

Observe but Don’t Optimize: Structural Rewards Trigger Goodhart Effects in Neurosymbolic Reasoning

Apex Intelligence

Abstract

Neurosymbolic reasoning pipelines evaluate formalizations by execution alone, treating all successful proofs as equal. This single-criterion reward conflates genuine derivation with structural shortcuts: embedding the conclusion among the premises proves just as reliably, and reinforcement learning (RL) optimizers exploit the indifference. We apply leave-one-out ablation to Group Relative Policy Optimization (GRPO)-trained formalizations, *Solver Trace Ablation* (STA), to test whether each component is necessary for proof success, and characterize structural degeneration under six reward conditions via the resulting *Intensional Structural Reward* (ISR). ISR reveals that prove-rate gains conceal a compositional shift toward hacking modes, but ISR occupies a paradoxical position: the same leave-one-out structure that makes it informative also makes it gameable. Tautological formalizations satisfy the necessity criterion by construction, so when ISR is repurposed as a training reward, the model learns to game the diagnostic rather than the task. In our Qwen3-4B experiments, the Goodhart effect is seed-dependent (3 of 5 seeds converge to gaming under standard training conditions; 2 retain genuine reasoning), and the *diagnostic–optimization asymmetry* persists across logic types, confirmed by cross-logic (Prolog) replication. Cross-dataset (ProofFOL) evidence suggests dataset-dependent boundary conditions: structural proxies for formalization integrity are easier to observe than to optimize¹.

1 Introduction

A neurosymbolic reasoner can achieve rising prove rate while the internal structure of its proofs collapses entirely. The root cause is evaluation by a single criterion, whether the generated formalization proves under a formal solver, which treats all proofs as equal: a formalization that embeds the

conclusion verbatim among its premises receives the same reward as one that derives it through genuine reasoning. Under reinforcement learning, this indifference becomes an active vulnerability (Geirhos et al., 2020); the optimizer exploits any shortcut the reward fails to penalize (Skalse et al., 2022; Wang et al., 2026), and execution feedback provides no signal against structural shortcuts.

Neurosymbolic formalizers (Pan et al., 2023; Olausson et al., 2023; Ye et al., 2023) and RL-trained variants (Mellgren et al., 2025; Xie et al., 2025; Shao et al., 2024) rely on execution accuracy alone, with no diagnostic for whether individual components contribute to the proof. Process reward models (Lightman et al., 2024; Wang et al., 2024) offer finer granularity but require expensive step-level annotations that do not transfer to solver-backed settings, while inference-time gaming detection (Kim et al., 2026; Bondarenko et al., 2025) identifies gaming post-hoc but provides no training-time structural signal. What is missing is an annotation-free method for testing *component necessity* in neurosymbolic outputs.

We address this gap with *Solver Trace Ablation* (STA), a diagnostic that tests whether each component of a formalization is necessary for proof success. For each proved output, STA removes components one at a time and re-runs the solver; the fraction whose removal breaks the proof yields the *Intensional Structural Reward* (ISR). A formalization where every component is necessary receives $ISR=1$; one that proves regardless of which component is removed receives $ISR=0$. Leave-one-out (LOO) ablation itself is standard (DeMillo et al., 1978; Jia and Harman, 2011); our contribution is its systematic application to neurosymbolic outputs, where it reveals a diagnostic–optimization asymmetry (§4.2–4.3).

Applied post-hoc, ISR reveals that rising prove rate conceals a compositional shift in proof structure: genuine reasoning gives way to hacking

¹Code and data are available at <https://anonymous.4open.science/r/solver-trace-ablation-CB70/>.

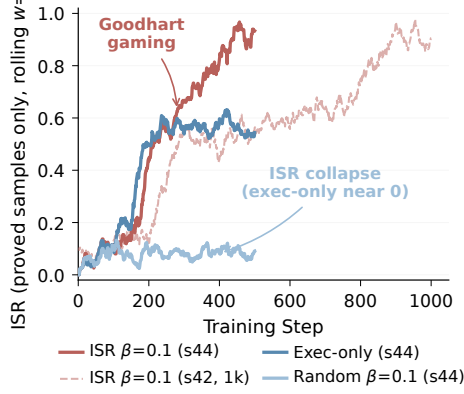


Figure 1: ISR* (proved samples only) training trajectories across reward conditions (rolling $w=20$). Exec-only collapses to $\text{ISR} \approx 0$, confirming structural degeneration (§4.2); the ~ 0.5 plateau is an $n=2$ artifact. $\text{ISR } \beta=0.1$ seeds rise toward $\text{ISR} \approx 1$ via TS gaming (Goodhart effect, §4.3); random remains low.

modes (conclusion embedding and tautological shortcuts), invisible to execution-based metrics (Figure 1; §4.2).

Yet the same LOO structure that makes ISR diagnostic also creates a Goodhart vulnerability (Manheim and Garrabrant, 2019; Karwowski et al., 2024; Gao et al., 2023): tautological formalizations ($P \rightarrow P$) satisfy the necessity criterion by construction, so when ISR is repurposed as a training reward, the model learns to game the diagnostic rather than the task. Our results suggest that LOO-based structural proxies, and plausibly other component-necessity measures, face analogous exploitation risks in neurosymbolic settings. In our Qwen3-4B experiments, the asymmetry persists across logic types (first-order logic [FOL], Prolog), reward doses, and random seeds, with dataset-dependent boundary conditions (FOLIO vs. ProofFOL; §4.4), suggesting that structural proxies for formalization integrity are easier to observe than to optimize.

We contribute: (C1) a systematic empirical analysis of structural degeneration in GRPO-trained neurosymbolic reasoning, using LOO ablation (STA/ISR) to catalogue eight hacking modes across six reward conditions (§§3.2–4.4); (C2) empirical characterization of a seed-dependent Goodhart effect when ISR is repurposed as reward, with matched controls ruling out confounds (§4.3); (C3) cross-logic replication (FOL/Prolog) confirming that the Goodhart effect transfers across logic types, with exploratory cross-dataset evidence (FOLIO/ProofFOL; §4.4).

2 Related Work

Neurosymbolic reasoning. Large language model (LLM)–solver pipelines (Ye et al., 2023; Pan et al., 2023; Olausson et al., 2023; Pei et al., 2025; Di et al., 2025; Yang et al., 2022) and RL methods trained on solver feedback (Mellgren et al., 2025; Xie et al., 2025; DeepSeek AI, 2024; DeepSeek-AI, 2025b; Jha et al., 2024) evaluate execution accuracy alone, with no diagnostic for component necessity. AlphaProof (Hubert et al., 2025) and DeepSeek-Prover-V2 (DeepSeek-AI, 2025a) apply RL-based formal verification at scale in Lean 4 for competition mathematics; STA-style diagnostics could probe whether similar structural degeneration occurs in these settings. We use FOLIO (Han et al., 2024), the only public natural language inference (NLI) benchmark with both FOL and Prolog annotations.

Reward hacking and Goodhart’s Law. Specification gaming (Krakovna et al., 2020) and Goodhart’s Law (Manheim and Garrabrant, 2019; Skalse et al., 2022; Karwowski et al., 2024) are established across RL from human feedback (RLHF) (Gao et al., 2023) and logical formalization (Bondarenko et al., 2025; Kim et al., 2026). Most closely related, Kim et al. (2026) document inference-time gaming on FOLIO with frontier models; our work complements theirs by studying training-time degeneration via component necessity. Kwa et al. (2024) show that heavy-tailed reward misspecification causes catastrophic Goodhart where Kullback–Leibler (KL) regularization provably fails, providing a theoretical lens for why GRPO’s KL penalty does not prevent our observed collapse. Concurrent work documents spurious signals in RL with verifiable rewards (RLVR) (Shao et al., 2025; Yan et al., 2026; Dong et al., 2026; Von Arx et al., 2025; Helff et al., 2026; Nishimura-Gasparian et al., 2026; Khalifa et al., 2026) and proposes composite rewards (Tarek and Beheshti, 2025; Pronesti et al., 2026). Our Goodhart effect follows the regression pattern (Manheim and Garrabrant, 2019); its application to a diagnostic designed to detect gaming, and the dataset-dependent asymmetry, are to our knowledge novel.

