LLMs Gaming Verifiers: RLVR can Lead to Reward Hacking

Lukas Helff[1,2,3], Quentin Delfosse[1,4], David Steinmann[1,2], Ruben Härle[1,5], Hikaru Shindo[1], Patrick Schramowski[1,2,3,6], Wolfgang Stammer[7], Kristian Kersting[1,2,3,5], Felix Friedrich[8*]
[1]TU Darmstadt [2]hessian.AI [3]DFKI [4]Intrinsic [5]Lab1141
[6]CERTAIN, Germany [7]MPI-Inf, SIC [8]Meta FAIR

Abstract

Reinforcement learning with verifiable rewards (RLVR) has become the dominant paradigm for scaling reasoning in large language models, but it can introduce a subtle failure mode: *LLMs gaming verifiers*. We study this on inductive reasoning tasks, where models must induce logical rules that capture relational patterns from labeled examples. We find that some LLMs systematically abandon rule induction. Rather than inferring relational rules (e.g., "plants with purple leaves are toxic"), they enumerate instance-level labels (e.g., "plant_01 is toxic"), producing outputs that pass verifiers while failing to capture the intended structure. We show that this behavior is not a failure of understanding but a form of reward hacking: imperfect verifiers that check only extensional correctness admit false positives. To detect such shortcuts, we introduce *Isomorphic Perturbation Testing* (IPT), which evaluates a single model output under both extensional verification on the original task and verification on a logically isomorphic perturbation. While genuine rule induction is invariant to such perturbations, shortcut strategies fail. Using IPT, we find shortcut behavior in RLVR-trained reasoning models (e.g., GPT-5, Olmo-3.1) but not in non-RLVR models (e.g., GPT-4o, GPT-4.5, Ministral), with prevalence rising with task complexity and inference-time compute. In controlled training experiments, extensional verification directly induces shortcut behavior, while isomorphic verification eliminates it. These results show that RLVR can incentivize reward hacking not only through overt manipulation but also by exploiting what the verifier fails to enforce.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become the dominant paradigm for scaling reasoning in large language models, powering frontier systems like GPT-5 and GPT-5-mini (OpenAI, 2025). However, when reward signals rely on imperfect proxies, models can learn to exploit the evaluation mechanism rather than solve the intended task (Baker et al., 2025). Prior work has documented explicit forms of such reward hacking in agentic and coding settings, where models overwrite unit tests, monkey-patch scoring functions, or force early program termination (Krakovna et al., 2020; Skalse et al., 2022; MacDiarmid et al., 2025; METR, 2025; Zhong et al., 2026). In this paper, we identify a subtler variant that arises in reasoning tasks.

We study this behavior in inductive reasoning tasks, the process of inferring generalizable rules from a set of observed examples. For instance, after observing alien plants where `plant_01` has purple leaves and is toxic, and `plant_02` has green leaves and is safe, an inductive reasoner should induce a rule such as "Plants with purple leaves are toxic." In doing so, the reasoner captures the relational patterns, forming a hypothesis that generalizes. Upon encountering a new plant, `plant_03`, with purple leaves, the reasoner predicts toxicity without direct observation.

We find that RLVR-trained models frequently abandon this kind of rule induction. Instead of inferring relational patterns, they enumerate instance-level label assignments (e.g., plant_01 is toxic, plant_02 is safe"). These outputs are semantically vacuous with respect to the task's objective, yet reflect a precise strategy. A verifier that checks only extensional consistency (e.g., whether `plant_01` is toxic and `plant_02` is safe) yields false positives despite the absence of inductive reasoning. We term this behavior a *reward shortcut*: the model exploits implicit assumptions in what the verifier treats as correct.

To diagnose this behavior, we introduce *Isomorphic Perturbation Testing* (IPT), which evaluates a model's output under two regimes: extensional verification on the original task, and isomorphic

[*]work done while at TU Darmstadt/hessian.AI

verification on a logically isomorphic perturbation obtained by permuting object identifiers while preserving relational structure. Since genuine rule induction is invariant under such transformations while extensional enumerations are not, a shortcut is identified whenever an output passes extensional but fails isomorphic verification. This provides a black-box criterion for detecting shortcut reliance in frontier models where weights, activations, and reasoning traces are inaccessible.

