SIMPLEHRL: MINIMAL HIERARCHICAL STRUCTURE IMPROVES OUT-OF-DISTRIBUTION GENERALIZATION IN RL POST-TRAINING OF LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) post-training has become a central paradigm for improving LLM reasoning, yet how the structure of the reasoning trace during training affects out-of-distribution (OOD) transfer remains poorly understood. Hierarchical RL (HRL) offers a natural inductive bias—separating *what to do* from *how to do it*—intended to encourage abstract, transferable reasoning, but HRL has been studied almost entirely outside the LLM RL post-training regime, leaving open whether its structural prior alone is enough to deliver OOD gains in LLMs. We introduce **SimpleHRL**, a deliberately minimal, HRL-inspired structural prior for LLM RL post-training: it imposes a hierarchical plan/execution structure on the reasoning trace during training, with no separate critic, no multi-level reward shaping, and no objectives beyond what flat baselines already use. Despite this minimality, when trained only on mathematics, SimpleHRL improves average OOD Pass@32 by $+4.0$ pp and Pass@1 by $+3.8$ pp over GRPO+CoT and entropy-exploration baselines across a broad OOD benchmark. To understand this effect, we conduct a series of mechanistic analyses and reveal two complementary insights into how hierarchical structure helps: SimpleHRL's plan-level reasoning becomes markedly more task-agnostic, supporting broader and more transferable strategy coverage; and hierarchical training concentrates gradient updates on the pivotal decision tokens that most determine outcome quality. Together, these findings argue for treating the structure of the reasoning trace during training, beyond surface-level CoT formatting, as a productive lever for OOD generalization in LLM post-training.

1 INTRODUCTION

Reinforcement learning (RL) post-training has become a central paradigm for improving LLM reasoning (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025). Recent algorithms such as GRPO (Shao et al., 2024) use policy-gradient optimization with verifiable rewards to unlock non-trivial solution strategies that supervised fine-tuning alone cannot produce (Chu et al., 2025). Beyond in-distribution gains, RL post-training has also been reported to improve performance on held-out tasks to some extent (Guo et al., 2025; Liu et al., 2025). A striking feature of the current recipe, however, is what it leaves unspecified: the model is free to interleave high-level planning, low-level computation, and verification within an undifferentiated token sequence, and the gradient signal acts uniformly across this stream. Without an explicit separation between abstract strategic decisions and task-specific tactical operations, the policy may reach correct answers in ways that conflate reusable reasoning strategies with problem-specific procedures, raising a basic question: *does the structural format of the reasoning trace produced during training matter for cross-task transfer?*

Hierarchical reinforcement learning (HRL) (Sutton et al., 1999; Parr & Russell, 1998) offers a principled alternative: by imposing an explicit separation between high-level planning (choosing *what* to do) and low-level execution (*how* to do it), HRL creates a structural bottleneck that ought to encourage abstract, reusable strategies. HRL has shown promise for improving transfer in classical RL settings (Vezhnevets et al., 2017), and has begun to be explored in the LLM regime (Wang et al., 2025), typically alongside additional optimization mechanisms (for example, advantage shaping at planning tokens). Existing LLM-side work, however, focuses on demonstrating empirical gains and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

leaves open *why* a hierarchical reasoning structure should help; the underlying mechanism by which hierarchical structure benefits LLM RL post-training has not, to our knowledge, been systematically analyzed. This in turn entangles two separable questions: does the hierarchical structure on the reasoning trace itself improve generalization, and through what mechanism does this benefit arise—a question that further bears on whether additional algorithmic machinery is necessary to realize it?

To address this gap, we introduce **SimpleHRL**: a deliberately minimal, HRL-inspired structural prior that strips HRL down to its essence—a hierarchical plan/execution structure over the reasoning trace, induced through a training-time prompt alone, with no separate critic, no multi-level reward shaping, and no additional objectives beyond what flat baselines already use. The minimalism is intentional: by intervening only at the level of the reasoning structure, the gap between SimpleHRL and a flat chain-of-thought baseline becomes a clean test of whether the structural prior itself is enough. Trained exclusively on mathematics, SimpleHRL achieves $+\mathbf{3.7}$ pp average OOD Pass@32 over a flat-RL baseline (GRPO with chain-of-thought prompting, abbreviated GRPO+CoT) at the 4B scale and $+\mathbf{4.2}$ pp at the 8B scale, on a diverse OOD benchmark spanning code, logic, science, and table reasoning; SimpleHRL also outperforms entropy-based exploration baselines (Cheng et al., 2025a).

To understand why this minimal intervention yields consistent OOD gains, we conduct a series of mechanistic analyses and find two complementary insights into how hierarchical structure helps. *Strategy-space exploration*: hierarchical training drives exploration of a broader, more transferable space of abstract reasoning patterns. SimpleHRL's planning steps become highly task-agnostic, a property GRPO+CoT, despite producing step-structured reasoning, fails to produce. *Credit assignment*: SimpleHRL concentrates gradient updates on pivotal tokens (Abdin et al., 2024)—those with the largest impact on outcome probability—more effectively than the flat baseline.

Our contributions are:

- **Method.** We introduce SimpleHRL, a deliberately minimal, HRL-inspired structural prior for LLM RL post-training that retains only the prior on the reasoning trace—no architectural changes, no separate critic, no additional RL objectives—and achieves substantial OOD-generalization gains over flat baselines at two model scales (Sections 3–4).
- **Mechanistic insight.** We provide a multi-lens analysis of why hierarchical structure improves OOD generalization, revealing (a) broader exploration of task-agnostic abstract reasoning strategies and (b) improved concentration of gradient signal on pivotal decision tokens (Section 5). Together, these findings argue for treating the structure of the reasoning trace during training—not merely the format of the chain-of-thought at inference—as a first-class lever for RL post-training.