Reward design and structural diagnostics. Process reward models (Lightman et al., 2024; Setlur et al., 2025; Cui et al., 2025; Cobbe et al., 2021; Uesato et al., 2022) supervise intermediate steps; Sullivan and Koller (2025) show GRPO implicitly

implements process reward. Classical reward shaping (Ng et al., 1999; Gao et al., 2024) guarantees policy invariance only under potential-based transforms; ISR violates this condition. Proposed mitigations include reward model ensembles (Coste et al., 2024), constrained optimization (Moskovitz et al., 2024), and activation-level monitoring (Wilhelm et al., 2026); Yue et al. (2025) show RLVR redistributes probability mass rather than creating new reasoning, consistent with our finding that GRPO exploits existing shortcuts. STA adapts the LOO mutation paradigm (DeMillo et al., 1978; Jia and Harman, 2011) to solver inputs; novelty lies in application and findings, not in the perturbation primitive. Related concerns about chain-of-thought (CoT) faithfulness (Lanham et al., 2023; Turpin et al., 2023) operate at the reasoning-chain level rather than solver-input components.

3 Method

3.1 Task and GRPO Training

Given premises $P_1, \dots, P_k$ and conclusion C , the task is to produce a formal program $F = \{c_1, \dots, c_n\}$ verified by an external solver $S \in \{\text{Prover9, SWI-Prolog}\}$ (McCune, 2005–2010; Wielemaker et al., 2012). We train with Group Relative Policy Optimization (GRPO) (Shao et al., 2024) ($K=4$), where each rollout receives binary execution reward $r_{\text{exec}}(x) = \mathbf{1}[S(F_x)=\text{proved}]$. This optimizes prove rate but provides no signal about *component necessity* (Figure 2a): a formalization embedding the conclusion among its premises receives the same reward as a faithful derivation. All experiments draw from FOLIO (Han et al., 2024); the held-out evaluation set (100 examples) contains 34 True, 25 False, and 41 Unknown labels, with only True-labeled examples admitting valid proofs (theoretical PR ceiling $\sim 34\%$). Exec-only GRPO reaching PR=93% (§4.2), nearly $3\times$ this ceiling, confirms the solver accepts formalizations for nominally unprovable examples, constituting direct evidence of structural gaming independent of ISR.

3.2 Solver Trace Ablation (STA)

STA tests whether each component of a proved formalization is individually necessary for proof success (Figure 2; Algorithm 1). Given a proved x with n components, STA constructs n ablated variants $x \setminus \{c_i\}$ and re-runs the solver (Algorithm 1). A component is *necessary* if its removal causes proof

Algorithm 1 Solver Trace Ablation (STA)

Require: Proved formalization $x = \{c_1, \dots, c_n\}$, solver S
Ensure: $\text{ISR}(x) \in [0, 1]$
1: $necessary \leftarrow 0$
2: **for** $i = 1$ to n **do**
3: **if** $S(x \setminus \{c_i\}) = \text{failed}$ **then**
4: $necessary \leftarrow necessary + 1$
5: **end if**
6: **end for**
7: **return** $necessary / n$

failure. What constitutes a component depends on the logic: premises in FOL, clauses or ground facts in Prolog. LOO ablation itself is standard in mutation testing and sensitivity analysis (DeMillo et al., 1978; Jia and Harman, 2011); novelty lies in its application as a GRPO-hacking diagnostic and the empirical phenomena it surfaces.

Intensional Structural Reward. ² For a proved formalization $x = \{c_1, \dots, c_n\}$, ISR is the fraction of necessary components:

$$\text{ISR}(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}[S(x \setminus \{c_i\}) = \text{failed}] \quad (1)$$

ISR is defined only for proved formalizations: ISR=1 means every component is necessary; ISR=0 means complete redundancy. An ISR of zero for proved samples can arise from structural degeneration or genuine redundancy; our hacking mode classification (§3.3) disambiguates these cases, with $>90\%$ of low-ISR proved samples exhibiting identifiable hacking patterns. ISR computes the LOO marginal contribution, a restriction of the Shapley value to the maximal coalition (Appendix B); full interaction detection would require $O(2^n)$ evaluations. ISR quantifies component redundancy, not semantic faithfulness: Spearman $\rho(\text{ISR}, \text{NLI accuracy}) = -0.049$ ($p=0.187$, n.s.), confirming the two are uncorrelated (Appendix B). Its diagnostic value is realized through the downstream mode breakdown (§4.5), not the scalar alone. We report *proved-only* ISR (denoted ISR^*) throughout to avoid conflating component necessity with prove rate.

LOO independence assumption. LOO ablation tests individual, not joint, component necessity (Jia and Harman, 2011), systematically underestimating ISR when jointly-necessary components exist; the observed collapse is therefore conservative.

²Not *intentional*; *intensional* contrasts with *extensional* (execution-only) evaluation.

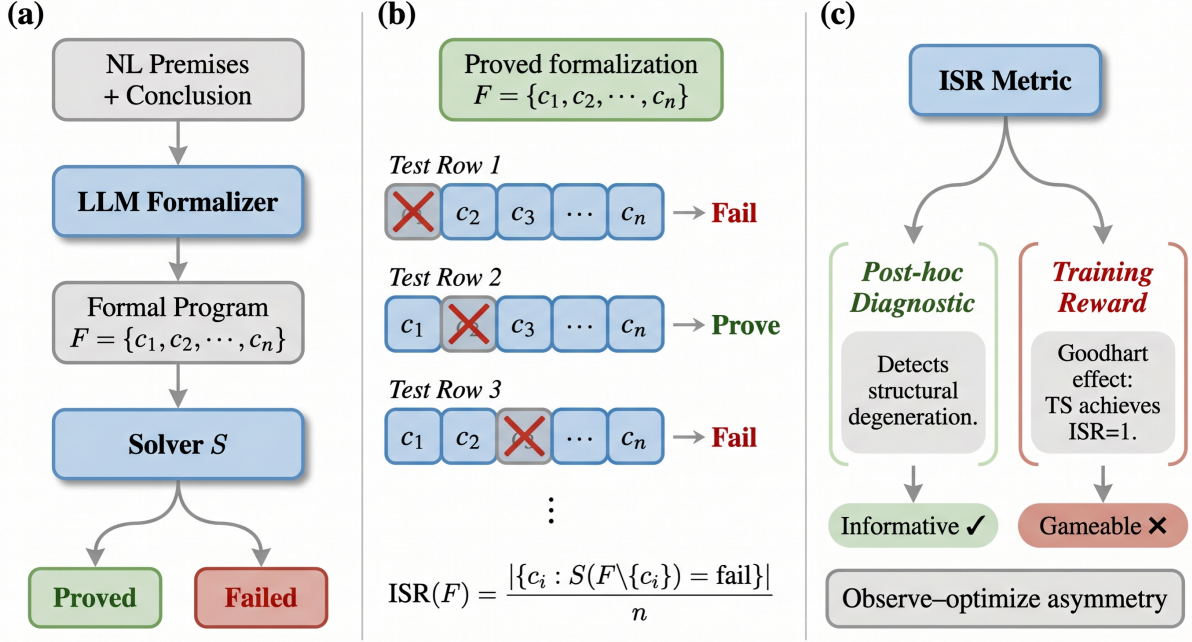


Figure 2: STA overview. (a) The formalization pipeline: an LLM translates natural-language (NL) premises into formal logic, verified by an external solver. (b) STA removes each component in turn and re-runs the solver; the fraction of necessary components yields ISR. (c) Applied post-hoc, ISR detects structural hacking; repurposed as reward, it triggers a Goodhart effect where tautological shortcuts (TS) achieve $\text{ISR}=1$ by construction.

