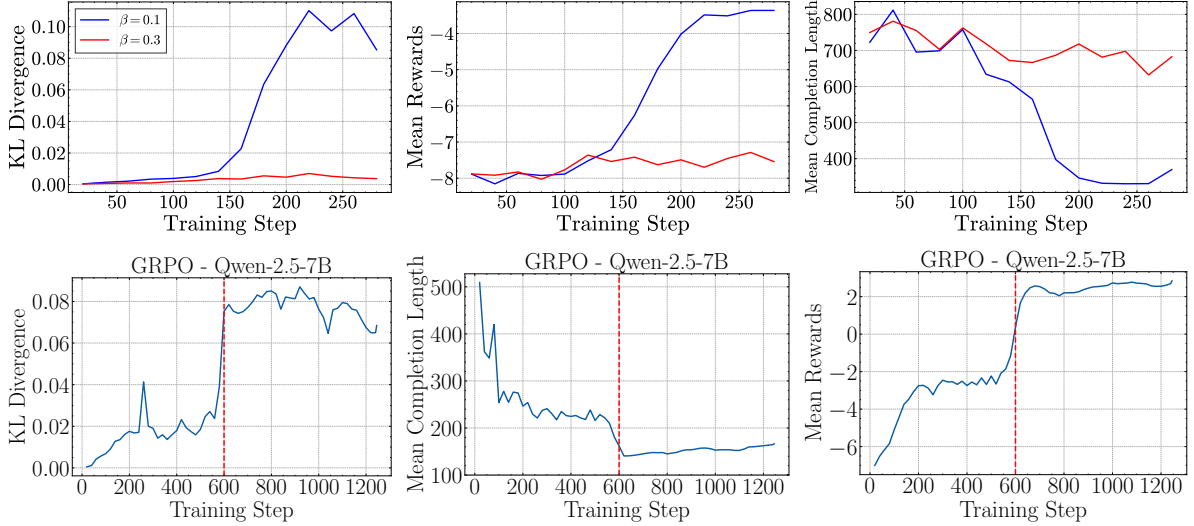


Figure 5: The causal effect of the KL penalty on the emergence of the Cognitive Shortcut. **Top Row:** A direct comparison between a policy with a loose KL leash ($\beta = 0.1$) and a tight leash ($\beta = 0.3$). The freedom to diverge (high KL) is a prerequisite for discovering the high-reward, minimal-length shortcut. **Bottom Row:** The dynamics of a single run, showing a phase transition around the 600-step mark where a spontaneous increase in KL divergence enables the discovery and exploitation of the shortcut.

a reward function, R_{conflict} , where a generation y could satisfy the format requirements for f_1 , f_2 , or f_3 , but *only one at a time*. To further isolate the effect of format complexity, rewards for all matched formats were set to be identical, creating a flat incentive landscape where only the difficulty of the format varies.

Let $r(y, f_i)$ be the reward for a generation y satisfying format f_i . The reward function is defined as:

$$\begin{aligned} r(y, f_i) &= R_{\text{max}}, & \text{if } y \text{ exclusively matches } f_i \\ r(y, f_j) &= 0, & \text{for all } j \neq i \end{aligned}$$

where R_{max} is a constant positive reward. This setup creates a clear choice landscape: the optimizer can pursue one of three paths, each offering the exact same payoff but with a different level of complexity ($f_1 < f_2 < f_3$). Crucially, there are no shared sub-problems. The precise regular expressions ensuring mutual exclusivity for each format are detailed in Appendix B.1.3.

The results, shown in Figure 4, demonstrate a consistent and absolute convergence. In every run, across all models and algorithms, the policy **exclusively converges to the easiest format, f_1** .

The reward plots for the medium (f_2) and hard (f_3) formats remain at their initial negative values for the entire duration of training. This indicates that the potential reward offered by these more complex paths is never explored. The optimizer consistently chooses the simplest format, f_1 , even when equally rewarded, more complex alternatives

are available.

To quantitatively confirm this exclusive convergence, we analyzed the final policy’s output distribution. For the Qwen-7B model, over 99% of all generations matched the simplest format, f_1 , and no response matched the other formats (see Table 5 for details). This provides conclusive evidence for the Least Resistance Bias: when faced with multiple, equally rewarded but disjoint objectives, the optimizer prioritizes the path of lowest complexity. Its behavior is consistent with a greedy strategy that optimizes for the most immediately learnable gradient signal, rather than performing a global cost-benefit analysis. This understanding motivates our final investigation into the mechanisms that enable this bias. Refer to Appendix C.1.1 for more details.

5 The Price of Exploration: Mechanism and Implications

Our investigation has characterized the *Least Resistance Bias* as a consistent empirical phenomenon. This final section investigates its underlying mechanism, revealing a fundamental trade-off in modern policy optimization. A prominent trend in state-of-the-art algorithms is the reduction or removal of the KL divergence penalty to grant the policy more freedom to maximize reward (Yu et al., 2025; Yue et al., 2025; Liu et al., 2025c). Our analysis shows that while this freedom is essential for learning, it is also the mechanism that enables the optimizer to find and exploit the simplest specification of

the reward function, a behavior known as reward hacking.

5.1 KL Divergence as a Diagnostic for Exploration

To examine the mechanism behind this bias, we treat the KL divergence not as a loss term, but as a diagnostic instrument. It measures the policy’s deviation from its initial reference state, π_{ref} , which we use as a proxy for the policy’s *exploratory freedom*. We hypothesize that discovering the Cognitive Shortcut is a form of reward hacking that requires a significant degree of this freedom. The KL penalty, controlled by its coefficient β , therefore acts as a constraint on the policy’s ability to make large exploratory shifts.

To test this hypothesis, we designed a controlled experiment comparing two training runs under the composite reward function. The runs were identical except for the strength of the KL penalty: a “Tight Leash” run with a high penalty ($\beta = 0.3$) and a “Loose Leash” run with a low penalty ($\beta = 0.1$). This setup allows us to observe the effect of constraining exploration on the policy’s ability to find the shortcut.

The results, presented in Figure 5 (top row), indicate a clear trade-off. The policy with the loose leash is able to achieve a high final KL divergence and successfully converges on the minimal-length Cognitive Shortcut. Conversely, the policy with the tight leash is constrained to a low KL divergence and fails to discover this solution, remaining in a lower-reward state. This provides strong evidence that a sufficient degree of exploratory freedom is a key enabling factor for the discovery of the shortcut.