2 Related Work

Reinforcement learning post-training for LLMs. Reinforcement learning from human feedback (Christiano et al., 2017; Ouyang et al., 2022) established RL as a tool for aligning LLM outputs with human preferences. More recently, outcome-reward RL trains LLMs directly on binary signals without explicit reward modeling. GRPO (Shao et al., 2024) removed the need for a separate critic by using group-relative advantage normalization as a non-parametric baseline for policy gradient on reasoning tasks; DeepSeek-R1 (Guo et al., 2025) demonstrated that RL post-training can produce reasoning capabilities competitive with proprietary reasoning-focused models such as o1 (OpenAI, 2024) at scale. A recurring challenge in this paradigm is the exploration-exploitation tradeoff: policies during training can collapse onto narrow, high-reward trajectories, curtailing broader exploration and converging prematurely to local optima (Cheng et al., 2025a). DAPO (Yu et al., 2025) introduces asymmetric clipping to encourage exploration, and entropy regularization (Cheng et al., 2025a) adds token-level entropy bonuses to encourage diversity. Our findings suggest that token-level diversity alone may be insufficient to induce the high-level reasoning strategies that transfer in our setting. There is also ongoing debate on whether RL post-training genuinely expands a model's reasoning capability beyond what is already in the base policy: Chu et al. (2025) and Liu et al. (2025) report that RL transfers and even extends reasoning to new tasks, while Yue et al. (2025) argue that RL primarily sharpens sampling within the base model's existing reasoning support rather than introducing fundamentally new patterns. Our findings suggest that the structural format of reasoning during training

(flat vs. hierarchical) is itself a meaningful factor in OOD transfer, complementary to algorithmic choices.

Hierarchical reinforcement learning. Classical HRL decomposes a policy into a manager selecting subgoals and a worker executing them over temporally extended action sequences (options) (Sutton et al., 1999). The option-critic framework (Bacon et al., 2017) jointly learns option selection, intra-option policies, and termination conditions end-to-end. Feudal Networks (Vezhnevets et al., 2017) further instantiate this with hierarchical architectures where managers operate at coarser time scales. A key insight across this literature is that hierarchical decomposition, when the hierarchy captures a meaningful semantic split, tends to improve transfer because abstract subgoals are more domain-invariant than specific action sequences (Nachum et al., 2018). In the LLM setting, HICRA (Wang et al., 2025) applies hierarchical objectives via token-level advantage shaping on planning tokens; the hierarchical structure is thus still enforced implicitly through token-level exploration, and the work focuses on demonstrating empirical gains rather than analyzing why hierarchical structure helps. Our work takes the opposite approach: SimpleHRL uses only a prompt-induced structure with standard GRPO, deliberately ablating all additional hierarchical objectives, and accompanies this with mechanistic analyses aimed at *why* the structural prior helps. That this minimal formulation already achieves OOD benefits suggests that structural inductive bias may account for a substantial part of the observed benefit. The pre-trained LLM's prior on natural step boundaries makes it possible to omit classical components like learned termination functions; we discuss this in detail in Section 3 and Appendix A.

Chain-of-thought reasoning and its limits. Prompting LLMs to reason step-by-step (Wei et al., 2022; Kojima et al., 2022) substantially improves performance on complex tasks, and structured variants such as Tree-of-Thoughts (Yao et al., 2023), Plan-and-Solve (Wang et al., 2023), and scratchpads (Nye et al., 2021) further extend this. A key open question is *whether prompting style constitutes a mere surface-level formatting choice or gives rise to qualitatively distinct reasoning processes*. Our analysis addresses this question in the RL training setting: GRPO trained and evaluated with a CoT prompt does not produce the representational bifurcation that hierarchical training does—plan-step embeddings remain as task-specific as the underlying flat optimization objective, even when surface step structure is present. This is consistent with the broader observation that surface-level step formatting and the latent reasoning representation shaped by training are distinct factors, and need not move together.

3 METHOD: SIMPLEHRL

3.1 MOTIVATION AND DESIGN PRINCIPLE

We consider the standard RL post-training setup (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025): given a problem prompt x, the model generates a complete response y and receives a binary outcome reward $R(x, y)$ indicating correctness. No step-level annotations or intermediate supervision signals are available; the only feedback is whether the final answer is correct. The policy is optimized via policy gradient to maximize this sparse reward.

In this setup, the response y is an undifferentiated token sequence: the model freely interleaves high-level planning, low-level computation, and verification, and the learning signal is applied uniformly across all tokens. Without an explicit separation between *what* to do (planning) and *how* to do it (execution), the model may acquire reasoning patterns that bundle abstract strategy with domain-specific computation as a single monolithic unit. Since the reward does not penalize domain-coupled solutions as long as they succeed on training problems, the policy may learn problem-specific procedures that do not transfer, rather than reusable reasoning patterns that generalize across domains.

SimpleHRL addresses this by imposing a plan/execution hierarchy as a *structural prior on the reasoning trace*, induced through the training prompt that asks the model to articulate a high-level plan before each execution step. Crucially, we do not modify the RL algorithm: the same RL substrate (GRPO with the standard outcome reward, see Section 3.2) is shared with our flat baselines; only the structural prior on the reasoning trace differs. This deliberately isolates the contribution of structural inductive bias from confounds such as additional reward signals, auxiliary objectives, or alternative optimizers used elsewhere in the HRL literature.

3.2 Training Objective and Implementation

The structural prior is induced by a fixed prefix prepended to the user query, which asks the model to articulate a brief high-level plan before each execution step (full prompt in Appendix E). The model decides the number of steps and their granularity from context; the markup syntax is an implementation detail. We then train with GRPO (Shao et al., 2024) augmented with the clip-higher technique from DAPO (Yu et al., 2025). For a prompt x, we sample n rollouts $\{y^{(i)}\}_{i=1}^{n}$ and compute normalized group advantages:

$$\hat{A}^{(i)} = \frac{R(x, y^{(i)}) - \bar{R}}{\sigma_R}, \quad \bar{R} = \frac{1}{n}\sum_{j} R(x, y^{(j)}), \quad \sigma_R = \mathrm{std}\{R(x, y^{(j)})\}, \tag{1}$$

where $R(x, y) = R_{\text{outcome}}(x, y)$ is the binary outcome reward. The policy gradient objective is:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}_{x,y^{(i)}}\left[\sum_{t} \min\Big(\rho_t^{(i)}\,\hat{A}^{(i)},\ \mathrm{clip}\big(\rho_t^{(i)},\,1-\varepsilon,\,1+\varepsilon'\big)\,\hat{A}^{(i)}\Big)\right], \quad \rho_t^{(i)} = \frac{\pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)})}{\pi_{\theta_{\text{old}}}(y_t^{(i)} \mid x, y_{<t}^{(i)})}, \tag{2}$$

where $\varepsilon < \varepsilon'$ (clip-higher) prevents over-suppression of high-advantage tokens. This is the only training objective: the planning and execution are trained jointly with no separate option-level loss.

3.3 Connection to Hierarchical RL: a Conceptual Mapping

Although SimpleHRL uses a single monolithic policy and adds no HRL-specific machinery, its prompt-induced structure maps cleanly onto the classical options framework (Sutton et al., 1999; Bacon et al., 2017): each plan-execute cycle plays the role of a temporally extended action, with the high-level plan content corresponding to option selection, the execution trace to intra-option behavior, and step-boundary markers to implicit termination. We omit the explicit option-critic machinery—learned termination, manager value function, option-level loss—to isolate the contribution of the structural prior alone, and rely on the pre-trained LLM's prior over natural reasoning boundaries to substitute for the omitted termination mechanism. A detailed component-by-component mapping is given in Appendix A.