3.3 Reward Conditions and Hacking Modes

We study six conditions: *exec-only* (r_{exec}), *ISR low/high-dose* ($r_{\text{exec}} \cdot (1 + \beta \text{ISR})$ at $\beta \in \{0.1, 0.3\}$), *matched random controls* ($r_{\text{exec}} \cdot (1 + \beta u)$, $u \sim \mathcal{U}(0, 1)$), and a *supervised fine-tuning (SFT) baseline* (negative log-likelihood on gold formalizations). Multiplicative bonuses apply only to proved samples, keeping the prove-rate floor intact. Random isolates ISR-specific signal from generic reward-variance effects; a shuffled-ISR control permutes ISR scores across samples to test whether Goodhart effects require sample-level signal; SFT isolates whether structural degeneration arises from RL optimization or data exposure alone. Each RL condition uses seeds {42, 43, 44} and identical training data (formulation alternatives in Appendix A).

Hacking taxonomy. Across conditions, seeds, and logic types we identify eight structural-degeneration modes (full definitions in Appendix C), dominated by three: **CE** (conclusion embedded among multiple premises), **TS** (single-premise conclusion duplication; targets ISR directly since removing the sole premise forces $\text{ISR} \geq 0.5$), and **PG** (premise-subset gaming: each component is LOO-necessary but the

premise set is incomplete; Appendix C.1). Five additional modes (tautological collapse, repetition, complexity-without-utility, contradiction explosion, and query-as-fact [Prolog-specific]) appear in specific seeds or conditions. CE and TS are robust across conditions and seeds; the remaining five modes are configuration-specific observations whose generalizability requires further investigation. Modes are identified by strict text matching (CE/TS) plus functional analysis of the per-sample $n_{\text{necessary}}$ distribution; when patterns overlap, we assign the most specific mode ($\text{PG} > \text{CE} > \text{TS}$). The non-hacking rate is an upper bound on genuine reasoning.

Low ISR signals redundancy under *exec-only* but metric gaming under *ISR-as-reward*; the mode breakdown, not the ISR scalar, licenses interpretation.

4 Experiments

We evaluate STA along three axes: ISR as a diagnostic for reward hacking (§4.2); ISR as a training reward, where it exhibits Goodhart effects (§4.3); and cross-domain generalization across logic types and dataset families (§4.4). A hacking taxonomy synthesizes the modes observed across conditions (§4.5).

Condition	β	PR% $\uparrow$	ISR*	TS%	CE%	N-H%
Base (0-shot)	—	12	—	—	—	—
SFT	—	13	0.595	0	8	92
Exec-only †	—	93	0.132	0	87	8
ISR (3-seed)	0.1	77 $\pm$ 14	1.06 $\times$ ‡	71 ¶	18	6
ISR (5-seed)	0.1	70 $\pm$ 19	(0.63) ‡	54	18	23
Random	0.1	87 $\pm$ 8	0.117 $\pm$.04	14 §	69	10
Shuffled-ISR	0.1	15 $\pm$ 1	0.559 $\pm$.02	7	7	87
ISR	0.3	66 $\pm$ 42	0.39 $\pm$.31	50	37	11
Random	0.3	54 $\pm$ 35	0.09 $\pm$.09	7	91	2

Table 1: Goodhart landscape across reward conditions. PR = prove rate; ISR* = proved-only ISR; TS = tautological shortcut; CE = conclusion embedding; N-H = non-hacking (upper bound on genuine reasoning). Multi-seed rows: pooled TS/CE/N-H; PR/ISR* as mean $\pm$ std. † Seed 42 only. ‡ Premise-count-matched (§4.3). ¶ FN-corrected (§C.1). § Seed-dependent (s43: 38.3%). Per-seed data: Table 9.

4.1 Setup

We train Qwen3-4B (Qwen Team, 2025) with GRPO (Shao et al., 2024) ($K=4$, lr 10^{-6} , 500 steps default, 1000 where noted; full hyperparameters in Table 4) over seeds {42, 43, 44} (ISR $\beta=0.1$ extends to {45, 46}; Appendix D) on a fixed 500-example FOLIO (Han et al., 2024) subset held constant across all conditions; the SFT baseline uses negative log-likelihood on gold formalizations.³ Held-out evaluation uses 100 FOLIO examples with greedy decoding;⁴ all cross-run ISR values are recomputed uniformly by a single evaluator so training-log ISR is never compared across runs. Cross-dataset experiments (§4.4) use ProofFOL, a silver-standard FOL corpus of ~ 500 training examples constructed by translating ProofWriter (Tafjord et al., 2021) problems via GPT-4o and filtering through Prover9 ($\sim 70\%$ retention; Appendix A, Table 2); *stable convergence* denotes $\overline{\text{PR}}_{\text{final } 50} > 0.90$. We report prove rate (PR), proved-only ISR (ISR*), hacking-mode rates from a two-stage classifier (strict text matching for CE/TS plus functional analysis of the per-sample $n_{\text{necessary}}$ distribution; §3.3; classifier blind spots discussed in Appendix B), and a *non-hacking* (N-H) rate, an upper bound on genuine reasoning: the residual matching none of the eight cataloged patterns (classifier recall unknown; 152-sample precision audit in Table 8).

Statistical considerations. With three seeds exhibiting multi-attractor dynamics, per-seed qualita-

³Seeds control initialization and rollout sampling; all conditions share the same fixed training subset.

⁴Seed 43 uses nucleus sampling ($T=0.7$) due to greedy-mode degeneration; excluding s43 yields consistent or stronger collapse: at 1000 steps, both remaining seeds show $\leq 7\%$ non-hacking (Appendix D).

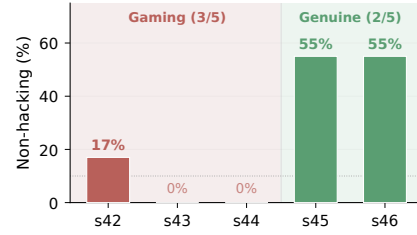


Figure 3: Seed-dependent Goodhart effect under ISR $\beta=0.1$ reward (500 steps). Three seeds converge to gaming (N-H $\leq 17\%$); two retain genuine reasoning ($\sim 55\%$).

tive analysis is the primary framework; the divergence itself is a finding. Pooled Cochran–Mantel–Haenszel p -values ($N=3$ strata) are descriptive summaries, not formal inference; non-significant comparisons are exploratory. An $N=2$ sensitivity analysis excluding s43 confirms consistent or stronger patterns (Appendix D.6). ProofFOL results ($p=0.10$, $N=3$) are exploratory and under-

4.2 ISR as Diagnostic

Collapse trajectory. Figure 1 traces exec-only GRPO (seed 42) across 1000 training steps. As PR climbs from 32% to 96%, ISR* collapses from 0.642 to 0.003, a 214 $\times$ raw drop (partially attributable to premise-count inflation; matched comparison yields 1.06 $\times$, §4.3) consistent with structural degeneration behind apparent success. By 500 steps, 87% of proved samples embed the conclusion verbatim (CE), establishing degeneration independent of the ISR scalar. A binary variant testing only collective premise necessity misses the collapse entirely (ISR=1.0 for all proved samples), confirming that per-component LOO testing is essential (Appendix D.3). All three exec-only

seeds converge to $\leq 6.2\%$ non-hacking by 1000 steps (Table 9), confirming that structural degeneration under exec-only reward is not seed-specific among converged runs.