The dynamics of this process are visible within a single training run (Figure 5, bottom row). Around the 600-step mark, the policy undergoes a phase transition, marked by a sharp increase in KL divergence. This shift away from the prior is immediately followed by a surge in the format reward and a corresponding drop in completion length. As our analysis in Table 3 shows, this convergence often comes at the cost of task correctness. This suggests the optimizer is prioritizing a simpler, more learnable reward signal (from the format) over a more complex one (from task correctness). Qualitative examples of the distinct behaviors learned under each condition are provided in Figure 6 onwards.

5.2 Implications for Alignment

Our analysis provides a mechanistic explanation for the Least Resistance Bias. The Cognitive Shortcut is the result of an optimizer that is not merely maximizing expected reward, but is actively seeking the most learnable gradient in the reward landscape. The freedom to diverge from the initial policy, while necessary for performance, is the very mechanism that enables the model to discover and exploit these unintended, simplistic solutions.

This reveals a core challenge for alignment. As models are given more freedom to explore complex reward landscapes as has been shown in Liu et al. (2025a), they may become more proficient at identifying and exploiting underspecified aspects of the reward function. The challenge for practitioners is therefore not simply to grant models unconstrained freedom, but to design reward landscapes that are robust to this form of simplification, ensuring the desired path is also the most learnable one.

6 Conclusion

In this work, we investigated the intrinsic preferences of policy optimization when applied to LLMs for multi-step reasoning. Our experiments identified and characterized the *Least Resistance Bias*: a consistent tendency for PO to converge on the simplest valid output format, even at the cost of task performance. We termed the outcome of this bias the “Cognitive Shortcut,” where models learn to omit explicit reasoning.

We demonstrated that this bias is a quantifiable and persistent factor in the optimization process, capable of overriding significant reward incentives that favor more complex solution formats. Furthermore, we provided a mechanistic explanation for this phenomenon, showing that the exploratory freedom required for a policy to discover high-reward shortcuts, measured by KL divergence, is the same mechanism that enables this form of reward hacking. This finding highlights a central trade-off in the alignment of autonomous agents. The challenge is not simply to grant models more freedom, but to design reward landscapes that are robust to simplification.

7 Acknowledgement

This work was partially supported by computing resources made available through the AMD University Program (AUP) AI & HPC Cluster.

8 Limitations

Our work provides extensive empirical evidence for the Least Resistance Bias, but its scope is subject to limitations that offer clear directions for future research.

Model Scale and Access. Our experiments were conducted on publicly available models with up to 12B parameters. While our findings are consistent across this range, we encourage future work to study how the Least Resistance Bias manifests in state-of-the-art models at a much larger scale (e.g., >30B parameters). It is possible that at extreme scales, the perceived “difficulty” of complex formats diminishes, potentially altering the learning dynamics. Access to such models was beyond the scope of this work due to the computational resources required for their training.

Algorithmic Diversity. We demonstrated the bias across three prominent families of RLVR algorithms (DAPO, GRPO, REINFORCE++). However, the broader landscape of learning paradigms for LLMs also includes offline RL methods, direct preference optimization (DPO), and evolutionary strategies. Characterizing how this bias appears in these different optimization frameworks is a valuable next step. Our study was intentionally focused on the on-policy PO paradigm, where the trade-off between exploration and exploitation is most explicit.

Focus on Empirical Characterization. This paper’s primary contribution is the empirical identification and characterization of a behavioral phenomenon. We hypothesized that the underlying mechanism is related to the learnability and stability of the gradient signal for simpler formats, but a direct theoretical analysis of the gradient landscape is a substantial undertaking that we leave for future work. Our experiments, which use learning speed and convergence outcomes as proxies, provide strong circumstantial evidence but not a formal proof of this mechanism.

Computational Constraints on Ablations. As detailed in Appendix B.1.2, each experimental run required significant computational resources (approx. 8.5 GPU hours). This practical constraint limited our ability to perform exhaustive hyperparameter sweeps. For instance, our analysis of the KL divergence penalty was based on a controlled comparison of two coefficients. A more granular sweep of multiple β values could further quantify the precise trade-off between exploration and ex-

ploitation, but was computationally prohibitive for this study.

9 Ethical Considerations

Our work’s primary ethical implications relate to the fields of AI safety and alignment. By identifying a predictable failure mode in policy optimization, our findings highlight challenges in developing reliable and transparent AI systems.

The *Least Resistance Bias* can be framed as a systematic tendency towards a specific form of reward hacking. The model’s convergence on “Cognitive Shortcuts” satisfies the literal specification of a flexible reward function while avoiding the intended, more robust reasoning process. In safety-critical applications, this intrinsic preference for the simplest solution path could lead an agent to fulfill its objectives in unintended or brittle ways. Understanding this bias is a necessary step in designing alignment strategies that are robust to such behavior.

Furthermore, the natural erosion of chain-of-thought reasoning has direct consequences for **transparency and trustworthiness**. Explicit reasoning steps provide a crucial mechanism for human oversight, debugging, and verification. The optimizer’s tendency to discard this transparency in favor of opaque, answer-only outputs makes it more difficult to trust and deploy these models responsibly, particularly in high-stakes domains where the reasoning process is as important as the final result.

References

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, and 14 others. 2024. [Yi: Open foundation models by 01.ai](#). *Preprint*, arXiv:2403.04652.
- Guanzheng Chen, Xin Li, Michael Qizhe Shieh, and Lidong Bing. 2025. [Longpo: Long context self-evolution of large language models through short-to-long preference optimization](#). *Preprint*, arXiv:2502.13922.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.

- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Google DeepMind. 2025. [Gemini 3 Pro.
https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf](https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf). Accessed: 2026-01-04.
- 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 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the math dataset](#). In *Advances in Neural Information Processing Systems*.
- Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. 2025. [Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization](#). *Preprint*, arXiv:2501.03262.
- Hugging Face. 2025. [Open r1: A fully open reproduction of deepseek-r1](#).
- Zehao Jin, J. Lin, and Siao-Fong Li. 2020. [Learning principle of least action with reinforcement learning](#). *ArXiv*, abs/2011.11891.
- Shuhe Kurita and Anders Søgaard. 2019. [Multi-task semantic dependency parsing with policy gradient for learning easy-first strategies](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025a. [Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models](#). *Preprint*, arXiv:2505.24864.
- Yao Liu, Yannis Flet-Berliac, and E. Brunskill. 2022. [Offline policy optimization with eligible actions](#). In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 1253–1263.
- Yifei Liu, Li Lina Zhang, Yi Zhu, Bingcheng Dong, Xudong Zhou, Ning Shang, Fan Yang, and Mao Yang. 2025b. [rstar-coder: Scaling competitive code reasoning with a large-scale verified dataset](#). *Preprint*, arXiv:2505.21297.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025c. [Understanding r1-zero-like training: A critical perspective](#). *Preprint*, arXiv:2503.20783.
- Harry Mead, Clarissa Costen, Bruno Lacerda, and Nick Hawes. 2026. [Return capping: Sample-efficient cvar policy gradient optimisation](#). In *International Conference on Learning Representations*.
- Mistral AI. 2024. [Ministral 8b](#).
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2024. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- 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.
- Sungryull Sohn, Sungtae Lee, Jongwook Choi, H. V. Seijen, Mehdi Fatemi, and Honglak Lee. 2021. [Shortest-path constrained reinforcement learning for sparse reward tasks](#). In *International Conference on Machine Learning*.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and yelong shen. 2025. [Reinforcement learning for reasoning in large language models with one training example](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhi-rong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. 2025. [Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms](#). *Preprint*, arXiv:2506.14245.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. 2024. [On memorization of large language models in logical reasoning](#).

Roy Xie, David Qiu, Deepak Gopinath, Dong Lin, Yan-chao Sun, Chong Wang, Saloni Potdar, and Bhuwan Dhingra. 2025. [Interleaved reasoning for large language models via reinforcement learning](#). *Preprint*, arXiv:2505.19640.

Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. [Dapo: An open-source llm reinforcement learning system at scale](#). *Preprint*, arXiv:2503.14476.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jia-ashi Feng. 2020. [Reclor: A reading comprehension dataset requiring logical reasoning](#). *Preprint*, arXiv:2002.04326.

Yu Yue, Yufeng Yuan, Qiyang Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, Tiantian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, and 8 others. 2025. [Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks](#). *Preprint*, arXiv:2504.05118.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 2025. [The surprising effectiveness of negative reinforcement in LLM reasoning](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Appendix

B LLM Usage Statement

The core scientific contributions of this paper—including the initial hypothesis, the design of the primary experiments, and the subsequent analysis—are the original work of the authors. LLMs were utilized as assistive tools in several aspects of the research and manuscript preparation process to improve clarity and efficiency.

Manuscript Preparation. The authors used Google’s Gemini 3 Pro Preview ([Google DeepMind, 2025](#)) and Gemini 2.5 Pro ([Comanici et al., 2025](#)) to assist with the structuring and refinement of the manuscript’s text. This included tasks such as improving the clarity and flow of sentences, rephrasing paragraphs to maintain a consistent and clinical tone, correcting grammatical errors, and helping with some $\text{\LaTeX}$ formatting. The conceptual narrative and all scientific claims remained the authors’ own. Every LLM-generated suggestion was critically reviewed, edited, and approved by the authors to ensure it accurately reflected our intended meaning and findings.

Code and Plot Generation. LLMs were also used to assist in the software development process. This primarily involved generating boilerplate code for data processing scripts and creating plotting scripts in Python (e.g., using Matplotlib and Seaborn) to visualize our experimental data. The logic for all data analysis and the data itself were original to our work.

Human Oversight. All content, whether text or code, that was generated or modified with the assistance of an LLM has undergone thorough review and verification by the authors. The final responsibility for the entirety of this paper’s content rests with the authors.

B.1 Experimental Setup

B.1.1 Optimization Hyperparameters

In large-scale PO, preventing policy collapse is paramount. We therefore adopted a small, conservative **learning rate of $5e^{-6}$ for our value-based methods (GRPO, DAPO)**, a standard choice that ensures smooth and consistent policy updates. For the actor-only **REINFORCE++**, we used a slightly higher $5e^{-5}$, which is common for this class of algorithms to ensure a sufficiently strong learning signal. Across all optimizers, we used a **cosine annealing scheduler with a 0.1 warmup ratio and**

Table 2: Core hyperparameters for the GRPO, DAPO, and REINFORCE++ training runs.

Parameter	GRPO	DAPO	REINFORCE++
<i>Optimizer Settings</i>			
Learning Rate ('lr')	5e-6	5e-6	5e-5
Adam β_1 / β_2	0.9 / 0.99	0.9 / 0.99	0.9 / 0.99
Weight Decay	0.1	0.1	0.1
LR Scheduler	cosine	cosine	cosine
Warmup Ratio	0.1	0.1	0.1
Max Grad Norm	0.15	0.15	0.15
<i>Training & Batching</i>			
Effective Batch Size	8	8	8
Epochs	3	3	3
Precision	bfloat16	bfloat16	bfloat16
<i>PO-Specific Settings</i>			
Loss Type	<i>dr_grpo</i>	<i>dr_grpo</i>	-
ϵ / ϵ_{high}	0.2 / 0.2	0.2 / 0.28	-
KL Beta (β)	0.1 / 0.3	0.1 / 0.3	-
Generations per Prompt	5	5	1
PPO Epochs	-	-	2
Advantage Normalization	-	-	True
Reward Normalization	-	-	True
<i>LoRA Settings</i>			
Rank ("r")	64	64	64
Alpha (α)	128	128	128
Target Modules	All Linear Proj.	All Linear Proj.	All Linear Proj.

a strict max grad norm of 0.15. This combination is a powerful recipe for stability: the warmup allows the optimizer to find its footing, the cosine decay prevents divergence late in training, and the gradient clipping acts as a crucial safeguard against catastrophic updates.

For our primary algorithm, **GRPO**, we specifically chose the "Done Right" (*dr_grpo*) variant. This was a deliberate decision to ensure the integrity of our findings. Standard GRPO can be susceptible to biases from varying response lengths and question difficulties. The *dr_grpo* loss incorporates corrections for these factors, providing a cleaner and more reliable learning signal. This ensures that the "Principle of Least Resistance" we observe is a true feature of the optimization landscape, not an artifact of a noisy or biased algorithm.