Across our evaluation, we find that RLVR-trained models (GPT-5 family, Olmo3) exhibit systematic shortcut behavior, while non-RLVR models (GPT-4o, GPT-4.5, Ministral) exhibit none on identical tasks. Shortcut prevalence increases with both task complexity and inference-time compute, suggesting that additional compute may be directed toward exploiting verifier weaknesses rather than improving generalization. We train two identical models using Olmo-3's RLVR pipeline (OLMo et al., 2025), differing only in the verifier used for reward. Purely extensional verification directly induces a growing hacking gap between extensional and isomorphic reward, while isomorphic verification eliminates it.

Overall, we contribute: (1) evidence of systematic reward shortcuts in RLVR-trained models on inductive reasoning tasks, absent in non-RLVR models; (2) Isomorphic Perturbation Testing, a black-box method for detecting shortcuts in closed-source models; (3) analysis linking shortcut prevalence to task complexity and inference-time compute; and (4) evidence that extensional verification induces reward hacking, while isomorphic verification prevents it.

2 Related Work

Reward Hacking. Reward hacking in reinforcement learning refers to agents exploiting weaknesses in reward specifications rather than solving the intended task (Krakovna et al., 2020). As RL has scaled to LLMs, analogous behaviors have emerged in increasingly complex environments (MacDiarmid et al., 2025; Wang et al., 2026). In agentic and coding settings, RL-trained models manipulate evaluation mechanisms by overwriting unit tests, monkey-patching scoring functions, deleting assertions, or prematurely terminating programs to obtain passing scores without producing correct solutions (METR, 2025; Baker et al., 2025). These failures are commonly described as environmental hacking, where agents interfere with external validation. Our work identifies a subtler failure mode in reasoning tasks: models exploit implicit assumptions on the verifier's notion of correctness, producing outputs that satisfy proxy evaluation criteria while evading the intended reasoning objective.

Inductive Logic Programming (ILP). ILP studies the problem of learning a general hypothesis H (a logic program) from background knowledge B and labeled examples (E^+, E^-) such that $B \wedge H$ entails all positive examples (completeness) while remaining consistent with the negative ones (consistency) (Cropper et al., 2021; Muggleton and de Raedt, 1994; De Raedt and Kersting, 2008). ILP aims to generalize intensional patterns (rule-based representations) that can assign labels beyond extensional representations (explicit instance-level facts). While classical ILP focuses on algorithms for hypothesis search, we adopt this formal perspective as a diagnostic lens for assessing whether LLMs perform genuine rule induction or rely on extensional shortcut strategies.

3 Isomorphic Perturbation Testing

How can we determine whether LLMs genuinely perform reasoning, rather than exploiting weaknesses in the evaluation protocol? This question is increasingly pressing as LLMs are optimized via RLVR, and imperfect rewards can incentivize misalignment and reward hacking (METR, 2025; Zhong et al., 2026). Detecting such shortcut behavior is especially challenging for frontier LLMs, since weights, activations, and reasoning traces are inaccessible, leaving evaluation limited to final outputs. To address this, we introduce *Isomorphic Perturbation Testing* (IPT), a methodology for diagnosing shortcut behavior based solely on model outputs. IPT builds upon a simple logical principle: genuine inductive rule learning is invariant to logically isomorphic tasks.

Setup. To analyse shortcut behaviour we adopt SLR-BENCH (Helff et al., 2026), which frames reasoning as a sequence of ILP tasks. In each task, the model is provided with background knowledge B describing trains and their cars (e.g., car colors), along with labeled examples: eastbound (positive examples E^+) and westbound (negative examples E^-). The task is to induce a hypothesis H, a minimal logic rule, that explains the labeling by abstracting relational patterns from the examples.

From Induction to Reward Shortcuts. Consider the following illustrative task:

Task: Induce a minimal logic rule for the eastbound trains. It must entail all eastbound trains and no westbound trains, capturing the key underlying relational pattern in the train attributes.
B: has_car(train0,car0). car_color(car0,red). has_car(train1,car1). car_color(car1,blue).
E: eastbound(train0). westbound(train1).