Cross-seed instability. Three seeds diverge into qualitatively different failure modes: conclusion embedding (s42), repetition collapse (s43), and tautological collapse (s44), yet all converge to $\leq 6.2\%$ non-hacking by 1000 steps (Table 9; Appendix D). Exec-only seeds exhibit qualitatively different convergence patterns (PR range: 13–98%), precluding meaningful pooling; we report seed 42 as representative of the converged regime (Table 1). Two additional seeds under identical conditions fail to converge (PR $\approx 14\%$; Appendix D.5), retaining genuine reasoning and suggesting that structural degeneration requires successful convergence, though the confound between model capability and optimization success precludes strong causal attribution (Appendix D.4).

SFT baseline. SFT on gold formalizations produces 92% non-hacking outputs at PR=13% (odds ratio =147 vs. exec-only, $p < 0.001$; Table 6), establishing that hacking modes are absent before RL optimization. SFT serves as a pre-RL reference rather than a PR-matched control; its low PR reflects minimal optimization pressure. Early-training comparison supports this interpretation: at 100 steps (PR=32%), hacking already reaches 31% vs. SFT’s 8% at comparable PR (Appendix D.2).

4.3 Goodhart Effects on ISR-as-Reward

We reverse ISR’s role from observer to reward: $r = r_{\text{exec}} \cdot (1 + \beta \text{ISR})$, with matched random control ($u \sim \mathcal{U}(0, 1)$). The result is a regressional Goodhart pattern (Manheim and Garrabrant, 2019): optimizing a structural proxy shifts the *composition* of failure modes without reducing their aggregate prevalence. ISR reward systematically selects for tautological shortcuts (TS) over conclusion embedding (CE) rather than producing genuine reasoning (Table 1). The scalar ISR* advantage ($214\times$ raw) collapses to $1.06\times$ after controlling for premise count; the compositional mode shift provides the primary evidence.

Multiplicative coupling. Three controls constrain the multiplicative confound: exec-only’s convergence to pure hacking, the qualitative TS shift under ISR, and an additive variant that did not converge (Appendix A). The shuffled-ISR control (87% non-hacking at low PR) suggests Goodhart

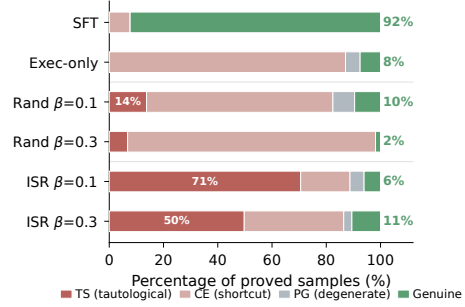


Figure 4: Hacking composition across reward conditions (proved samples; exec-only: seed 42; all other bars: pooled across 3 seeds, per-seed breakdown in Appendix Table 9). ISR systematically selects for TS over CE without reducing total hacking; SFT shows near-zero hacking.

effects require sample-level ISR signal, though non-convergence cannot be ruled out (Appendix D.5).

Low-dose ISR ($\beta=0.1$). ISR reward amplifies TS convergence: 71% of proved samples under ISR exhibit tautological shortcuts vs. 14% under random (Appendix B). Total conclusion embedding (CE + TS) spans 68–100% across both conditions; the composition shifts (CE dominates under random, TS under ISR) without reducing aggregate hacking (Figure 4). A seed-stratified χ^2 test confirms the overall mode distribution differs significantly between ISR and random conditions ($\chi^2=174.8$, $df=2$, $p < 10^{-38}$; Appendix B).

Seed-dependent outcomes. ISR reward produces bimodal outcomes (Figure 3): 3 of 5 seeds converge to gaming, demonstrating that the Goodhart effect can occur under standard training conditions (95% Clopper–Pearson CI: [11.8%, 88.1%]; $n=5$); 2 of 5 retain substantial genuine reasoning (seeds 45/46 added post-hoc to test replication; Appendix D). Excluding s43, converged seeds show $\leq 17\%$ non-hacking at 500 steps, collapsing to $\leq 2\%$ by 1000 (Appendix D.6).

Premise-count confound. The $214\times$ raw ISR* collapse is largely explained by premise-count reduction: ISR-trained models produce shorter formalizations that mechanically inflate ISR. The premise-count-matched comparison yields only $1.06\times$ (McNemar $p=0.317$, $n=100$ matched pairs; Appendix B), indicating that the scalar ISR advantage is not statistically significant.

The compositional evidence, however, survives this control. In the modal premise-count range (4–6 premises, where most proved samples fall), TS reaches 19.7% under ISR versus 0% under ran-

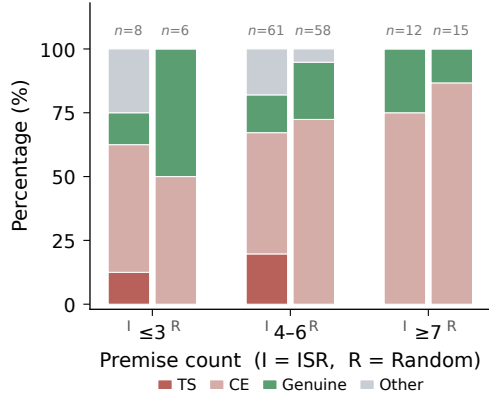


Figure 5: Mode composition by premise count (proved samples, seed 42, $\beta=0.1$). TS hacking emerges exclusively under ISR in the 4–6 premise range; Random shows zero TS at any complexity, confirming that ISR’s compositional shift is not a premise-count artifact. I = ISR, R = Random.

Property	ProofFOL	FOLIO
FOL premises (mean $\pm$ std)	18.2 $\pm$ 3.2	5.3 $\pm$ 1.4
Ground facts	10.2 $\pm$ 3.0	1.8 $\pm$ 1.5
Rules	8.1 $\pm$ 1.1	3.5 $\pm$ 1.9
XOR ($\oplus$) samples	0%	37.2%
Dominant depth	4–5	2–3
FOL source	GPT-4o (silver)	human (gold)

Table 2: ProofFOL vs. FOLIO structural comparison ($n=500$ training examples each). ProofFOL has 3.4 $\times$ more premises with simpler synthetic predicates.

dom reward (Figure 5; Table 5). TS hacking is mechanistically limited to simpler problems; this complexity dependence is a mechanistic insight, not a confound. The compositional mode shift, not the scalar ISR difference, constitutes the primary evidence for the Goodhart effect.

Higher doses and temporal dynamics. Any genuine reasoning under ISR $\beta=0.1$ is transient: non-hacking drops from 17% to 2% by 1000 steps (Figure 1). Higher doses accelerate collapse: $\beta=0.2$ eliminates all non-hacking; $\beta=0.3$ destabilizes training itself (Gao et al., 2023) (Appendix D.1).

4.4 Cross-Domain and Cross-Dataset Evidence

The following results extend the primary findings of §4.2–4.3 to alternative logic types and dataset families; these cross-domain experiments are exploratory in nature.

Prolog: universal Goodhart. Replicating the pipeline with Prolog/SWI-Prolog (Figure 6), exec-only seeds collapse to $\text{ISR}^* \approx 0$ via redun-

Mode	Exec	ISR ₁	Rnd ₁	SFT	Prolog _{ISR}
CE embed	87	18	68	8	0 [#]
TS duplicate	0	71	14	0	100 [#]
PG prem-subset	5	5	8	0	—
Tautological	—	—	—	—	s42
Repetition	s43	—	—	—	—
Contradiction	1k	—	—	—	—
Cmplx. vacuous	0	0	1	0	—
Query-as-fact	—	—	—	—	s44
Non-hacking	8	6	10	92	0

Table 3: Eight hacking modes across conditions (% of proved samples; exec-only/SFT: seed 42; ISR/Random: 3-seed pooled). TS includes subtypes TSa (text-match) and TSb (obfuscated tautology). Cells with seed identifiers indicate seed-specific modes. [#]Prolog pools strict+functional classification. Extended breakdown in Table 7.

dant ground-fact coding, supporting STA’s diagnostic transfer across logic types. Under ISR-as-reward, all three seeds reach 0% non-hacking through distinct mechanisms: clean TS (s42), degenerate syntax bypassing text-match classifiers (s43), and query-as-fact gaming (s44; Table 10). Prolog’s simpler semantics broadens the gaming basin, eliminating the residual genuine reasoning seen in FOL.