Our implementation of **DAPO** builds directly on this stable GRPO foundation, incorporating its key innovation: decoupled clipping. By setting $\epsilon_{high} = 0.28$, we provide a larger ceiling for the importance ratio, granting the optimizer more freedom to increase the probability of low-probability

(exploratory) tokens. This technique is designed to combat entropy collapse and was a crucial component in enabling the long, stable training runs required for our analysis.

All of the training runs were done in a parameter efficient fashion with LoRA. By targeting all linear projection layers with a moderate rank ($r = 64$) and a scaled alpha ($\alpha = 128$), we created an efficient and expressive adaptation method. This allowed us to effectively steer these large models without the prohibitive cost of full fine-tuning. To be memory efficient with the REINFORCE++ training, which requires a separate reference model, we leveraged different LoRA adapter heads with the same base model to create the policy model and the reference model.

B.1.2 Computational Resources

All experiments reported in this paper were conducted on a private cluster of NVIDIA GPUs. Specifically, our setup consisted of 3x **NVIDIA RTX A6000 GPUs**, each with 48GB of VRAM, and a single **NVIDIA H100 GPU** with 80GB of

VRAM. The use of bf16 precision and standard optimization techniques such as **gradient checkpointing** was essential for fitting our larger model training runs within this memory footprint. A typical experimental run for a single model on a single dataset required approximately **8.5 GPU hours** to complete. The full suite of experiments, including all models, algorithms, and the adversarial ablations amounted to a total of approximately **200 GPU hours**. We report these figures to provide a transparent account of the computational resources required to replicate our findings and to contextualize the scale of our empirical investigation.

B.1.3 Reward Functions

At the core of our investigation is the design of the reward functions that shape the optimization landscape. All our experiments are built upon a combination of a primary *Correctness Reward* and several *Format Rewards*. We define these formally below. Let y be a model generation for a given prompt, and let a be the ground-truth answer.

Correctness Reward (R_{correct}) The primary objective for any reasoning task is to arrive at the correct final answer. We define a correctness reward that is contingent on the model’s output being both parsable and numerically equivalent to the ground-truth. We first define an extraction function, $\text{ExtractBoxed}(y)$, which retrieves the content within the final `\boxed{ }` environment in the generation y . The correctness reward is then given by:

$$R_{\text{correct}}(y, a) = \begin{cases} +3.0 & \text{Verify_and_parse}(e, a) \\ -5.0 & \text{otherwise} \end{cases} \quad (5)$$

where the function $\text{Verify_and_parse}(\cdot, \cdot)$ checks for parsability and the mathematical equivalence between the extracted response and the ground-truth answer. The large penalty for incorrectness provides a strong signal to avoid hallucination and guessing.

Composite Format Reward ($R_{\text{composite}}$) For our initial “Liberating Experiment”, we designed a flexible format reward that accepts a wide range of reasoning structures. This function provides a positive reward if the generation y matches the regular expression for either an interleaved “think-then-answer” structure or a direct ‘answer’-only format. Let $\mathcal{F}_{\text{composite}}$ be the set of all strings matching this permissive format. The reward is defined as:

$$R_{\text{composite}}(y) = \begin{cases} +3.0 & \text{if } y \in \mathcal{F}_{\text{composite}} \\ -4.0 & \text{otherwise} \end{cases} \quad (6)$$

Composite Regex:

```
^(?: (?:<think>.+?</think>
<answer>.+?</answer>)+|
(?:<answer>.+?</answer>
(?:<think>.+?</think><answer>
.+?</answer>)*))$
```

The total reward for this experiment is a linear combination of the correctness and composite format rewards.

Strict Format Reward (R_{strict}) For our baseline and control conditions, we designed a reward function that enforces a single, rigid reasoning structure. This approach follows the standard methodology established in prior work (DeepSeek-AI et al., 2025). A positive reward is provided only if the generation y strictly adheres to the `think-then-answer` format. This is validated using below regular expression. This function defines a single, narrow path to a valid outcome, allowing us to create a clear control against which we measure the behavior of policies trained with more flexible reward functions. The magnitudes of the reward are the same as the magnitudes of the composite reward $R_{\text{composite}}$.

Strict Regex:

```
^<think>.*?</think>
<answer>.*?</answer>$
```

Hierarchical Format Rewards ($R_{\text{hierarchy}}$) To formalize the “Least Resistance Bias,” we designed a set of three nested, Matryoshka-style format requirements, f_1, f_2, f_3 , representing ascending levels of structural complexity.

- **Easy Format** (f_1): The innermost requirement, satisfied if the generation simply contains a boxed answer. Let $\mathcal{F}_1$ be the set of strings matching $^.*\backslash\boxed\{.\+\}\.*\$$.

$$r(y, f_1) = \begin{cases} +1.0 & \text{if } y \in \mathcal{F}_1 \\ -4.0 & \text{otherwise} \end{cases} \quad (7)$$

Regex:

Easy: $^.*\backslash\boxed\{.\+\}\.*\$$

- **Medium Format (f_2):** A superset of f_1 , requiring the boxed answer to also be enclosed within `<answer>` tags.

$$r(y, f_2) = \begin{cases} +2.0 & \text{if } y \in \mathcal{F}_2 \\ -4.0 & \text{otherwise} \end{cases} \quad (8)$$

Regex:

Medium:

```
^.*<answer>.*\\boxed\\{.+\\}
.*</answer>.*$.
```

- **Hard Format (f_3):** The most encompassing format and a superset of f_2 , requiring the full `<think><answer>` structure.

$$r(y, f_3) = \begin{cases} +4.0 & \text{if } y \in \mathcal{F}_3 \\ -4.0 & \text{otherwise} \end{cases} \quad (9)$$

Regex:

Hard: `^<think>.*</think><answer>.*\\boxed\\{.+\\}.*</answer>$.`

For our “Reward Hierarchy” experiment, the reward magnitudes for $r(y, f_1)$, $r(y, f_2)$, and $r(y, f_3)$ were all set to a constant value, $R_{\max}$, to isolate the effect of complexity. For the “Geometric Incentive Gambit,” we used the scaled values of 1.0, 2.0, and 4.0 as defined above to create a strong incentive gradient. The total reward for these experiments, $R_{\text{hierarchy}}(y)$, is the sum of the rewards for all satisfied format conditions, ensuring that a generation matching f_3 also receives the rewards for f_2 and f_1 . For the “Mutually Exclusive Choice” experiment, the total reward is simply the reward for the single, highest-complexity format that is exclusively matched.