4 Experiments

We first examine the empirical performance of SimpleHRL versus flat-RL baselines on in-distribution (ID) mathematics and across a diverse out-of-distribution (OOD) benchmark, at two model scales. The goal of this section is to establish *whether* hierarchical training improves generalization. Section 4.3 then takes an initial look at *where* the improvement comes from, ruling out alternative explanations such as the hierarchical prompt at inference time and response length.

4.1 Experimental Setup

Datasets and ID/OOD evaluation protocol. Following the training-data setup of Zheng et al. (2025), all methods are trained exclusively on mathematical reasoning data, combining LightEval-MATH and DAPO-Math (Yu et al., 2025). We then evaluate along two axes: *(i) in-distribution* on standard mathematical reasoning benchmarks (MATH-500, AIME, AMC, Gaokao, Minerva, OlympiadBench), and *(ii) out-of-distribution* on a diverse held-out benchmark (Cheng et al., 2025b) spanning code generation (HumanEval, MBPP), logical puzzles (Zebra Puzzle), simulation (CodeIO), scientific reasoning (GPQA Diamond, SuperGPQA), table understanding (FinQA, HiTab, MultiHier), and open-domain instruction following (LiveBench).

Baselines. We compare against three representative baselines:

- **GRPO+CoT** (Shao et al., 2024): Our primary flat-RL baseline. The same RL substrate as SimpleHRL (GRPO with the clip-higher technique from DAPO (Yu et al., 2025) for improved empirical performance), trained and evaluated under a flat reasoning structure induced by a "Let's solve the problem step by step" CoT prompt.
- **Entropy Adv.** (Cheng et al., 2025a): Augments the GRPO advantage with a token-level entropy bonus to encourage exploration, using the same CoT reasoning structure as GRPO+CoT.

Table 1: **Main evaluation results** (Pass@32 and Pass@1, in %) across in-distribution and out-of-distribution tasks. SimpleHRL improves OOD performance at both model scales while preserving in-distribution accuracy. SimpleHRL rows are highlighted in light blue.

		In-distribution							*Out-of-distribution*										
Method	Metric	*MATH-500*	*AIME*	*AMC*	*Gaokao*	*Minerva*	*Olympiad*	**ID Avg**	*HumanEval*	*MBPP*	*Zebra*	*LiveBench*	*CodeIO*	*GPQA-D*	*SuperGPQA*	*FinQA*	*HiTab*	*MultiHier*	**OOD Avg**
Qwen3-4B																			
GRPO+CoT (Shao et al., 2024)	Pass@32	**97.7**	67.4	89.1	**88.4**	57.6	79.0	79.9	97.6	83.0	33.1	60.0	20.5	82.8	**76.9**	**67.0**	85.2	69.0	67.5
	Pass@1	**94.7**	53.8	85.6	83.9	52.3	**71.3**	73.6	78.2	67.1	4.6	43.2	6.6	54.2	44.0	46.9	63.3	46.5	45.5
Entropy Adv. (Cheng et al., 2025a)	Pass@32	97.5	71.2	**92.8**	87.8	59.2	77.9	81.1	98.2	82.0	44.9	59.8	26.5	**84.3**	75.6	65.7	**86.0**	**71.4**	69.4
	Pass@1	94.4	**55.2**	85.5	84.4	52.2	69.0	73.5	81.4	67.0	**11.4**	**46.2**	10.5	55.1	**44.9**	44.9	65.6	50.6	47.8
SimpleHRL	Pass@32	97.5	**73.9**	91.2	88.0	**59.4**	**79.9**	**81.6**	**98.8**	**94.4**	**45.6**	**63.1**	**29.0**	84.3	74.8	66.7	85.6	69.9	**71.2**
	Pass@1	**94.7**	55.1	**85.7**	**84.5**	**52.8**	70.6	**73.9**	**90.2**	**77.6**	10.6	42.8	**18.4**	**55.2**	42.2	**47.3**	**70.8**	**51.6**	**50.7**
Qwen3-8B																			
GRPO+CoT (Shao et al., 2024)	Pass@32	**99.2**	83.3	**96.4**	89.4	59.6	84.1	85.3	**98.2**	86.2	36.5	61.2	19.5	88.4	**80.2**	62.8	86.5	**72.0**	69.1
	Pass@1	**95.4**	52.9	84.6	84.0	**48.5**	**70.5**	72.6	89.0	70.3	5.6	49.2	6.3	**60.3**	**49.1**	45.6	68.0	49.8	49.3
Entropy Adv. (Cheng et al., 2025a)	Pass@32	99.2	80.0	96.4	**91.7**	59.2	**85.6**	85.3	98.2	85.4	36.4	60.9	28.0	87.9	78.6	**64.3**	85.6	70.8	69.6
	Pass@1	94.8	53.6	83.9	82.5	45.6	70.1	71.8	**89.3**	70.9	7.4	47.3	10.4	58.7	47.8	45.7	69.1	48.6	49.5
SimpleHRL	Pass@32	99.0	**86.7**	94.0	91.4	**60.7**	83.2	**85.8**	98.2	**96.8**	**48.6**	**72.3**	**28.5**	**90.4**	78.8	62.0	**86.6**	70.8	**73.3**
	Pass@1	95.3	**58.4**	**85.7**	**84.5**	47.5	67.9	**73.2**	85.4	**74.2**	**9.5**	**50.6**	**18.3**	59.6	46.7	**46.7**	**72.8**	**53.6**	**51.7**

- **HRL-prompt only**: The SimpleHRL hierarchical prompt applied *only at inference time*, on top of a GRPO+CoT-trained model or the untrained base model. This baseline isolates the effect of the hierarchical prompt alone from the effect of training under hierarchical structure.

We experiment at two model scales, Qwen3-4B and Qwen3-8B (Yang et al., 2025), on the VeRL framework (Sheng et al., 2025). All methods share the same base model at each scale and a common set of hyperparameters; full implementation details are given in Appendix D.

Evaluation metrics. We adopt Pass@k (Chen et al., 2021) as our primary metric, which measures the probability that at least one correct solution appears in k independent samples (estimated from $n = 32$ rollouts). Pass@k captures not only accuracy but also the *diversity and upper bound* of reasoning capability under inference-time scaling, making it more sensitive to exploration quality than Pass@1 alone.