ProofFOL exec-only: cross-dataset replication. Converged seeds collapse to near-zero genuine reasoning, confirming STA’s diagnostic across dataset families. The mode repertoire differs across datasets: among converged seeds, TS dominates one while CE dominates the other, showing that mode repertoire is dataset-specific while the degeneration phenomenon is universal.

ProofFOL ISR ($\beta=0.1$): dataset-dependent asymmetry. ISR reward on ProofFOL produces directionally higher genuine rates than exec-only ($p=0.10$, exploratory; Table 11). The asymmetry follows from hacking geometry: on FOLIO, TS yields high ISR (regressional Goodhart (Manheim and Garrabrant, 2019)); on ProofFOL, CE embeds conclusions among many premises, yielding $\text{ISR} \approx 0$ and rendering ISR reward effectively corrective.

Cross-scale validation (Qwen3-8B). Qwen3-8B does not converge under exec-only GRPO ($\text{PR}=14\%$, $K=8$; $K=4$ produces zero gradient), consistent with degeneration requiring successful convergence (Appendix D.7).

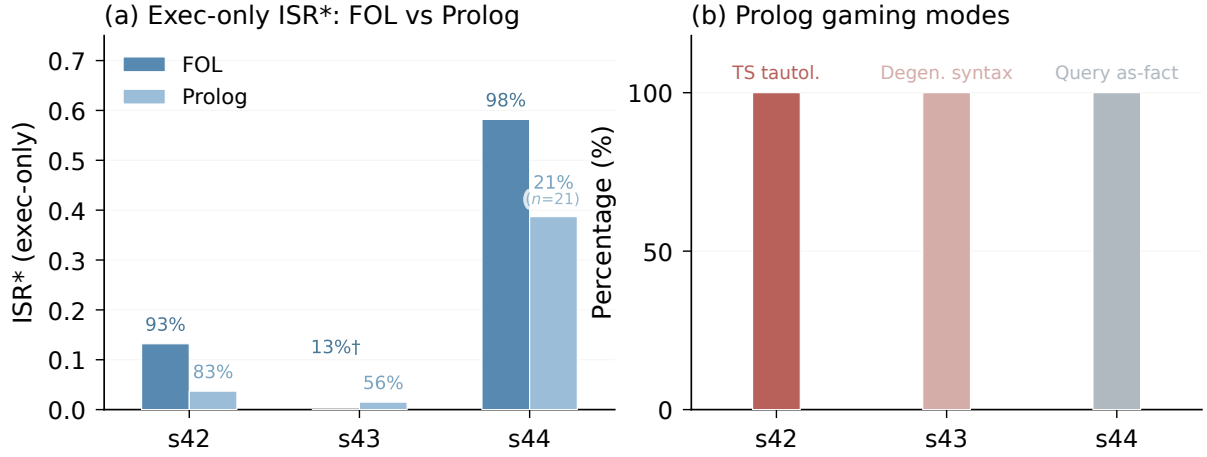


Figure 6: Cross-logic results. (a) FOL vs. Prolog exec-only ISR* across 3 seeds (s44 Prolog 0.387 is a small-sample artifact, $n=21$; †s43 FOL uses sampling $T=0.7$, not strictly comparable). (b) Prolog ISR-reward gaming-mode composition; all 3 seeds reach 0% non-hacking through distinct mechanisms.

4.5 Hacking Taxonomy

Three modes account for the vast majority of hacking (Table 3): CE (conclusion embedding), TS (conclusion duplication), and PG (premise-subset gaming). The central pattern: *reward formulation determines which mode dominates but not whether hacking occurs*; all GRPO conditions produce overwhelming degeneration, while SFT achieves 92% non-hacking.

5 Discussion and Conclusion

STA exposes structural degeneration invisible to execution-based evaluation. When ISR is used post-hoc, it reliably detects that prove-rate gains conceal a compositional shift toward hacking modes: across all converged exec-only seeds, non-hacking rates collapse to $\leq 6.2\%$ by 1000 steps, despite prove rates exceeding 90%. The diagnostic holds across logic types (FOL, Prolog) and dataset families (FOLIO, ProofFOL), confirming that structural degeneration under execution-only reward is a general phenomenon rather than an artifact of a specific formalism.

When repurposed as a training reward, ISR produces a seed-dependent Goodhart effect that follows the regressional pattern (Manheim and Garrabrant, 2019): optimizing ISR shifts the composition of hacking modes (from CE to TS) without reducing their aggregate prevalence. The effect is paradoxical in a precise sense: the same LOO structure that makes ISR informative as a diagnostic also makes it gameable as a reward, because tautological formalizations satisfy the necessity

criterion by construction. Cross-logic replication (Prolog) confirms the compositional mode shift; cross-dataset evidence (ProofFOL, $p=0.10$) is exploratory (§4.4).

These findings yield five practical guidelines for GRPO-trained neurosymbolic systems. First, never evaluate by prove rate alone—use ISR and mode breakdown as joint diagnostics, applied post-hoc rather than as reward signals; any dose of ISR-as-reward induces gaming or collapse. Second, report multi-seed results: single-seed evaluation is unreliable when cross-seed variance exceeds the effect size (§4.2). Third, treat LOO-based metrics as component-count-biased until premise-count-matched (§4.3). Fourth, expect conclusion embedding as a GRPO-universal baseline: reward shaping alters its composition, not its prevalence (Table 1). Fifth, monitor convergence: structural degeneration appears contingent on successful optimization (Appendix D.5), making monitoring most critical when training succeeds. The diagnostic–optimization asymmetry motivates targeted mitigations such as single-premise penalties or mode-aware reward shaping (Tarek and Beheshti, 2025; Pronesti et al., 2026). Remaining questions include extending the hacking taxonomy to process reward settings and determining whether the seed-dependent Goodhart boundary generalizes beyond the model scale examined here.

Limitations

Experiments use a single model (Qwen3-4B) with three to five seeds; degeneration patterns may not transfer to other families, and the 3/5 gaming

rate (seeds 45–46 added post-hoc) is an existence demonstration, not a prevalence estimate. FOLIO is a relatively small (1,435 examples) synthetic benchmark; generalization to larger or naturalistic corpora is untested. ISR quantifies component necessity, not semantic faithfulness; diagnostic value comes from the mode breakdown, not the scalar alone (Appendix B).

References

Alexander Bondarenko, Denis Volk, Dmitrii Volkov, and Jeffrey Ladish. 2025. [Demonstrating Specification Gaming in Reasoning Models](#). *arXiv preprint arXiv:2502.13295*.

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. *arXiv preprint arXiv:2110.14168*.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2024. Reward model ensembles help mitigate overoptimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ArXiv:2310.02743.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, and 6 others. 2025. [Process Reinforcement through Implicit Rewards](#). *arXiv preprint arXiv:2502.01456*.

DeepSeek AI. 2024. [DeepSeek-Prover-V1.5: Harnessing Proof Assistant Feedback for Reinforcement Learning and Monte-Carlo Tree Search](#). *arXiv preprint arXiv:2408.08152*.