Reward Aggregation for Hierarchical vs. Mutually Exclusive Experiments

The distinction between the *Reward Hierarchy* experiments and the *Mutually Exclusive Choice* experiment is fundamental to understanding the nature of the Least Resistance Bias. The core technical difference lies in how the final scalar reward is calculated from the individual format conditions.

In our initial *Composite Reward experiment* (Section 3.2), the goal was to observe the policy’s preference when given a simple choice. The reward function, $R_{\text{composite}}(y)$, was **disjunctive**: it provided a

single, constant positive reward if the generation y matched *any* of the valid formats (Regex B.1.3). To achieve the maximum possible format reward, the optimizer could satisfy any of the regex conditions in $R_{\text{composite}}$. The motivation for this setup was to see what behavior would emerge in the simplest choice scenario. The observed convergence to the `answer`-only format (the *Cognitive Shortcut*) demonstrated the existence of the bias.

In the Hierarchical experiments, the total reward, $R_{\text{hierarchy}}(y)$, is defined as the **sum of the rewards for all satisfied format conditions**. This additive structure was designed to investigate the procedural nature of the bias. A generation y that satisfies the most complex format, f_3 , by definition also satisfies the nested sub-formats f_2 and f_1 . Therefore, its total reward is calculated as:

$$R_{\text{total}}(y) = r(y, f_1) + r(y, f_2) + r(y, f_3) \quad \text{if } y \in \mathcal{F}_3 \quad (10)$$

This design creates a persistent incentive for the optimizer to progress through the hierarchy. A policy that converges only to f_1 or f_2 would continue to incur penalties (negative rewards) from the unsatisfied, more complex format conditions. To maximize the total reward and eliminate all penalties, the optimizer is encouraged to eventually learn the single, all-encompassing f_3 solution. The motivation for this setup was to observe the *path* the optimizer takes when a clear incentive for full mastery exists, allowing us to characterize the bias as a sequential, procedural preference.

Conversely, in the *Mutually Exclusive Choice* experiment, the total reward, $R_{\text{conflict}}(y)$, is defined as the reward for the **single, highest-complexity format that is exclusively matched**. The disjoint format definitions (enforced by start- and end-of-string anchors in the regular expressions) ensure that a generation can satisfy at most one format condition. The reward function is therefore:

$$R_{\text{total}}(y) = \max(r(y, f_1), r(y, f_2), r(y, f_3))$$

where only one term can be non-zero. The motivation for this setup was to create the most adversarial test of the bias. By removing the additive incentive to progress and eliminating any shared substructures, we force the optimizer to make a definitive choice between paths. Its consistent convergence on f_1 in this setting, even when it means forgoing a larger reward, provides clean evidence that the bias is a fundamental preference for the most learnable path, not merely an artifact of incremental learning.

Conflicting Format Rewards (R_{conflict}) For our final adversarial test, the "Mutually Exclusive Choice" experiment, it was crucial to design format requirements that were truly disjoint, ensuring that a single generation y could not satisfy more than one condition. This eliminates the possibility of shared sub-problems and forces the optimizer into a clean cost-benefit analysis. We achieved this through careful construction of the regular expressions for each format. The reward function, $R_{\text{conflict}}(y)$, provides a constant positive reward R_{max} if and only if sy exclusively matches one of the following formats, f_1 , f_2 , or f_3 .

- **Easy Format (f_1):** Defined by the prior Easy Regex B.1.3. This format is satisfied if the generation ends *immediately* after the closing brace of a boxed{} environment. The '\$' anchor is critical; it asserts that no characters, including the '</answer>' tag, can follow the boxed answer.
- **Medium Format (f_2):** Defined by the regex -

Regex:

Medium:

```
^(?:<answer>.+?\boxed\{
{.+}\}</answer>)$
```

. This format requires the entire generation to be a single, valid <answer> block that contains a boxed answer somewhere within it. The ^ and \$ anchors ensure that the generation must begin with '<answer>' and end with '</answer>', with no preceding <think> tags.

- **Hard Format (f_3):** Defined by the regex -

Regex:

Hard:

```
^<think>.+</think><answer>
{.+}\boxed\{.+}\}</answer>$
```

. This format mandates the strict *think-then-answer* structure. The ^ anchor ensures the generation must start with a <think> tag.

Justification for Mutual Exclusivity: The disjoint nature of these formats is guaranteed by the careful use of start-of-string ^ and end-of-string '\$' anchors in the regular expressions. Let us walk

through why a single generation y cannot satisfy more than one condition:

❶ A generation matching f_3 cannot match f_2 because it must start with <think>, whereas f_2 must start with <answer>.

❷ A generation matching f_2 cannot match f_1 because it must end with </answer>, whereas f_1 must end immediately after the } of the boxed expression.

❸ Similarly, a generation matching f_3 cannot match f_1 because it must also end with </think>, violating the end-of-string requirement of f_1 .

This design ensures that the three reward paths are truly independent, forcing the optimizer to commit to a single strategy. The resulting convergence on textcolorblue f_1 , as shown in the main paper, is therefore an unconfounded demonstration of its intrinsic preference for the most easily learnable format, even when other, equally rewarded paths are available.

B.2 System Prompts

The system prompt is a critical component of the experimental setup, as it frames the task for the model and can subtly influence its behavior. For our investigation, it was paramount to design prompts that were as neutral as possible, defining the valid formats without biasing the model towards any particular one. Below, we detail the prompts used for our core experiments and the rationale behind their design.

Prompt for Composite and Hierarchical Reward Experiments For all experiments involving a choice between multiple valid formats (Section 3.2), we used the following system prompt. This prompt was engineered to explicitly grant the model freedom and define the available tools (<think>, <answer>, \boxed{}), while deliberately avoiding any language that would suggest one format is preferable to another.