4.2 Main Results

SimpleHRL consistently improves OOD generalization across diverse tasks and model scales. Table 1 reports Pass@32 and Pass@1 for all three methods across both model sizes. SimpleHRL achieves the highest OOD average at both scales: **+3.7 pp** over GRPO+CoT at 4B (Pass@32) and **+4.2 pp** at 8B. At Pass@32 the gains are most pronounced on tasks requiring diverse reasoning: code generation (MBPP 4B: +11.4 pp; 8B: +10.6 pp), logical puzzles (Zebra 4B: +12.5 pp; 8B: +12.1 pp), and simulation (CodeIO 4B: +8.5 pp; 8B: +9.0 pp). The advantage is consistent across both model scales.On the in-distribution math evaluation, all methods retain comparable performance indicating no apparent trade-off between ID and OOD performance. SimpleHRL also outperforms the entropy-based exploration baseline across most OOD tasks, suggesting that token-level entropy exploration alone is not sufficient for learning transferable high-level reasoning patterns in this setting.

4.3 Additional Analysis

Given the consistent OOD generalization gain reported above, we run an initial set of analyses to narrow down where the improvement comes from. We rule out two natural alternative explanations—inference-time prompt effects and response length—as the cause of the gap.

Hierarchical training, not inference-time prompting, is what matters. We isolate the effect of *training* from that of the prompt itself with two complementary pieces of evidence. **(i) Gradual OOD improvement during training.** Figure 5 tracks OOD Pass@k across training steps. The untrained base model evaluated under both prompts (CoT and SimpleHRL) starts from comparable performance, so the *hierarchical prompt alone has minimal effect at step 0*. SimpleHRL then improves monotonically: at early checkpoints it already outperforms GRPO+CoT on most tasks, and the gap widens with further training. The benefit therefore appears to be acquired during training rather than induced solely by the inference-time prompt. **(ii) HRL-prompt only does not close the gap.** Applying the SimpleHRL prompt to a GRPO+CoT-trained model at inference time gives a small gain

over plain GRPO+CoT on some tasks, but still substantially underperforms the SimpleHRL-trained model across all OOD tasks. Combined with the comparable performance of the two prompts on the untrained model, this supports the interpretation that the gain comes from *training under* the hierarchical structure rather than from the structural prompt at inference alone.

Response length: a co-occurring property, not the driver of the gap. SimpleHRL produces longer responses than GRPO+CoT (cross-task mean: $2{,}531$ vs. $1{,}525$ tokens), and longer responses have been associated with stronger reasoning in some settings (Cheng et al., 2025a). To probe whether length itself is the operative variable, in Appendix C, we evaluate an additional baseline, GRPO+long-CoT: the same flat GRPO+CoT model evaluated with an inference-time prompt that explicitly asks for longer, more deliberate reasoning (e.g., *"Take your time and think through this carefully and thoroughly, step by step…"*). The long-CoT prompt successfully lengthens responses on average, yet cross-task OOD accuracy is essentially flat and slightly lower. Pooled across all rollouts in this analysis, the correlation between response length and per-rollout correctness is mildly *negative* ($r = -0.15$). We read these as suggestive rather than definitive evidence that length is a co-occurring property of SimpleHRL's reasoning, not the main driver of its OOD advantage.

5 Why Does SimpleHRL Work? A Multi-Lens Analysis

The empirical results establish that SimpleHRL improves OOD generalization. To investigate possible mechanisms behind this gap, we analyze the trained models through two complementary lenses. **Lens 1: Transferable reasoning strategies** (Section 5.1): we present evidence consistent with SimpleHRL's high-level reasoning-strategy space being more task-agnostic and more similar across related tasks than that of flat baselines. **Lens 2: Credit assignment** (Section 5.2): we provide evidence that, under the SimpleHRL prompt, training signal (gradient norm) is more concentrated on pivotal tokens—the decisions most associated with outcome quality.

5.1 Lens 1: SimpleHRL Discovers and Learns Transferable High-Level Reasoning Strategies

Hypothesis. We hypothesize that SimpleHRL's explicit planning phase forces the model to operate at an abstract level (e.g., articulating *"perform algebraic simplification"* before doing the algebra), so the high-level strategies visited during training are governed by the grammar of abstract problem solving rather than the surface features of the training-domain mathematics. Flat baselines, including those with token-level exploration bonuses, lack this forced abstraction and may reach correct answers through patterns tightly tied to training-domain specifics. Abstract strategies learned on math should then transfer to OOD tasks where similar abstractions apply.

We test the hypothesis with **two complementary measures** of cross-task similarity: **(i) behavioral**—do the high-level reasoning patterns the model produces behave similarly across tasks where transferability is plausible?—and **(ii) semantic-embedding**—do reasoning steps inhabit overlapping regions of a strong embedding space when they share the same underlying strategy?

Reasoning-step segmentation. Both measures take *reasoning steps* as their unit of analysis, so we first segment each response with a frontier LLM (Claude Sonnet 4.5) and then run the embedding and reasoning-pattern analyses on the resulting steps. SimpleHRL imposes a hierarchical structure—high-level planning followed by low-level execution—so we analyze it under two views: *plan-only* content (the high-level reasoning steps alone) and *plan+exec* content (each plan together with its execution). Flat baselines do not separate high- and low-level steps explicitly, so they undergo the identical segmentation procedure as the *plan+exec* view: both yield a single sequence of unsegregated reasoning steps.

Behavioral evidence: similarity of reasoning patterns. Inspired by Jiang et al. (2025), we extract a directed reasoning graph from each model response. Each node is a segmented reasoning step, classified into one of 10 high-level reasoning-strategy types from a fixed taxonomy (e.g., *problem understanding*, *verification*; full taxonomy in Appendix B); edges follow the logical flow of the reasoning trace (not necessarily linear: branching, backtracking, and revisits are preserved). From this graph, we compute the empirical transition matrix $\hat{P}(\text{type}_{t+1} \mid \text{type}_t)$—the probability that a

step of a given type is followed by a step of each other type—aggregated over 50 sample rollouts per task.

Two systematic differences emerge between the methods (Figure 1). **(i) Deliberative vs. explorative bias.** SimpleHRL allocates $+1.6$ pp probability mass to *verification*, $+2.3$ pp to *domain-knowledge invocation*, and $+1.4$ pp to *problem understanding* relative to GRPO+CoT, while GRPO+CoT spends $+3.6$ pp more on *exploration*. **(ii) Verify-and-commit vs. explore-and-loop closure.** SimpleHRL routes verification toward conclusion ($+0.11$ in $P(\text{conclusion} \mid \text{verification})$); GRPO+CoT routes it back into exploration, suggesting that GRPO+CoT performs surface-level (often redundant) exploration rather than committing to verified conclusions.