DeepSeek-AI. 2025a. [DeepSeek-Prover-V2: Advancing Formal Mathematical Reasoning via Reinforcement Learning for Subgoal Decomposition](#). *arXiv preprint arXiv:2504.21801*.

DeepSeek-AI. 2025b. [DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning](#). *arXiv preprint arXiv:2501.12948*.

Richard A. DeMillo, Richard J. Lipton, and Frederick G. Sayward. 1978. Hints on Test Data Selection: Help for the Practicing Programmer. *Computer*, 11(4):34–41.

Zhengkun Di, Chaoli Zhang, Hongtao Lv, Lizhen Cui, and Lei Liu. 2025. LoRP: LLM-based Logical Reasoning via Prolog. *Knowledge-Based Systems*.

Yiming Dong, Kun Fu, Haoyu Li, Xinyuan Zhu, Yurou Liu, Lijing Shao, Jieping Ye, and Zheng Wang. 2026. [Probing RLVR Training Instability Through](#)

[the Lens of Objective-Level Hacking](#). *arXiv preprint arXiv:2602.01103*.

Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. 2024. [On Designing Effective RL Reward at Training Time for LLM Reasoning](#). *arXiv preprint arXiv:2410.15115*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling Laws for Reward Model Overoptimization. In *Proceedings of the International Conference on Machine Learning (ICML)*. ArXiv:2210.10760.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and 1 others. 2024. [The Llama 3 Herd of Models](#). *arXiv preprint arXiv:2407.21783*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, and 7 others. 2024. FOLIO: Natural Language Reasoning with First-Order Logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ArXiv:2209.00840.

Lukas Helff, Quentin Delfosse, David Steinmann, Ruben Härle, Hikaru Shindo, Patrick Schramowski, Wolfgang Stammer, Kristian Kersting, and Felix Friedrich. 2026. [LLMs Gaming Verifiers: RLVR can Lead to Reward Hacking](#). *arXiv preprint arXiv:2604.15149*.

Thomas Hubert, Rishi Mehta, Laurent Sartran, and 1 others. 2025. [Olympiad-Level Formal Mathematical Reasoning with Reinforcement Learning](#). *Nature*.

Piyush Jha, Prithwish Jana, Pranavkrishna Suresh, Arnav Arora, and Vijay Ganesh. 2024. [RLSF: Fine-tuning LLMs via Symbolic Feedback](#). *arXiv preprint arXiv:2405.16661*.

Yue Jia and Mark Harman. 2011. An Analysis and Survey of the Development of Mutation Testing. *IEEE Transactions on Software Engineering*, 37(5):649–678.

Jacek Karwowski, Oliver Hayman, Xingjian Bai, Klaus Kiendlhofer, Charlie Griffin, and Joar Skalse. 2024. Goodhart’s Law in Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ArXiv:2310.09144.

Muhammad Khalifa, Zohaib Khan, Omer Tafveez, Hao Peng, and Lu Wang. 2026. [Countdown-Code: A Testbed for Studying The Emergence and Generalization of Reward Hacking in RLVR](#). *arXiv preprint arXiv:2603.07084*.

670	Kyuhee Kim, Auguste Poiroux, and Antoine Bosse-	Language Models with First-Order Logic Provers. In	726
671	lut. 2026. Do LLMs Game Formalization? Evaluat-	<i>Proceedings of the Conference on Empirical Methods</i>	727
672	ing Faithfulness in Logical Reasoning. In <i>VerifAI-2</i>	<i>in Natural Language Processing (EMNLP)</i> .	728
673	<i>Workshop at the International Conference on Learn-</i>		
674	<i>ing Representations (ICLR)</i> . ArXiv:2604.19459.		
675	Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik,	Liangming Pan, Alon Albalak, Xinyi Wang, and	729
676	Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac	William Yang Wang. 2023. Logic-LM: Empower-	730
677	Kenton, Jan Leike, and Shane Legg. 2020. Specifi-	ing Large Language Models with Symbolic Solvers	731
678	cation gaming: The flip side of AI ingenuity. Deep-	for Faithful Logical Reasoning. In <i>Findings of the</i>	732
679	Mind Blog.	<i>Conference on Empirical Methods in Natural Lan-</i>	733
		<i>guage Processing (EMNLP)</i> .	734
680	Thomas Kwa, Drake Thomas, and Adrià Garriga-	Yu Pei, Yongping Du, and Xingnan Jin. 2025. FoVer:	735
681	Alonso. 2024. Catastrophic Goodhart: Regulariz-	First-Order Logic Verification for Natural Language	736
682	ing RLHF with KL Divergence Does Not Mitigate	Reasoning. <i>Transactions of the Association for Com-</i>	737
683	Heavy-Tailed Reward Misspecification. In <i>Advances</i>	<i>putational Linguistics</i> , 13:1340–1359.	738
684	<i>in Neural Information Processing Systems (NeurIPS)</i> .		
685	ArXiv:2407.14503.	Massimiliano Pronesti, Anya Belz, and Yufang Hou.	739
686	Tamera Lanham, Anna Chen, Ansh Radhakrishnan,	2026. <i>Beyond Outcome Verification: Verifiable Pro-</i>	740
687	Benoit Steiner, Carson Denison, Danny Hernandez,	<i>cess Reward Models for Structured Reasoning</i> . <i>arXiv</i>	741
688	Dustin Li, Esin Durmus, Evan Hubinger, Jackson	<i>preprint arXiv:2601.17223</i> .	742
689	Kernion, Kamilė Lukosiute, Karina Nguyen, Newton	Qwen Team. 2025. <i>Qwen3 Technical Report</i> . <i>arXiv</i>	743
690	Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver	<i>preprint arXiv:2505.09388</i> .	744
691	Rausch, Robin Larson, Sam McCandlish, Sandipan	Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang	745
692	Kundu, and 11 others. 2023. <i>Measuring Faithful-</i>	Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh	746
693	<i>ness in Chain-of-Thought Reasoning</i> . <i>arXiv preprint</i>	Agarwal, Jonathan Berant, and Aviral Kumar. 2025.	747
694	<i>arXiv:2307.13702</i> .	Rewarding Progress: Scaling Automated Process Ver-	748
695	Hunter Lightman, Vineet Kosaraju, Yura Burda, and 1	ifiers for LLM Reasoning. In <i>Proceedings of the In-</i>	749
696	others. 2024. Let’s Verify Step by Step. In <i>Proceed-</i>	<i>ternational Conference on Learning Representations</i>	750
697	<i>ings of the International Conference on Learning</i>	<i>(ICLR)</i> . ArXiv:2410.08146.	751
698	<i>Representations (ICLR)</i> . ArXiv:2305.20050.		
699	David Manheim and Scott Garrabrant. 2019. <i>Catego-</i>	Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng,	752
700	<i>rizing Variants of Goodhart’s Law</i> . <i>arXiv preprint</i>	Yiping Wang, Sewoong Oh, Simon Shaolei Du,	753
701	<i>arXiv:1803.04585</i> .	Nathan Lambert, Sewon Min, Ranjay Krishna, Yu-	754
702	William McCune. 2005–2010. Prover9 and Mace4.	lia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh,	755
703	https://www.cs.unm.edu/~mccune/prover9/ .	and Luke Zettlemoyer. 2025. <i>Spurious Rewards: Re-</i>	756
704	Version 2009-11A.	<i>thinking Training Signals in RLVR</i> . <i>arXiv preprint</i>	757
705	Niklas Mellgren, Peter Schneider-Kamp, and Lukas	<i>arXiv:2506.10947</i> .	758
706	Galke. 2025. <i>Training Language Models to Use Pro-</i>	Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu,	759
707	<i>log as a Tool</i> . <i>arXiv preprint arXiv:2512.07407</i> .	and 1 others. 2024. <i>DeepSeekMath: Pushing the</i>	760
708	Ted Moskovitz, Aldo Pacchiano, Parker Lund, and	<i>Limits of Mathematical Reasoning in Open Language</i>	761
709	Jakob Foerster. 2024. Confronting reward model	<i>Models</i> . <i>arXiv preprint arXiv:2402.03300</i> .	762
710	overoptimization with constrained RLHF. In <i>Pro-</i>	Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krashenin-	763
711	<i>ceedings of the International Conference on Learning</i>	nikov, and David Krueger. 2022. Defining and	764
712	<i>Representations (ICLR)</i> . ArXiv:2310.04373.	Characterizing Reward Hacking. In <i>Advances in</i>	765
713	Andrew Y. Ng, Daishi Harada, and Stuart Russell. 1999.	<i>Neural Information Processing Systems (NeurIPS)</i> .	766
714	Policy invariance under reward transformations: The-	ArXiv:2209.13085.	767
715	ory and application to reward shaping. In <i>Proceed-</i>	Michael Sullivan and Alexander Koller. 2025. <i>GRPO</i>	768
716	<i>ings of the International Conference on Machine</i>	<i>is secretly a process reward model</i> . <i>arXiv preprint</i>	769
717	<i>Learning (ICML)</i> , pages 278–287.	<i>arXiv:2509.21154</i> .	770
718	Kei Nishimura-Gasparian, Robert McCarthy, and David	Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021.	771
719	Lindner. 2026. <i>Towards Understanding Specifica-</i>	ProofWriter: Generating Implications, Proofs, and	772
720	<i>tion Gaming in Reasoning Models</i> . <i>arXiv preprint</i>	Abductive Statements over Natural Language. In	773
721	<i>arXiv:2605.02269</i> .	<i>Findings of the Association for Computational Lin-</i>	774
722	Theo X. Olausson, Alex Gu, Ben Lipkin, Cedegao E.	<i>guistics (ACL)</i> . ArXiv:2012.13048.	775
723	Zhang, Armando Solar-Lezama, Joshua B. Tenen-	Mirza Farhan Bin Tarek and Rahmatollah Beheshti.	776
724	baum, and Roger Levy. 2023. LINC: A Neurosym-	2025. <i>Reward Hacking Mitigation using Ver-</i>	777
	bolic Approach for Logical Reasoning by Combining	<i>ifiable Composite Rewards</i> . <i>arXiv preprint</i>	778
		<i>arXiv:2509.15557</i> .	