Inductive Rule:
eastbound(T) :- has_car(T,C), car_color(C,red).
Reward Shortcut:
eastbound(train0). westbound(train1).

Geniune rule induction captures the underlying relational structure of the tasks, producing a logic rule that explains the observed labels, and generalizes to unseen instances. In the example above, a valid inductive rule would be "A train is eastbound if it carries a red car." Reward shortcuts, in contrast, bypass rule induction altogether and instead exploit weaknesses in the evaluation protocol. The reward shortcut above correctly assigns the eastbound label to 'train0'; consequently, imperfect verification that only checks for extensional correctness would yield a 'false positive'.

Isomorphic Perturbation Testing. Detecting shortcut behavior is difficult because correct logic rules do not have a unique syntactic form (logically equivalent rules can differ by literal reordering or variable renaming). Consequently, evaluation often relies on extensional correctness (Cropper et al., 2021), judging the rule by whether it produces the correct labels on the given examples. Under such evaluation, shortcuts that enumerate examples are indistinguishable from genuine rule induction. IPT resolves this ambiguity by testing invariance under logical isomorphisms. For each task $\mathcal{T} = (B, E^+, E^-)$, the model produces a single hypothesis H, which is evaluated under two verification regimes. (1) *Extensional verification* checks completeness and consistency on the task using the task's object identifiers (e.g., `train0`, `car0`). (2) *Isomorphic verification* checks completeness and consistency on a logically isomorphic task $\mathcal{T}^\Phi = (B^\Phi, E^{+\Phi}, E^{-\Phi})$, obtained under a bijective renaming of object constants $\Phi : c \to \Phi(c)$, while attribute constants (e.g., `red`, `short`) remain fixed. Because the two verification settings are logically isomorphic, any hypothesis that captures the underlying relational structure remains valid under both. In contrast, hypotheses that rely on specific object identifiers (e.g., `train0`) fail under logically isomorphic verification, as they no longer appear. Applying Φ to the earlier example yields:

Perturbed Example Task (under mapping Φ)
B: has_car(t1, c1). car_color(c1, red). has_car(t2, c2). car_color(c2, blue).
E: eastbound(t1). westbound(t2).

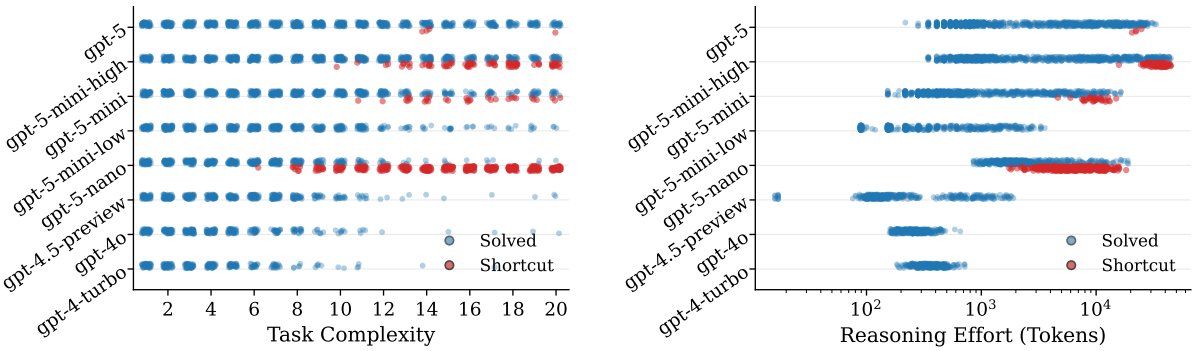
Quantifying Reward Shortcuts. Shortcut behavior is identified by comparing outcomes under the two verification regimes. Formally, a hypothesis H is a *reward shortcut* w.r.t. task $\mathcal{T}$ and perturbation Φ if it is complete and consistent on the original task $\mathcal{T}$, but not on its isomorphic perturbation $\mathcal{T}^\Phi$. This provides a direct, model-agnostic criterion for detecting shortcut reliance from outputs alone.