Beyond *which* strategies dominate, we ask whether SimpleHRL reuses similar reasoning patterns on related tasks where transfer is plausible. Figure 2b computes the pairwise cosine similarity between row-normalized per-task transition matrices. SimpleHRL has higher pairwise transition-matrix similarity than GRPO+CoT (mean off-diagonal 0.54 vs. 0.49), and the gap widens precisely on the natural transfer clusters: the table-heavy tasks (*FinQA*, *HiTab*, *MultiHier*), the coding tasks (*HumanEval*, *MBPP*), and the scientific-reasoning pair (*GPQA-D*, *SuperGPQA*). Within each of these clusters, SimpleHRL's per-task reasoning patterns remain mutually similar—i.e., similar strategies are reused on related OOD tasks—while GRPO+CoT's patterns drift further apart even between tasks of the same family.

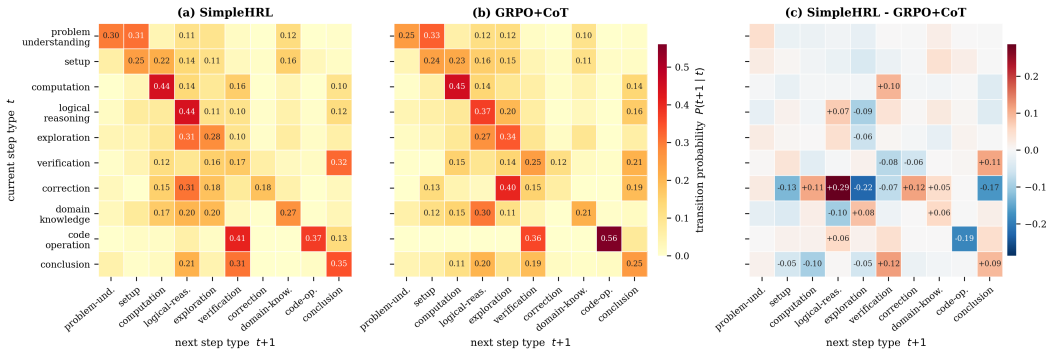


Figure 1: **Reasoning-pattern transition probabilities, aggregated over the OOD benchmark.** Each cell is the empirical $\hat{P}(\text{type}_{t+1} \mid \text{type}_t)$. **(a)** SimpleHRL concentrates probability on deliberative transitions (e.g., *verification→conclusion*); **(b)** GRPO+CoT spreads more mass into *exploration* and loops verification back into exploration; **(c)** difference $\Delta P = P_{\text{SimpleHRL}} - P_{\text{GRPO+CoT}}$.

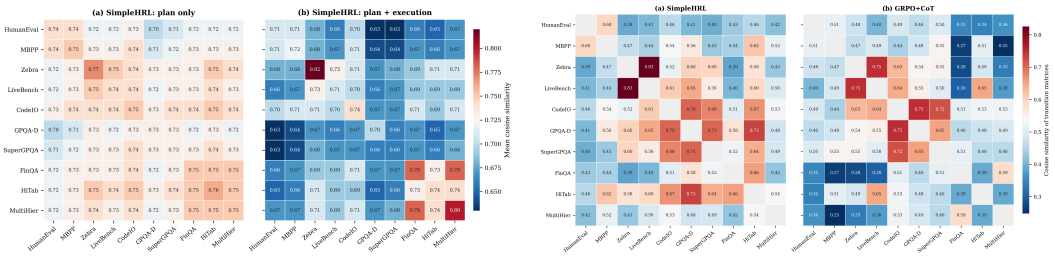
Semantic-embedding evidence: reasoning-step similarity. We embed each segmented step with Qwen3-Embedding-4B and characterize the resulting space using three metrics: (1) the *Cross/Within cosine ratio* (↑)—similarity between steps from *different* tasks vs. from the *same* task, with a value near 1.0 indicating a shared strategy space; (2) the *domain linear-probe accuracy* (↓)—how easily a logistic regression classifier can recover a step's task identity from its embedding; and (3) the *strategy cluster entropy* (↑)—the normalized task entropy per cluster after K-means, where high entropy means each cluster mixes many tasks.

Takeaway 1 — Transferable Reasoning Patterns

Across both behavioral and embedding measures, SimpleHRL's high-level reasoning steps are more similar across related OOD tasks than those of flat baselines, while the execution layer carries the task-specific content—a pattern consistent with SimpleHRL having discovered a more task-agnostic high-level reasoning-strategy space.

SimpleHRL's plan-only embeddings rank best on all three metrics among the methods evaluated (Table 2), consistent with its high-level reasoning steps inhabiting a more unified, shared space across tasks. The pairwise-task similarity heatmaps in Figure 2a make the same pattern visible: the plan-only panel is uniformly similar across tasks, while the plan+exec panel shows clear within-task blocks. Together with Table 2, this suggests a separation in which the planning layer is more task-agnostic

and the execution layer carries the task-specific content—a composition reminiscent of meta-learning approaches (Finn et al., 2017) that have long been argued to support transfer.



(a) Step-embedding similarity (SimpleHRL). (b) Reasoning-pattern transition similarity.

Figure 2: **Two views of cross-task transferability of high-level reasoning strategies.** *(a) Step-embedding view.* Pairwise cosine similarity between step embeddings across tasks: SimpleHRL's plan-only embeddings stay uniformly similar (mean 0.73); folding in execution drops the mean to 0.68 and surfaces clear within-task blocks—task-specificity lives in execution, not in plans. *(b) Reasoning-pattern view.* Pairwise cosine similarity between per-task transition matrices: SimpleHRL's patterns remain more similar across tasks (mean 0.54) than GRPO+CoT's (0.49), indicating transfer of the same high-level strategies between related tasks.

5.2 Lens 2: SimpleHRL Concentrates Training Signal on Pivotal Tokens

Hypothesis. A complementary question is *why RL training is more effective* under the hierarchical structure. We hypothesize that SimpleHRL's structural decomposition steers gradient updates toward *pivotal tokens*: the key decision points that most determine whether a reasoning trajectory succeeds or fails. If the training signal concentrates on pivotal tokens, each gradient step provides more informative learning signal, potentially leading to better credit assignment.