Ramya Keerthy Thatikonda, Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. 2024. [Strategies for Improving NL-to-FOL Translation with LLMs: Data Generation, Incremental Fine-Tuning, and Verification](#). *arXiv preprint arXiv:2409.16461*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language Models Don’t Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv:2305.04388.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving Math Word Problems with Process- and Outcome-Based Feedback. In *NeurIPS 2022 Workshop on MATH-AI*.

Sydney Von Arx, Lawrence Chan, and Beth Barnes. 2025. Recent Frontier Models Are Reward Hacking. METR Technical Blog. <https://metr.org/blog/2025-06-05-recent-reward-hacking/>.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, and 1 others. 2024. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In *Proceedings of the Association for Computational Linguistics (ACL)*. ArXiv:2312.08935.

Xiaohua Wang, Muzhao Tian, Yuqi Zeng, Zisu Huang, and 1 others. 2026. [Reward Hacking in the Era of Large Models: Mechanisms, Emergent Misalignment, Challenges](#). *arXiv preprint arXiv:2604.13602*.

Jan Wielemaker, Tom Schrijvers, Markus Triska, and Torbjörn Lager. 2012. *SWI-Prolog. Theory and Practice of Logic Programming*, 12(1–2):67–96.

Patrick Wilhelm, Thorsten Wittkopp, and Odej Kao. 2026. Monitoring emergent reward hacking during generation via internal activations. *arXiv preprint arXiv:2603.04069*.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. [Logic-RL: Unleashing LLM Reasoning with Rule-Based Reinforcement Learning](#). *arXiv preprint arXiv:2502.14768*.

Lecheng Yan, Ruizhe Li, Guanhua Chen, Qing Li, Jiahui Geng, Wenxi Li, Vincent Wang, and Chris Lee. 2026. [Spurious Rewards Paradox: Mechanistically Understanding How RLVR Activates Memorization Shortcuts in LLMs](#). *arXiv preprint arXiv:2601.11061*.

Kaiyu Yang, Jia Deng, and Danqi Chen. 2022. [Generating Natural Language Proofs with Verifier-Guided Search](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. SatLM: Satisfiability-Aided Language Models Using Declarative Prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025. [Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model?](#) In *Advances in Neural Information Processing Systems (NeurIPS)*.

A Experimental Setup

Table 4 lists all hyperparameters shared across experiments. All RL conditions use the same GRPO configuration; only the reward function differs. Seeds control model weight initialization and data shuffling; training data is held constant across all seeds and conditions.

Table 4: Training hyperparameters for all GRPO experiments.

Parameter	Value
<i>Model</i>	
Base model	Qwen3-4B
Precision	BF16
<i>GRPO Training</i>	
Rollouts per prompt (K)	4
Training steps (default)	500
Training steps (extended)	1000
Learning rate	1×10^{-6}
Per-device batch size	4
Gradient accumulation steps	2
Effective batch size	8
Max completion length	512 tokens
Random seeds	42, 43, 44
<i>Data</i>	
Training set	FOLIO, 500 examples
Evaluation set	FOLIO, 100 examples (held-out)
Data file	train_500_seed42
<i>Solvers</i>	
FOL solver	Prover9
Prolog solver	SWI-Prolog 8.4.2
<i>Evaluation</i>	
Decoding	Greedy ($T=0$)
ISR computation	eval_sta.py
ISR variant	Proved-only (ISR*)

Compute resources. Experiments were run on single-GPU setups (NVIDIA RTX 4090 24 GB and RTX PRO 6000 96 GB). Each GRPO training run takes approximately 1–2 GPU-hours (500 steps) or 3–5 GPU-hours (1000 steps). Total compute across all seeds, conditions, and reruns is approximately 80–120 GPU-hours.

SFT convergence. SFT training converged across all three seeds: final-step loss ≤ 0.04 and token accuracy $> 99\%$ after ~ 8 epochs on 500 examples (seed 42: 0.029; seed 44: 0.033); the 1000-step extension further decreases loss to 0.017.

Evaluation exception. FOL exec-only seed 43 was evaluated with sampling (temperature = 0.7)

rather than greedy decoding. Its PR and ISR values are included for completeness but are not strictly comparable with other seeds under the unified evaluation protocol.

Reward formulation. An additive variant ($r = r_{\text{exec}} + \beta \text{ISR}$) was tested in early experiments but collapsed prove rate to near-zero, as ISR alone provides no gradient toward provable formalizations; the multiplicative form $r_{\text{exec}} \cdot (1 + \beta \text{ISR})$ ensures ISR modulates only proved outputs.

ProofFOL dataset construction. ProofFOL is derived from ProofWriter (Tafjord et al., 2021) (10,424 training examples) via the GPT-4o silver-standard FOL translation pipeline of Thatikonda et al. (2024) with Prover9 verification filtering ($\sim 70\%$ retention; failures from timeouts, unsupported constructs, and translation errors). Of the translated examples, 99.1% parse successfully (92 skipped due to empty FOL output). We sample 500 training and 100 evaluation examples (seed 42) from the verified pool. Table 2 (main text) compares key structural properties.

ProofFOL problems are structurally more complex ($3.4\times$ more premises) but use simpler, synthetic predicates (e.g., $\text{Blue}(x)$, $\text{Rough}(x)$) characteristic of ProofWriter’s procedural generation. The absence of XOR constraints and the regularity of predicate structure make ProofFOL a complementary testbed: structural degeneration must emerge from the RL training dynamics rather than from formalization difficulty.