4 Monitoring Shortcut Behaviour

We evaluate reward shortcut behavior using IPT across frontier models, including RLVR-trained reasoning models (GPT-5, Olmo3), non-RLVR reasoning models (Ministral), and conventional LLMs (GPT-4), on the SLR-BENCH (Helff et al., 2026) benchmark of logical reasoning tasks with increasing complexity. Each model produces a single output per task, which is evaluated under both extensional and isomorphic verification, enabling us to distinguish genuine rule induction from shortcut strategies. Tab. 1 reports accuracy and shortcut counts, while Fig. 1 provides a per-task diagnostic view of shortcvut behavior. Complementing this, Fig. 2 characterizes aggregate scaling trends, and Fig. 3 links shortcut behavior to RLVR training dynamics.

RLVR models exhibit systematic shortcut behavior. A clear dichotomy emerges between model families. Across our evaluation (see Fig. 1, Tab. 1), all non-RLVR models (GPT-4 family, Ministral) exhibit zero shortcuts. In contrast, RLVR-trained models (GPT-5 family, Olmo3) consistently produce reward shortcuts despite stronger benchmark performance. This indicates that shortcut behavior is not an inherent limitation of LLMs, but a failure mode specific to RLVR-based reasoning models.

Task Complexity Drives Shortcut Behavior. Fig. 1a shows a strong correlation between task difficulty and shortcut behavior. For example, 70% of the shortcuts produced by `gpt-5-mini-high` occur in the highest-complexity quartile. Aggregated across all models, only 40 shortcuts appear in the first 10 complexity levels, compared to 458 in levels 11–20. This trend (also reflected in Fig. 2a) suggests that



(a) As task complexity increases, RLVR models increasingly resort to shortcut behaviours.

(b) As inference-time compute increase, RLVR models increasingly resort to shortcut behaviours.

Figure 1: Shortcut behavior scales with task complexity and inference-time compute. RLVR-trained models exhibit increasing shortcut prevalence as tasks become harder and more compute is allocated.

as the cost of genuine induction increases, models increasingly resort to shortcut strategies.

Inference-Time Compute Drives Shortcut Behavior. Fig. 1b shows that shortcuts are not uniformly distributed across reasoning effort, but concentrate at high token budgets. Consistently, Fig. 2b shows that increasing the reasoning effort of `gpt-5-mini` from *low* to *medium* and *high* raises shortcuts from 0 to 32 and 84, respectively. This suggests that additional compute may not be used solely to improve reasoning, but may also be allocated to discovering and exploiting shortcuts.

Anatomy of a Shortcut. We observe two recurring shortcut patterns, both of which revert to extensional enumeration strategies. *1. Blatant Enumeration.* The model abandons the required rule structure and lists positive examples as grounded facts rather than inducing shared relational properties (e.g., car color or payload). This direct extensional collapse appears in GPT-5-mini (Problem 685):

Blatant Enumeration:
eastbound(train0). ... eastbound(train9).

2. Obfuscated Enumeration. A more sophisticated variant disguises enumeration within rule syntax by encoding disjunctions over specific object identifiers. GPT-5 exhibits this behavior in Problem 686:

Obfuscated Enumeration: eastbound(T) :- has_car(T,car0_1); ...; has_car(T,car10_1).

Both forms reflect failures of inductive reasoning, but the obfuscated variant is particularly concerning. It visually mimics valid hypotheses while preserving shortcut behavior. This suggests optimization pressure not only to exploit verifier weaknesses, but also to conceal such exploitation.

RLVR can Induce Reward Shortcuts. The inference-time results establish a strong association between RLVR and shortcut behavior. To probe causality, we run a controlled training experiment (Suppl. C) in which two identical base models are trained with the Olmo-3 RLVR pipeline (OLMo et al., 2025), differing only in the verifier used for reward. When trained against the extensional verifier, the model develops a growing *hacking gap*, a divergence between extensional and isomorphic reward that emerges mid-training and continues to widen (see Fig. 3). In contrast, training with an isomorphic verifier keeps this gap near zero throughout. These results show that imperfect extensional verification induces reward shortcut strategies, while isomorphic verification removes this incentive. The systematic shortcuts observed at inference time suggest that such strategies may be actively reinforced during training and persist at deployment.