Defining pivotal tokens. Following Abdin et al. (2024), we define the *pivotality* of token x_t as the absolute change in expected success probability when generation continues from the context that includes the token versus the context that excludes it:

$$\mathrm{Pivot}(t) = \left| \mathbb{E}_{y \sim \pi(\cdot|x_{0:t})} \left[\mathbf{1}\{y \text{ is correct}\} \right] - \mathbb{E}_{y \sim \pi(\cdot|x_{0:t-1})} \left[\mathbf{1}\{y \text{ is correct}\} \right] \right|. \tag{3}$$

A large $\mathrm{Pivot}(t)$ indicates that token t constitutes a critical branching point in the solution path.

Experimental setup. We use gradient norm per token as a proxy for training signal intensity, and policy entropy per token as a proxy for the additional exploration bonus introduced by entropy-based algorithms. To isolate the effect of prompt format on gradient flow, we use the same base model checkpoint for both setups, generating responses under the SimpleHRL prompt versus the CoT prompt used by GRPO+CoT. We select the top-k ($k = 16$) tokens under each selection criterion and estimate $\mathrm{Pivot}(t)$ by sampling $m = 128$ continuations from both prefixes and computing empirical success-rate differences. We also include a random selection baseline.

Takeaway 2 — Implicit Credit Assignment

Under the SimpleHRL prompt, the top-k tokens by gradient norm have substantially higher pivotality than under the CoT prompt used by GRPO+CoT, suggesting that hierarchical structure concentrates the learning signal on more outcome-relevant decisions.

Results. Figure 3 highlights three findings. ❶ Under the SimpleHRL prompt, the top-k tokens by gradient norm have substantially higher average pivotality than under the CoT prompt used by GRPO+CoT. ❷ Gradient norm is a better predictor of pivotality than entropy for both prompts (both exceed random selection): under both SimpleHRL and GRPO+CoT, $\|\nabla\|$ selection outperforms entropy selection. This reveals a concrete mismatch: *increasing token-level uncertainty is not the same as concentrating signal on high-impact decisions*—a token can be high-entropy yet non-pivotal

(a stylistic word choice at a non-critical juncture), or low-entropy yet pivotal (a committed problem-framing step that is deterministic but decisive). ❸ The same underlying model exhibits different signal concentration under different output structures, supporting the hypothesis that training under the hierarchical structure may repeatedly reinforce more informative decision points over time.

6 Conclusion

We introduced **SimpleHRL**, a deliberately minimal, HRL-inspired structural prior for LLM RL post-training that imposes a hierarchical plan/execution structure on the reasoning trace, with no separate critic, no reward shaping, and no objectives beyond what flat baselines already use. Surprisingly, despite this minimality, across a diverse OOD benchmark and two model scales, the structural prior alone delivers consistent generalization gains over flat baselines. Two complementary, observational analyses suggest *why* this might happen: SimpleHRL's plan-level reasoning is markedly more task-agnostic than that of the flat baseline, consistent with discovering more transferable high-level reasoning strategies; and, under the hierarchical structure, training signal (gradient norm) is more concentrated on pivotal decision tokens, consistent with better implicit credit assignment.

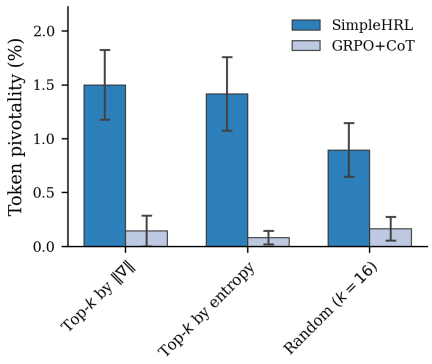


Figure 3: **Pivotality of training-signal-concentrated tokens.** Mean pivotality of the top-k (k=16) tokens selected by gradient norm, policy entropy, or random. Higher means the criterion concentrates on tokens that more strongly determine outcome quality.

Limitations. Three caveats constrain the scope of these conclusions. First, our credit-assignment analysis is observational: we measure that gradient norm concentrates on pivotal tokens under SimpleHRL, but we do not establish that this concentration *causes* the improved OOD generalization. Second, the reasoning-graph analysis relies on a frontier LLM as an annotator that we have not validated against human labels, so the behavioral evidence in Section 5.1 should be read as suggestive rather than definitive. Third, the extent to which the observed gains depend on the quality of the underlying base model is unclear. Our experiments use Qwen3-4B and Qwen3-8B, which were pre-trained with substantial chain-of-thought and structured reasoning data and therefore already possess some latent capacity for hierarchical reasoning. SimpleHRL may be eliciting and reinforcing a structure that pretraining has made accessible, rather than introducing it from scratch; in settings where the base model lacks this prior, the gains may differ. We note that this is consistent with broader recent findings that RL post-training largely elicits and amplifies capabilities already latent in the pre-trained model rather than introducing entirely new ones (Yue et al., 2025), and view it as a scope condition for our claims rather than a fundamental obstacle.

Broader implications and future directions. Beyond the specific SimpleHRL prompt, our results suggest that the granularity at which reasoning behavior is optimized may be a design axis for RL post-training comparable to the reward signal and the optimization algorithm. They also complement recent post-hoc analyses linking structural patterns in reasoning chains to generalization (Jiang et al., 2025; Li et al., 2026), by showing that such patterns can be *shaped* prospectively during RL training. Natural next steps include learned, task-adaptive structural priors beyond a fixed plan/execute schema and extending the same structure-versus-format distinction to settings such as tool use and agentic planning.

Ethics Statement

This work uses only publicly available benchmarks and open-weight models, and involves no human subjects and no personally identifiable or sensitive data. We are not aware of any ethical concerns arising from this work.

Reproducibility Statement

All experiments are repeated three times and we report the mean. Implementation details and the verbatim SimpleHRL prompt are given in Appendices D and E. We will release the code upon acceptance of the paper.

Use of Large Language Models

We used LLMs to polish the writing of this paper and to help implement parts of our codebase; all LLM-assisted code was reviewed and verified by the authors. In addition, as described in Section 5.1, a frontier LLM is used as an annotator in our post-hoc analysis pipeline, to segment reasoning traces and label reasoning steps; it is not part of the SimpleHRL method, the training loop, or the reward signal, and none of the benchmark results depend on it. We take responsibility for the final content of this work.

References

Marah Abdin et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025a.

Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Yonghao Zhuang, Nilabjo Dey, Yuheng Zha, Yi Gu, Kun Zhou, Yuqi Wang, Yuan Li, Richard Fan, Jianshu She, et al. Revisiting reinforcement learning for LLM reasoning from a cross-domain perspective. *arXiv preprint arXiv:2506.14965*, 2025b.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

Daya Guo et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Gangwei Jiang, Yahui Liu, Zhaoyi Li, Qi Wang, Fuzheng Zhang, Linqi Song, Ying Wei, and Defu Lian. What makes a good reasoning chain? uncovering structural patterns in long chain-of-thought reasoning. *arXiv preprint arXiv:2505.22148*, 2025.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35, 2022.

Zhaoyi Li, Xiangyu Xi, Zhengyu Chen, Wei Wang, Gangwei Jiang, Ranran Shen, Linqi Song, Ying Wei, and Defu Lian. On the role of reasoning patterns in the generalization discrepancy of long chain-of-thought supervised fine-tuning. *arXiv preprint arXiv:2604.01702*, 2026.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, 2018.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

OpenAI. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 2022.

Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems*, 10, 1998.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, 2025. doi: 10.1145/3689031.3696075.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pp. 3540–3549. PMLR, 2017.

Haozhe Wang, Qixin Xu, Che Liu, Junhong Wu, Fangzhen Lin, and Wenhu Chen. Emergent hierarchical reasoning in LLMs through reinforcement learning. *arXiv preprint arXiv:2509.03646*, 2025.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Dong, Yunxiao Yao, Yian Hu, Jikun Pan, Jiaming Liu, Muning Cao, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*, 2025.

A Option-Critic Mapping for SimpleHRL

SimpleHRL can be understood as a minimal instantiation of the option-critic framework (Bacon et al., 2017). We give a precise component-by-component mapping below.

Options ↔ plan-execute cycles. Each high-level plan together with its subsequent execution trace constitutes an *option*: a temporally extended action that the high-level policy selects and the low-level policy executes. The full response is a sequence of such options.

Manager (option selection) ↔ high-level plan content. In the option-critic, the manager selects which option to invoke at each decision point. In SimpleHRL, this role is played by the high-level plan content emitted at each step, which names the abstract strategy to pursue. There is no separate value function for option selection: GRPO's group-relative advantage serves as the credit signal for the entire response.

Worker (intra-option policy) ↔ execution content. The worker executes the selected option until termination. In SimpleHRL, the execution content following each plan statement is the intra-option behavior, implementing the abstract strategy named in the plan.

Termination function ↔ step boundary markers. The option-critic requires a learned termination function $\beta_\omega(s)$. In SimpleHRL, termination is implicit: the model places step boundary markers at natural reasoning boundaries, guided by the pre-trained LLM's prior on step structure from large-scale reasoning text. A learned termination function is therefore redundant.

Why this minimal instantiation is justified for our study. The motivation of our study is to reduce HRL to its minimal structural form—retaining only the structural prior on the reasoning trace and discarding all additional machinery (option-level critic, learned termination, manager value function, option-level loss)—and ask whether this minimal instantiation is effective in the LLM RL post-training regime and what mechanisms might explain any gains. We deliberately omit these components for two reasons. First, this isolates the structural inductive bias from confounds introduced by additional objectives or auxiliary losses, so any observed gain over flat baselines can be attributed to structure alone. Second, the pre-trained LLM already encodes a strong prior on natural step boundaries from large-scale reasoning text, which substitutes for the omitted termination mechanism, while GRPO's group-relative advantage provides credit signal at the response level without requiring a separate manager value function.

B Reasoning Pattern Analysis

We extract directed reasoning graphs from each response using Claude Sonnet 4.5 with the lcot2tree pipeline of Jiang et al. (2025), annotating each node with one of 10 semantic operation types. The main text reports global transition statistics in Section 5.1; here we provide the strategy-space coverage table and the per-task transition breakdown.

Strategy-space coverage (embedding geometry). Table 2 reports the three embedding-geometry metrics referenced in Section 5.1: SimpleHRL plan-only embeddings have the highest Cross/Within ratio, the lowest domain probe accuracy, and the highest cluster entropy across all methods, while plan + execution embeddings reverse all three metrics—confirming that the planning layer carries the transferable structure while the execution layer is task-specific.

Per-task reasoning patterns. Figure 4 shows the per-task transition probability difference $\Delta P = P_{\text{SimpleHRL}} - P_{\text{GRPO+CoT}}$ for each task in the OOD benchmark. The pattern is broadly consistent across tasks for SimpleHRL (its global transition signature reproduces in every task), whereas GRPO+CoT's

Table 2: Strategy-space coverage analysis via step-embedding geometry. Arrows indicate the direction of broad, task-agnostic coverage. SimpleHRL plan-only steps have the most task-agnostic embeddings across all metrics. Appending execution to the plan collapses this agnosticism.

Method	Cross/Within Ratio (↑)	Domain Probe Acc. (↓)	Cluster Entropy (↑)
SimpleHRL (plan-only)	**0.976**	**0.571**	**0.659**
SimpleHRL (plan+exec)	0.927	0.795	0.341
Entropy Adv.	0.958	0.665	0.533
GRPO+CoT	0.963	0.667	0.492

matrix appears to rely more heavily on a near-linear *setup→computation→conclusion* pattern on tabular OOD tasks (FinQA, HiTab, MultiHier), with much sparser use of *verification* and *correction* transitions there. Behaviorally, GRPO+CoT shows reduced use of these meta-cognitive transitions once it leaves its training distribution; SimpleHRL does not. This is the behavioral counterpart of the task-invariance observed in plan-only embeddings (Section 5.1).

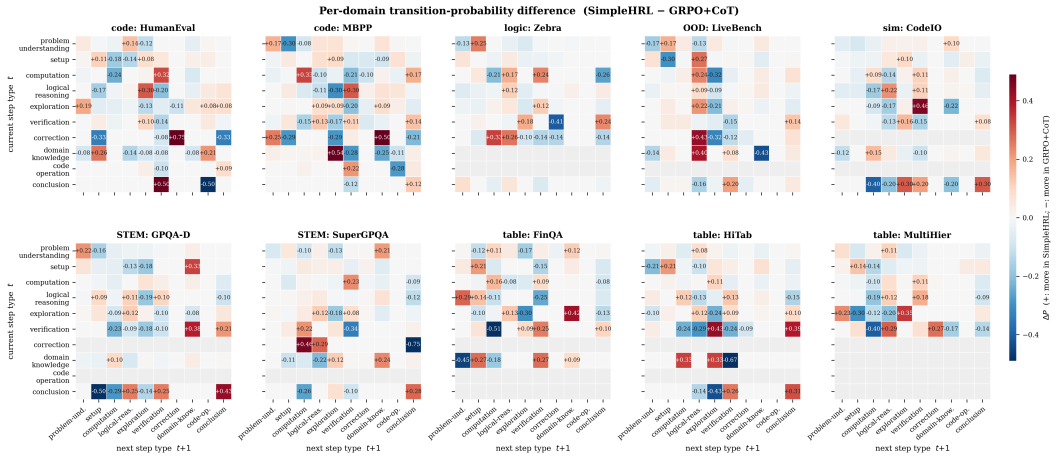


Figure 4: **Per-task transition-probability difference $P_{\text{SimpleHRL}} - P_{\text{GRPO+CoT}}$ across the OOD benchmark.** Sparse rows (gray) indicate fewer than 2 transitions of that current-type in either method, and are masked. SimpleHRL uses its full taxonomy of meta-cognitive transitions across domains; GRPO+CoT shows much sparser use of *verification/correction* transitions on the tabular OOD tasks.