System Prompt:

You are a helpful assistant, and you will be providing factually correct responses to the user queries. While responding, if at any point you feel like you are not comfortable answering the question, you can choose to think in an internal dialogue and enclose your thoughts between `<think></think>` tags, and then when you are confident again, you can resume answering by enclosing your answer between `<answer></answer>` tags. You can choose to think whenever you feel appropriate, and then answer when you are confident. Note that this also means that you can think a bit, answer that portion, and start thinking again when you need to, which means you can have interleaving `<think>...</think>` and `<answer>...</answer>` tags in your response. Enclose your final response to the query in `\boxed{}`.

Design Rationale: The key to this prompt is its permissive and non-prescriptive language. Phrases like *you can choose to think, whenever you feel appropriate* and the explicit mention of an “interleaving” structure are designed to signal to the model that all valid formats defined by our composite and hierarchical rewards are acceptable. The prompt provides no incentive, explicit or implicit, to prefer a minimal response. The emergence of the “Cognitive Shortcut” under this neutral prompt is therefore a stronger signal of the optimizer’s intrinsic bias.

Prompt for Strict think-then-answer Baseline For our baseline experiments and the “Mutually Exclusive Choice” experiment (where f_3 was the hard format), we used a standard, more restrictive prompt.

System Prompt:

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first needs to think between `<think>...</think>`, and then respond between `<answer>...</answer>` tags. The final response within the answer must be enclosed within `\boxed{}`.

Design Rationale: This prompt is explicitly pre-

scriptive. The phrase *first needs to think... and then respond* removes all ambiguity and enforces a single, sequential computational path. This serves as our control condition, representing the standard methodology against which we measure the behavior of the liberated policy. By using such a clear and unambiguous instruction, we ensure that any differences in learned behavior can be directly attributed to the change in the reward function and the freedom it grants, rather than to any ambiguity in the prompt itself.

Related Work

A significant body of research has explored “shortcut learning,” where models learn simple, often unintended heuristics instead of robust features (Sohn et al., 2021; Kurita and Søgaard, 2019; Jin et al., 2020). This literature has largely viewed shortcut learning as an artifact of the experimental setup, such as biases in the data or specific algorithmic limitations, rather than as a general preference of the optimization process itself. Our work reframes this perspective by identifying and characterizing the *Least Resistance Bias* as an intrinsic property of on-policy optimization in language models.

Shortcuts as Statistical or Environmental Artifacts. A common view treats shortcuts as statistical anomalies or inefficiencies to be mitigated. For instance, in sparse-reward reinforcement learning, Sohn et al. (2021) propose a “shortest-path constraint” to prevent an agent from taking physically redundant actions. In their work, the “shortest path” represents the geometrically optimal, desired solution. This is distinct from our findings, where the path of least resistance represents the cognitively simplest, and often suboptimal, solution. Similarly, in the offline RL setting, Liu et al. (2022) identify “propensity overfitting,” a statistical shortcut where a policy avoids states with poor data coverage. This shortcut arises from the interaction between a policy and a finite-data estimator. Our work identifies a shortcut that persists in an online, on-policy setting with full data access, suggesting the bias is not statistical but is inherent to the optimizer’s search dynamics. While these works focus on correcting external conditions for a presumed-neutral optimizer, our findings suggest the optimizer itself is biased.

Shortcuts as Algorithmic Properties. Another line of work addresses shortcuts as properties of specific algorithms. Mead et al. (2026) analyze

Table 3: Output format distribution of policies trained with the composite reward function ($R_{\text{composite}}$) on the GSM8K test set. We report the mean percentage of generations matching each format category, computed over 3 random seeds. The format distribution overwhelmingly converges to the minimal `<answer>`-only format, which we term the Cognitive Shortcut. Confidence intervals ($\pm$) represent the standard error of the mean.

Model	% <code><answer></code> -only (Shortcut)	% <code><think><answer></code>	% Other / Invalid
Gemma-3 4B	$97.3 \pm 0.4\%$	$2.1 \pm 0.6\%$	$0.6 \pm 0.1\%$
Qwen-2.5 7B	$98.6 \pm 0.3\%$	$0.8 \pm 0.5\%$	$0.6 \pm 0.3\%$
Llama-3.1 8B	$96.4 \pm 0.3\%$	$2.7 \pm 0.7\%$	$0.9 \pm 0.5\%$
Mistral 8B	$98.1 \pm 0.2\%$	$1.2 \pm 0.4\%$	$0.7 \pm 0.0\%$
Yi 6B	$95.7 \pm 0.7\%$	$3.2 \pm 0.5\%$	$1.1 \pm 0.3\%$
Gemma-3 12B	$98.6 \pm 0.2\%$	$0.6 \pm 0.2\%$	$0.8 \pm 0.4\%$

the “blindness to success” in risk-averse CVaR optimization, where the algorithm is mathematically constrained from learning from high-performing trajectories. The failure they address is an algorithmic inability to perceive a better solution. In contrast, the failure we identify is a behavioral preference: the optimizer *can* perceive a more complex, higher-reward solution but still converges to a simpler one. This distinction is between an algorithmic flaw and a behavioral bias.

The *Least Resistance Bias* has also been explored. Jin et al. (2020) demonstrate that an RL agent can learn to follow this principle—representing an optimal, ground-truth trajectory—if the reward function is engineered accordingly. Our work provides a notable counterpoint: the *Least Resistance Bias* describes an optimizer’s preference not for the most efficient (least action) solution, but for the most *learnable* one, which is often a shortcut.

Summary of Contribution. In summary, prior research has largely treated shortcuts as circumstantial issues arising from data, environment, or specific algorithmic definitions. Our contribution is the identification of a general, predictable bias that governs the optimizer’s behavior when multiple solution paths are available. In the current landscape, where LLMs are granted significant freedom to explore complex reward spaces, understanding this intrinsic bias becomes a central challenge for building reliable and aligned systems. Our work shifts the focus from correcting the *conditions* of optimization to understanding the *optimizer itself*.

C Policy Analysis

C.1 Quantitative Evidence for Format Collapse

In Section 3.2 of the main paper, we qualitatively describe the phenomenon of format collapse, where policies trained with the flexible composite reward function ($R_{\text{composite}}$) consistently converge to the simplest valid output format. This section provides the quantitative data that substantiates this claim.