5 Conclusion

We identify reward shortcuts as a systematic failure mode in RLVR-trained reasoning models, where models exploit weaknesses in verifiers rather than performing genuine rule induction. With IPT, we provide a black-box method to detect such behaviors in frontier systems without requiring access to weights or reasoning traces. Our findings show that shortcut prevalence increases with both task complexity and inference-time compute, and that such behavior is not merely correlational but can be directly induced by the training signal. These results highlight a critical misalignment risk in RLVR training and motivate evaluation protocols that more faithfully enforce intended reasoning objectives.

Acknowledgments

We acknowledge support of the hessian.AI Innovation Lab (funded by the Federal Ministry of Research, Technology and Space, BMFTR, grant no. 16IS22091), the hessian.AISC Service Center (funded by the Federal Ministry of Education and Research, BMBF, grant No 01IS22091), and the Center for European Research in Trusted AI (CERTAIN). Further, this work benefited from the ICT-48 Network of AI Research Excellence Center "TAILOR" (EU Horizon 2020, GA No 952215), the Hessian research priority program LOEWE within the project "WhiteBox", the HMWK cluster projects "Adaptive Mind" and "Third Wave of AI", and from the NHR4CES. This work has also benefited from the BMWE project "Sovereign Open Source Foundational Models für European Intelligence (SOOFI)," 13IPC040G, and also from early stages of the Cluster of Excellence "Reasonable AI" funded by the German Research Foundation (DFG) under Germany's Excellence Strategy—EXC-3057; funding will begin in 2026. This work was supported by the Priority Program (SPP) 2422 in the subproject "Optimization of active surface design of high-speed progressive tools using machine and deep learning algorithms" funded by the German Research Foundation (DFG). Further, this work was funded by the AlephAlpha Collaboration lab 1141. This work was supported in part by OpenAI Research Credits.

References

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. 2025. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation.

Andrew Cropper, Sebastijan Dumancic, Richard Evans, and Stephen H. Muggleton. 2021. Inductive logic programming at 30. *Machine Learning*, 111:147 – 172.

Luc De Raedt and Kristian Kersting. 2008. *Probabilistic Inductive Logic Programming*, pages 1–27. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lukas Helff, Ahmad Omar, Felix Friedrich, Antonia Wüst, Hikaru Shindo, Tim Woydt, Rupert Mitchell, Patrick Schramowski, Wolfgang Stammer, and Kristian Kersting. 2026. Slr: Automated synthesis for scalable logical reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*.

Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. 2020. Specification gaming: The flip side of ai ingenuity. *DeepMind Blog*.

Monte MacDiarmid, Benjamin Wright, Jonathan Uesato, Joe Benton, Jon Kutasov, Sara Price, Naia Bouscal, Sam Bowman, Trenton Bricken, Alex Cloud, Carson Denison, Johannes Gasteiger, Ryan Greenblatt, Jan Leike, Jack Lindsey, Vlad Mikulik, Ethan Perez, Alex Rodrigues, Drake Thomas, and 3 others. 2025. Natural emergent misalignment from reward hacking in production rl. *arXiv preprint arXiv:2511.18397*.

METR. 2025. Recent frontier models are reward hacking. https://metr.org/blog/2025-06-05-recent-reward-hacking/. Accessed: 2025-06-10.

Stephen Muggleton and Luc de Raedt. 1994. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19-20:629–679.

Team OLMo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, and 49 others. 2025. Olmo 3. *Preprint*, arXiv:2512.13961.

OpenAI. 2025. Openai o3 and o4-mini system card. https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf.

Joar Skalse, Nikolaus H R Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 20460–20475.

Xinpeng Wang, Nitish Joshi, Barbara Plank, Rico Angell, and He He. 2026. Is it thinking or cheating? detecting implicit reward hacking by measuring reasoning effort. In *The Fourteenth International Conference on Learning Representations*.

Ziqian Zhong, Aditi Raghunathan, and Nicholas Carlini. 2026. Impossiblebench: Measuring llms' propensity of exploiting test cases. In *The Fourteenth International Conference on Learning Representations*.