C Response-Length Analysis

The three methods compared in Figure 6 are GRPO+CoT (the flat RL baseline), GRPO+long-CoT (the same checkpoint, evaluated with an inference-time prompt asking for longer, more deliberate reasoning), and SimpleHRL. The long-CoT prompt was instantiated as the verbatim suffix *"Take your time and think through this carefully and thoroughly, step by step. Consider multiple angles before committing to an answer."*; we did not retrain or otherwise modify the model.

The long-CoT prompt produces longer responses on most tasks, with the largest length increases on tasks where GRPO+CoT was previously the most concise (e.g., MBPP: $256 \rightarrow 1{,}214$ tokens; FinQA: $573 \rightarrow 897$; HiTab: $620 \rightarrow 1{,}307$). Within each method, the per-response point-biserial correlation between length and correctness is also mildly negative on average

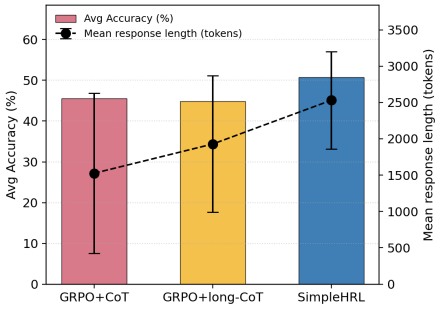


Figure 6: **Length increases, but accuracy does not track length** (cross-task averages over the OOD benchmark). The long-CoT prompt lengthens responses by $+26\%$ over GRPO+CoT but its accuracy is essentially flat; SimpleHRL has the highest average response length and the highest average accuracy among the three methods.

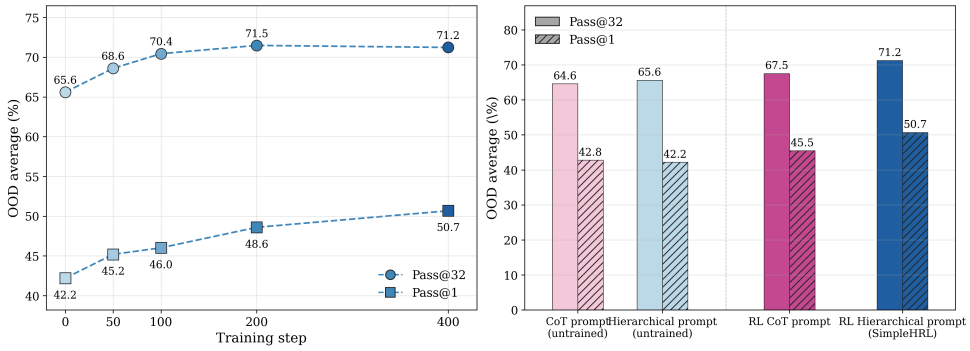


Figure 5: **Training dynamics on Qwen3-4B.** *Left:* SimpleHRL OOD accuracy improves with training. *Right:* on the same untrained base model the CoT and hierarchical prompts give comparable accuracy; the generalization gap opens up after RL training under the hierarchical structure.

($\bar{r}_{\text{GRPO+CoT}} = -0.19$, $\bar{r}_{\text{long-CoT}} = -0.18$, $\bar{r}_{\text{SimpleHRL}} = -0.16$), so within a fixed method the longer rollouts are not the more accurate ones either.

D Implementation Details

All methods share the same base model at each scale (Qwen3-4B and Qwen3-8B (Yang et al., 2025)) and a common set of hyperparameters: learning rate 1×10^{-6}, training batch size 128, maximum prompt length 1,792 tokens, maximum response length 8,192 tokens, 8 rollout samples per prompt per step, trained for 2 epochs. All experiments are conducted on a single node of 8 NVIDIA H200 GPUs using the VeRL framework (Sheng et al., 2025).

E SimpleHRL Prompt and Reasoning-Strategy Taxonomy

SimpleHRL prompt. The SimpleHRL prompt is prepended to the user query and reproduced verbatim below. Both baselines and SimpleHRL share the same RL setup otherwise.

SimpleHRL prefix prepended to every user query

```
Please solve the following problem step by step. Before each
reasoning step, write a brief high-level plan inside <step> and
</step> tags, then carry out the computation OUTSIDE the tags.
The <step> tags should ONLY contain a short description of what
you will do, NOT the actual work.

Example format:
<step>Set up the equation from the given conditions</step>
Let x be the unknown. From the problem, we know that 2x + 3 = 11...
<step>Solve for x by isolating the variable</step>
Subtracting 3 from both sides: 2x = 8. Dividing by 2: x = 4...

Now solve this problem:

{question}
```

Reasoning-strategy taxonomy. For the analyses in Section 5.1, every reasoning step extracted from a rollout (by Claude Sonnet 4.5) is classified into exactly one of the 10 high-level reasoning-strategy types in Table 3. The same taxonomy is applied identically to all methods.

Table 3: The 10 reasoning-strategy types used to label each segmented reasoning step. The taxonomy is designed to span math, code, logic, science, and tabular tasks while distinguishing meaningfully different cognitive operations.

Type	Covers
problem_understanding	Reading/restating the problem, extracting constraints, identifying goals
setup	Defining variables, formulating equations or expressions, stating hypotheses
computation	Algebraic manipulation, arithmetic, substitution, estimation
logical_reasoning	Logical deduction, case analysis, proof by contradiction, induction
exploration	Trying alternatives, branching, comparing approaches, elimination
verification	Checking or validating results, sanity checks, error detection
correction	Backtracking, self-correction, abandoning dead ends, debugging
domain_knowledge	Invoking theorems, definitions, or known facts; pattern recognition; analogy
code_operation	Writing or running code, simulation, building tables, enumeration
conclusion	Stating final or intermediate answer, combining partial results, summarizing