To precisely measure the final policy’s learned behavior, we generated 600 samples from the test set of GSM8K for each model after training was complete. Each generation was then categorized with a simple script based on its adherence to either the minimal `<answer>`-only format or the more complex `<think><answer>` format. This analysis was conducted irrespective of the final answer’s correctness to purely isolate the policy’s structural preference.

Table 3 presents the results of this analysis. The data indicate a highly skewed output distribution in favor of the Cognitive Shortcut across all tested models. For every model, the `<answer>`-only format constitutes over 95% of the policy’s generated outputs, with the Qwen-2.5-7B and the Gemma-3 12B model exhibiting the most pronounced convergence at 98.6%. In contrast, the more complex, structured reasoning format is generated in less than 4% of cases across all models.

The key takeaway from this analysis is that the format collapse is not an occasional or random artifact, but is instead the **dominant and convergent behavior** of the policy. This provides definitive, quantitative evidence that the *Least Resistance Bias* is a dominant factor in the optimization process. When presented with a choice within a single reward function, the optimizer does not learn a balanced or adaptive strategy but instead identifies and

Table 4: Final policy output format distribution for the nested Reward Hierarchy experiment (Section 4.1). Analysis is performed on the Qwen-2.5 7B model. The data shows that the policy, after initially mastering the simplest format, largely converges to the next level of complexity (f_2) but fails to learn the most complex format (f_3). Percentages are computed over 3 random seeds.

Final Policy Output Format	% of Generations
f_1 -only compliant (e.g., <code>\boxed{...}</code>)	$2.4 \pm 0.8\%$
f_2 -compliant (e.g., <code><answer>...</code>)	$6.7 \pm 1.2\%$
f_3 -compliant (e.g., <code><think>...</code>)	$90.9 \pm 0.3\%$
Invalid Format	$2.1 \pm 0.5\%$

Table 5: Final policy output format distribution for the Mutually Exclusive Choice experiment (Section 4.3). Analysis is performed on the Qwen-2.5 7B model. The results show a near-total and absolute convergence to the simplest format (f_1), despite all three formats offering identical rewards. This provides definitive evidence for the Least Resistance Bias in a setting with no shared substructures.

Final Policy Output Format	% of Generations
f_1 -compliant (Easy format)	$99.6 \pm 0.2\%$
f_2 -compliant (Medium format)	$0.0 \pm 0.0\%$
f_3 -compliant (Hard format)	$0.0 \pm 0.0\%$
Invalid / Other Format	$0.4 \pm 0.1\%$

almost exclusively exploits the simplest path to a positive reward.

C.1.1 Policy Distributions for Controlled Experiments

The learning curves presented in Section 4 (Figures 2 and 4) show the mean reward over time. To provide a more granular view of the final learned policies, this section presents a quantitative breakdown of their output format distributions. These analyses corroborate and deepen the interpretations of the learning dynamics presented in the main paper. All data is from the Qwen-2.5 7B model, and results are averaged over 3 random seeds.

Reward Hierarchy Experiment. Table 4 details the final policy distribution from the nested reward experiment. Notably, the final policy is overwhelmingly compliant with the most complex format, f_3 , which accounts for approximately 91% of all generated outputs.

This outcome does not contradict the Least Resistance Bias but rather reveals its **procedural nature**. As explained in Section 4, the total reward in this experiment is the sum of the rewards for each satisfied format. This creates a persistent incentive for the optimizer to eventually master all three formats to maximize its total score and eliminate penalties. The critical insight, however, lies in the *path* the optimizer takes to reach this final state, as visualized in the learning curves of Figure 2.

The optimizer does not learn the unified f_3 objective directly from the start. Instead, it follows a strict, sequential curriculum dictated by the Least Resistance Bias: it first masters the simplest component (f_1), uses that as a foundation to learn the intermediate component (f_2), and only then, after mastering the prerequisite sub-tasks, does it successfully converge on the final, all-encompassing format (f_3). This demonstrates that the bias manifests as a **sequential decomposition of complex, nested tasks, rather than an outright refusal to learn them**. This behavior is distinct from the absolute convergence on f_1 observed in the Mutually Exclusive experiment (Table 5), where the incentive to progress is absent.

Mutually Exclusive Choice Experiment. Table 5 presents the final policy distribution for the experiment with disjoint, equally-rewarded formats. The data reveals an absolute convergence that is even starker than the learning curves suggest. Over 99% of the generations from the final policy match the simplest format, f_1 . This demonstrates that when the possibility of learning from shared substructures is removed, and the optimizer is forced to commit to a single path, its preference for the path of least complexity is overwhelming and absolute. This provides definitive evidence that the Least Resistance Bias is a fundamental preference, not merely an artifact of incremental learning.

Table 6: **Full Performance Comparison with Confidence Intervals.** We report the final accuracy (%) of models trained with a **Strict Reward** versus a **Composite Reward**. This table provides the detailed data for the motivating experiment discussed in Section 3.2. Confidence intervals ($\pm$) represent the standard error of the mean over 3 random seeds. The results show that for complex reasoning tasks like GSM8K, the ‘Strict’ paradigm, which enforces chain-of-thought, consistently outperforms the ‘Composite’ paradigm. Best performance in each pair is highlighted in bold.