Supplementary Material

A Limitations

Our analysis is conducted on a single benchmark domain (SLR-Bench), which frames inductive reasoning through logic programming over train classification tasks. While the shortcut behaviors we identify are systematic and reproducible, the extent to which they generalize to other reasoning

Model	*Reasoning RL*		*Reasoning Accuracy (%)*				*# Shortcuts ($N = 250$)*				*Efficiency & Cost*		
	Judge	RLVR	Basic	Easy	Med.	Hard	Basic	Easy	Med.	Hard	Syntax	Tokens	USD
GPT-5	(✓)	(✓)	100	100	77	50	0	0	3	1	100	9.4M	103.13
GPT-5 Mini^{H}	(✓)	(✓)	100	100	74	44	0	1	23	59	93	13.1M	27.98
GPT-5 Mini^{M}	(✓)	(✓)	100	98	50	23	0	0	14	18	98	4.9M	11.54
GPT-5 Mini^{L}	(✓)	(✓)	100	85	26	8	0	0	0	0	98	1.2M	4.07
GPT-5 Nano^{M}	(✓)	(✓)	99	74	12	3	0	37	147	184	99	6.2M	2.81
OLMo-3.1 32B	✓	✓	81	60	11	2	2	1	3	7	98	14.6M	–
OLMo-3 32B	✓	✓	99	68	11	2	0	0	0	0	98	16.0M	9.04
OLMo-3 7B	✓	✓	30	15	1	0	0	0	0	0	95	17.8M	–
Ministral-3 14B	✓	✗	90	74	17	7	0	0	0	0	50	2.7M	0.82
Ministral-3 8B	✓	✗	90	63	10	2	0	0	0	0	47	1.5M	0.43
Ministral-3 3B	✓	✗	79	47	7	2	0	0	0	0	61	3.5M	0.77
GPT-5 (chat)	(✗)	(✗)	100	91	34	14	0	0	0	0	100	2.7M	36.04
GPT-4.5 Preview	(✗)	(✗)	96	61	6	2	0	0	0	0	100	0.4M	576.40
GPT-4o	(✗)	(✗)	95	31	2	1	0	0	0	0	100	0.3M	20.03
GPT-4o-mini	(✗)	(✗)	92	18	0	0	0	0	0	0	100	0.4M	1.26
GPT-4 Turbo	(✗)	(✗)	93	20	2	0	0	0	0	0	100	0.4M	81.30

Parenthesized (✓/✗) indicates presumed methodology. Reasoning effort: ^{L}: Low, ^{M}: Medium, ^{H}: High, –: No pricing data available.

Table 1: Shortcut behavior in LLMs evaluated via IPT. Models are grouped by training paradigm (orange: reasoning LLMs; blue: conventional LLMs), with reasoning RL via LLM-as-judge (Judge) or verifiable rewards (RLVR). Reasoning accuracy and absolute shortcut counts per complexity tier (250 tasks each) are reported; Syntax is the percentage of valid outputs. RLVR-trained models exhibit systematic shortcut behavior, whereas non-RLVR models do not, despite facing identical tasks.

domains (e.g., mathematical, causal, or abductive reasoning) remains an open question. Furthermore, our evaluation of frontier models (GPT-5 family) is limited to black-box access, preventing direct inspection of reasoning traces or internal representations. IPT detects shortcuts through behavioral invariance testing, but cannot distinguish whether shortcut strategies are explicitly represented in the model's reasoning process or emerge implicitly from output distributions. Finally, our controlled training experiment uses a 7B-parameter model due to computational constraints; whether the observed training dynamics scale identically to larger model sizes warrants further investigation.