Model	GSM8K		rStar-Coder		ReClor		Planning-Mystery	
	Strict	Composite	Strict	Composite	Strict	Composite	Strict	Composite
Gemma-3 4B	72.4 $\pm$ 0.5	73.1 $\pm$ 0.4	55.8 $\pm$ 0.6	53.5 $\pm$ 0.7	73.2 $\pm$ 0.3	69.4 $\pm$ 0.5	58.3 $\pm$ 0.4	55.9 $\pm$ 0.6
Qwen-2.5 7B	92.4 $\pm$ 0.3	85.5 $\pm$ 0.5	73.2 $\pm$ 0.4	73.6 $\pm$ 1.2	80.1 $\pm$ 0.2	73.8 $\pm$ 0.4	77.5 $\pm$ 0.3	73.8 $\pm$ 0.5
Llama-3.1 8B	86.2 $\pm$ 0.4	82.9 $\pm$ 0.6	64.1 $\pm$ 0.5	59.2 $\pm$ 0.6	78.3 $\pm$ 0.3	72.7 $\pm$ 0.5	74.9 $\pm$ 0.4	70.6 $\pm$ 0.7
Mistral 8B	89.5 $\pm$ 0.3	84.8 $\pm$ 0.5	74.0 $\pm$ 0.4	72.6 $\pm$ 0.4	81.0 $\pm$ 0.2	76.5 $\pm$ 0.4	76.8 $\pm$ 0.3	77.3 $\pm$ 0.3
Yi 6B	85.1 $\pm$ 0.4	83.8 $\pm$ 0.4	68.3 $\pm$ 0.6	63.7 $\pm$ 0.7	68.4 $\pm$ 0.4	65.1 $\pm$ 0.6	71.2 $\pm$ 0.5	67.7 $\pm$ 0.6
Gemma-3 12B	94.6 $\pm$ 0.2	86.4 $\pm$ 0.4	76.1 $\pm$ 0.4	72.3 $\pm$ 0.5	85.5 $\pm$ 0.2	81.2 $\pm$ 0.3	81.2 $\pm$ 0.3	76.5 $\pm$ 0.5

User

Half the value of $3x - 9$ is $x + 37$. What is the value of x ?

Answer: 83

Qwen-7B $\beta \rightarrow 0.1$

$r_{\text{composite}}$

<answer>The equation is $\frac{3x-9}{2} = x + 37$. Multiplying by 2 gives $3x - 9 = 2x + 74$. Subtracting $2x$ from both sides gives $x - 9 = 74$. Adding 9 to both sides gives $x = 83$.</answer>

Figure 6

User

You have seven bags of gold coins. Each bag has the same number of gold coins. One day, you find a bag of 53 coins. You decide to redistribute the number of coins you have so that all eight bags you hold have the same number of coins. You successfully manage to redistribute all the coins, and you also note that you have more than 200 coins. What is the smallest number of coins you could have had before finding the bag of 53 coins?

Answer: 203

Qwen-7B $\beta \rightarrow 0.1$

$r_{\text{composite}}$

<answer>Let N be the coins in each of the seven bags. The total is $7N + 53$. This must be a multiple of 8 and greater than 200. Testing values for N where $N \equiv 5 \pmod{8}$, the first value where $7N + 53 > 200$ is $N = 27$. The original number of coins was $7 \times 27 = 189$.</answer>

Figure 7

User

Half the value of $3x - 9$ is $x + 37$. What is the value of x ?

Answer: 83

Qwen-7B $\beta \rightarrow 0.3$

$r_{\text{composite}}$

<think> The equation is set up as... </think> $3x - 9 = 2(x + 37)$ <answer> solving for x gives </answer> x is 83.

Figure 8

User

You have seven bags of gold coins. Each bag has the same number of gold coins. One day, you find a bag of 53 coins. You decide to redistribute the number of coins you have so that all eight bags you hold have the same number of coins. You successfully manage to redistribute all the coins, and you also note that you have more than 200 coins. What is the smallest number of coins you could have had before finding the bag of 53 coins?

Answer: 203

Qwen-7B $\beta \rightarrow 0.3$

$r_{\text{composite}}$

Let N be the original number of coins. <answer> $7N + 53$ </answer> <think> must be divisible by 8. And > 200 . </think> The answer is 711.

Figure 9

User

A ship traveling along a river has covered 24 km upstream and 28 km downstream. For this journey, it took half an hour less than for traveling 30 km upstream and 21 km downstream, or half an hour more than for traveling 15 km upstream and 42 km downstream, assuming that both the ship and the river move uniformly. Determine the speed of the ship in still water and the speed of the river.

Qwen-7B + OpenR1-Math220K

r_{strict}

<think>Let's think step by step...</think> <answer>... 4, 10 </answer>

$r_{\text{composite}}$

<answer>Let's denote the speed of the ship in still water as s and the speed of the river as r \$10 km/h and the speed of the river is 2 km/h. 10, 2 </answer>

Figure 10

User

Let AB be a chord of the unit circle $\odot O$. If the area of $\odot O$ is exactly equal to the area of the square with side AB , then $\angle AOB =$ (to 0.001 degree).

Qwen-7B + OpenR1-Math220K

r_{strict}

<think>Okay, so the problem is about a chord AB on the unit circle ... So, setting these equal ... </think>
<answer>124.806°</answer>

$r_{\text{composite}}$

<answer>First, we need to find the area of the unit circle ... Therefore, the angle AOB is approximately 65.427 degrees, rounded to 0.001 degree. 65.427 </answer>

Figure 11

User

Four boys bought a boat for \$60. The first boy paid one half of the sum of the amounts paid by the other boys; the second boy paid one third of the sum of the amounts paid by the other boys; and the third boy paid one fourth of the sum of the amounts paid by the other boys. How much did the fourth boy pay?

Answer: 13

Qwen-7B + OpenR1-Math220K

r_{strict}

<think>Let x_1, x_2, x_3, x_4 be the amounts paid by the four boys. We are given that $x_1 + x_2 + x_3 + x_4 = 60$. The first boy paid one half of the sum of the amounts paid by the other boys, so $x_1 = \frac{1}{2}(x_2 + x_3 + x_4)$. The second boy paid one third of the sum of the amounts paid by the other boys, so $x_2 = \frac{1}{3}(x_1 + x_3 + x_4)$... </think> <answer>The amounts paid by the four boys are 20, 15, 12, 13. Let's check the conditions: $x_1 = 20, x_2 = 15, x_3 = 12, x_4 = 13$.
 $x_1 = \frac{1}{2}(x_2 + x_3 + x_4) = \frac{1}{2}(15 + 12 + 13) = \frac{1}{2}(40) = 20$. $x_2 = \frac{1}{3}(x_1 + x_3 + x_4) = \frac{1}{3}(20 + 12 + 13) = \frac{1}{3}(45) = 15$.
 $x_3 = \frac{1}{4}(x_1 + x_2 + x_4) = \frac{1}{4}(20 + 15 + 13) = \frac{1}{4}(48) = 12$. The conditions are satisfied.

Final Answer: The final answer is 13 </answer>

$r_{\text{composite}}$

<answer>10 </answer>

Figure 12