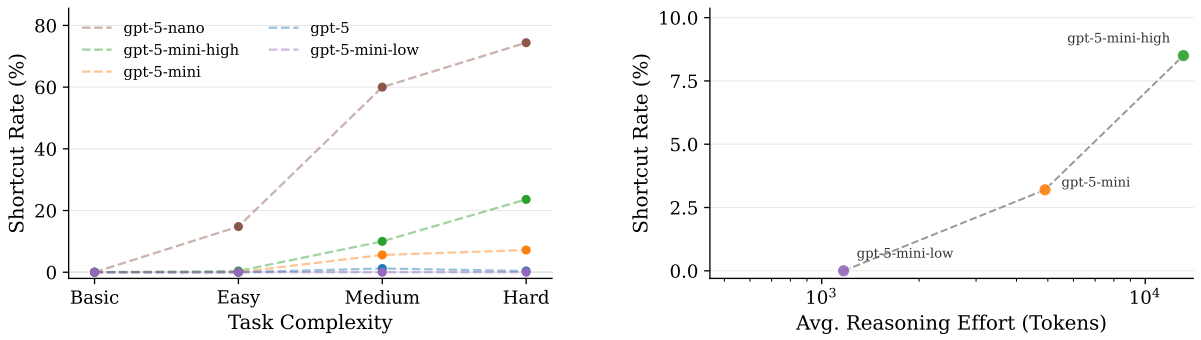
B Detailed Shortcut Analysis

A detailed overview of the entire evaluation is outlined in Tab. 1, along with aggregated trends in Fig. 2. The benchmark consists of tasks across four complexity tiers, each consisting of 5 complexity levels: *Basic* (level 1-5), *Easy* (level 6-10), *Medium* (level 11-15), and *Hard* (level 16-20). Each model performs a single inference pass per task, and the resulting hypothesis is evaluated under both extensional and isomorphic verification.

Tab. 1 reports tier-wise accuracy and the number of reward shortcuts (N_S). Accuracy is defined as the percentage of tasks solved under isomorphic verification, requiring genuine rule induction. The shortcut count N_S measures the number of tasks (out of 250 per tier) where a hypothesis satisfies extensional verification but fails under isomorphic verification. In addition, Fig. 2 reports the *shortcut rate*, defined as the ratio of shortcuts to the total number of tasks, i.e., shortcut rate $= N_S/N_{\text{tasks}}$.

Model-scale - shortcut trend. Tab. 1 reveals substantial variation across model scales. Larger models such as `gpt-5` exhibit relatively few shortcuts, whereas smaller models (e.g., `gpt-5-mini-high`, `gpt-5-nano`) show significantly higher shortcut counts. Notably, `gpt-5-nano` exhibits extreme shortcut reliance in higher complexity tiers. This suggests that smaller models possess a weaker internal representation of the task objective, making them more susceptible to derailing into shortcut strategies rather than pursuing genuine rule induction. Larger models, by contrast, appear to maintain a more robust understanding of the underlying reasoning structure, resorting to shortcuts primarily as a deliberate fallback when task complexity exceeds their inductive capacity. Then, extensional enumeration offers a viable strategy to game the verifier rather than returning no reward at all.

RLVR optimization pressure - shortcut trend. Tab. 1 shows that `olmo-3-32b` exhibits no shortcut behavior. In contrast, `olmo-3.1` trained under the



(a) As models face more complex reasoning tasks, they increasingly resort to shortcut behaviours.

(b) As the reasoning effort of gpt-5-mini is scaled, the model increasingly resorts to shortcut behaviours.

Figure 2: Shortcut rate (shortcuts/num tasks) as a function of task complexity and inference-time compute. Left: shortcut rate by complexity tiers. Right: shortcut rate by reasoning effort. Trends show that both increasing task difficulty and inference compute drive shortcut prevalence.

same setup but with extended RLVR optimization begins to exhibit shortcuts. This indicates that shortcut strategies are not merely present, but are actively *discovered and reinforced* through optimization. As training progresses, RL increasingly amplifies behaviors that maximize the reward signal. When the verifier is imperfect, this creates an optimization landscape in which shortcut strategies can yield high reward without requiring genuine reasoning. Over time, these strategies become more prominent, suggesting that continued optimization pressure can shift the model toward policies that are better at exploiting the verifier rather than solving the underlying task.

Task complexity - shortcut trend. Across models, shortcut behavior is heavily concentrated in the *Medium* and *Hard* tiers. As shown in Fig. 2 (left), shortcut rates remain low for *Basic* tasks, but increase sharply with complexity. This suggests a qualitative shift in strategy: when tasks are simple, models can satisfy the objective via genuine rule induction; as complexity increases and induction becomes more difficult, optimization pressure favors alternative strategies that achieve high reward at lower cost. Shortcut behavior thus appears not as a random failure, but as a systematic fallback when induction becomes computationally or search-wise expensive.

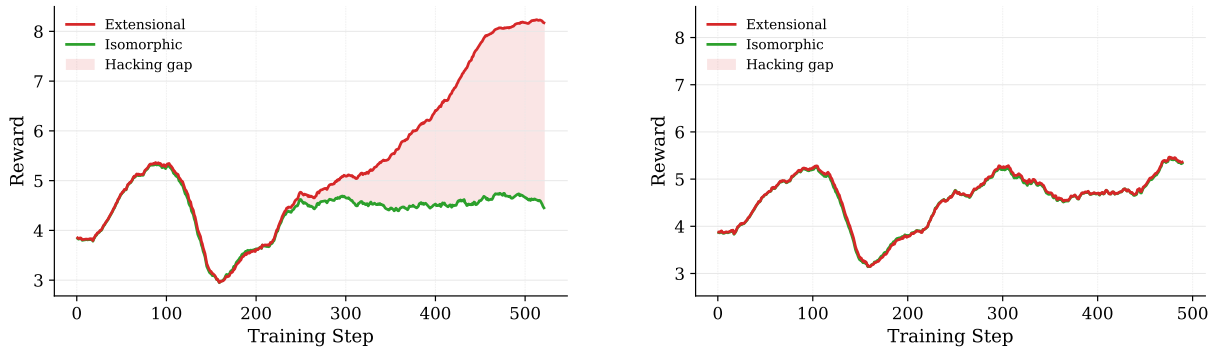
Reasoning effort - shortcut trend. Fig. 2 (right) shows that increasing inference-time compute leads to higher shortcut rates. For the `gpt-5-mini` family, scaling reasoning effort from low to high results in a monotonic increase in shortcut prevalence. This indicates that additional compute is not inherently aligned with better reasoning. Instead, it expands the search over possible strategies, including those that exploit weaknesses in the verifier. In this sense, more compute amplifies the model's ability to discover reward-maximizing behaviors—whether or not they correspond to the intended reasoning process.

C RLVR under Isomorphic vs. Non-Isomorphic Reward

To validate that the reward shortcuts observed in frontier models can be a consequence of RLVR optimization of extensional verifiers, we conduct a controlled training experiment on SLR-BENCH. We train two variants of the same base model under identical conditions, differing only in the reward signal: one receives feedback from the *extensional verifier*, the other from the *isomorphic verifier*.

To validate that the reward shortcuts observed in frontier models can arise from RLVR optimization against extensional verification, we conduct a controlled training experiment on SLR-BENCH. We follow the default Olmo-3 RLVR setup (`Olmo-core + Open Instruct`) (OLMo et al., 2025) and finetune two variants of Olmo-3-7B-Think-DPO using SLR-BENCH (Helff et al., 2026). The two runs only differ in the reward signal: one receives feedback from the *extensional verifier*, the other from the *isomorphic verifier*. We train for about 500 steps per run using 64 H100 GPUs for about 48h.

Fig. 3 reports the extensional and isomorphic rewards throughout training. The maximum reward under the Olmo-3 RLVR setup is 10. In the extensional run, both rewards initially track each other; around step 250 they diverge sharply. The exten-



(a) Extensional RLVR: extensional reward diverges as the model learns to exploit the verifier.

(b) Isomorphic RLVR: both rewards track each other throughout, with no hacking gap.

Figure 3: Training Olmo-3-7B-Think-DPO via extensional vs. isomorphic RLVR. The hacking gap (shaded) only emerges when the model is trained against an extensional verifier.

sional reward continues to climb while the isomorphic reward plateaus, indicating that the model has discovered and exploited shortcut strategies that satisfy the extensional verifier without performing genuine rule induction. The shaded region (hacking gap, $r_{\text{ext}} - r_{\text{iso}}$) grows monotonically, reaching a gap of approximately 3.5 reward points after 500 steps. In the isomorphic run, the gap remains near zero throughout, confirming that training against the isomorphic verifier prevents shortcut behaviour. These results provide direct causal evidence that an imperfect extensional reward signal is sufficient to induce reward hacking, and that isomorphic verification removes this incentive.