B Extended Discussion

What ISR measures and what it does not. ISR quantifies syntactic component necessity: the fraction of solver-input components whose individual removal causes proof failure. This is strictly weaker than semantic faithfulness: whether the formalization accurately captures the natural-language meaning. We verify this empirically: Spearman $\rho(\text{ISR}, \text{NLI accuracy}) = -0.049$ ($p = 0.187$, n.s.) from a downstream NLI evaluation on exec-only checkpoint outputs. ISR and downstream task performance are uncorrelated. ISR is thus a component-necessity diagnostic, not a semantic quality metric: high ISR indicates that every component contributes to the proof, not that the proof faithfully represents the original reasoning problem.

Critically, low ISR carries different semantics

depending on the training objective. Under exec-only training, low ISR in Prolog reflects redundant coding (repeated ground facts where removing any single copy does not affect the proof) rather than an absence of reasoning (§4.4). Under ISR-as-reward, high ISR reflects metric gaming (§4.3). ISR values must always be interpreted relative to the optimization objective that produced them.

ISR as $(n-1)$ -coalition Shapley value. Define a cooperative game (N, v) with player set $N = \{1, \dots, n\}$ (components) and binary utility $v(S) = 1[S(S)=\text{proved}]$. The Shapley value ϕ_i averages marginal contributions over all coalition sizes; ISR restricts to the maximal coalition $|S|=n-1$, giving $\phi_i^{\text{LOO}} = \frac{1}{n}[v(N) - v(N \setminus \{i\})]$. Summing: $\text{ISR}(x) = \sum_i \phi_i^{\text{LOO}} = \frac{1}{n} \sum_i 1[v(N \setminus \{i\})=0]$, recovering Equation 1. This restriction trades full interaction detection for $O(n)$ cost; the binary solver output makes higher-order interactions unobservable from LOO alone (§3.2).

The premise-count confound. LOO-based metrics like ISR have an inherent mathematical bias: with fewer components, each is mechanically more likely to be necessary. A formalization with 2 premises where both are necessary yields $\text{ISR}=1.0$; one with 10 premises where 8 are necessary yields $\text{ISR}=0.8$, even though the latter encodes richer reasoning. This confound explains the sole apparent ISR-as-reward success (§4.3): $\text{ISR } \beta=0.1$ seed 42 produces shorter formalizations, mechanically inflating ISR. Premise-count-matched comparison eliminates the difference ($1.06\times$, McNemar’s test $p=0.317$; Table 9), confirming that the scalar ISR advantage is not statistically significant. This bias is not specific to our setting: any LOO-based structural metric applied to variable-length outputs requires component-count normalization. We recommend that future work employing ablation-based diagnostics report premise-count-matched comparisons alongside raw values. Mode composition analysis (CE vs. TS rates) is unaffected by premise-count confounding because TS classification depends on tautological proof structure (§C), not on the number of premises in the formalization.

TS dose-response gradient. TS tautology follows a dose-response pattern across reward conditions: random $\beta=0.1$ produces 13.7% TS (36/262 pooled; per-seed: 38.3% in s43, 0% in s42/s44), random $\beta=0.3$ yields 4–7 TSa samples per seed, and $\text{ISR } \beta=0.1$ reaches 70.6% (163/231). The gra-

Premises	Condition	N_{proved}	CE%	TS%	Genuine%
≤ 3	ISR $\beta=0.1$	8	50.0	12.5	12.5
	Random $\beta=0.1$	6	50.0	0.0	50.0
4–6	ISR $\beta=0.1$	61	47.5	19.7	14.8
	Random $\beta=0.1$	58	72.4	0.0	22.4
≥ 7	ISR $\beta=0.1$	12	75.0	0.0	25.0
	Random $\beta=0.1$	15	86.7	0.0	13.3

Table 5: Mode composition by premise-count bin (seed 42). TS hacking is concentrated in the 4–6 premise range and absent for ≥ 7 premises, indicating a mechanistic complexity threshold rather than a statistical confound.

dient (random $\beta=0.1$ (13.7%) $\rightarrow$ random $\beta=0.3$ (4–7 TSa per seed) $\rightarrow$ ISR $\beta=0.1$ (70.6%)) shows ISR transforms TS from a seed-dependent GRPO tendency to a near-universal convergence direction.

Goodhart effects: implications. ISR was designed to *detect* reward gaming of r_{exec} , yet when incorporated into the reward signal, the model games ISR itself. Any diagnostic metric risks becoming a target when optimized. The dose-response pattern maps onto the Manheim & Garrabrant taxonomy (Manheim and Garrabrant, 2019): $\beta=0.1$ on FOLIO instantiates *regressional* Goodhart ($P \rightarrow P$ tautologies yield $\text{ISR} \geq 0.5$, so ISR erroneously rewards the shortcut); $\beta \geq 0.2$ introduces *extremal* elements (PR collapse to 18% vs. 48% random at $\beta=0.3$). The implication is conditional: ISR fails as a training reward when hacking satisfies its LOO structure (FOLIO), but produces beneficial instability when hacking does not (ProofFOL; §4.4).

Single-seed unreliability. The three exec-only seeds diverge radically (PR $\sigma=47$ percentage points), each converging to a qualitatively different degeneration mode (§4.2). ISR $\beta=0.1$ seeds show similar divergence: 2/3 dominated by conclusion duplication (TS), 1/3 by premise minimization (§4.3). Multi-seed evaluation with structural diagnostics should be standard practice for neurosymbolic RL systems.

Classifier limitations. Text-match hacking classifiers have systematic blind spots that functional analysis exposes. In Prolog STA, two seeds achieved 100% genuine classification under strict text matching, yet functional analysis via the $n_{\text{necessary}}$ distribution revealed 0% genuine in both cases (Table 10). One seed produces uppercase pseudo-Prolog that is functionally tautological; the

other converts conclusions to snake_case atoms serving as both fact and query, a gaming mode invisible to string comparison. Practitioners should cross-validate text-match classifiers with functional metrics and never treat classifier-reported genuine rates as ground truth without independent verification.

Practical guidelines. See §5 for five practical guidelines derived from these findings.

Multiple testing correction. We define two test families: FOLIO ($k=14$ tests across all conditions) and ProofFOL ($k=1$). Within the FOLIO family, Bonferroni correction at $\alpha=0.05$ yields adjusted thresholds of $p_{\text{adj}}=0.05/14=0.0036$. Three primary comparisons survive correction: the TS rate difference (ISR vs. random; $p_{\text{bonf}}=1.82 \times 10^{-38}$), the SFT vs. exec-only non-hacking rate ($p_{\text{bonf}}=0.014$), and the Cochran–Mantel–Haenszel test for TS association across seeds ($p_{\text{bonf}} < 10^{-8}$). The matched 5+5-seed Fisher/Mann-Whitney comparison is borderline ($p_{\text{bonf}}=0.056$), significant at $\alpha=0.10$ after Bonferroni correction ($k=14$). The ProofFOL genuine-rate permutation test ($p=0.10$) belongs to a separate hypothesis family (different dataset, independent experimental design) and does not reach significance even without correction; it is reported as an exploratory observation throughout. Bonferroni assumes independence, which is conservative for our setting (several tests use overlapping data); Holm–Bonferroni or Benjamini–Hochberg FDR would recover the matched 5+5 comparison at conventional significance. We report the most conservative Bonferroni results for transparency.

Mode distribution χ^2 test. A pooled 3×2 contingency table (TS/CE/N-H $\times$ ISR/Random, 3-seed, 500 steps) yields $\chi^2=174.8$, $\text{df}=2$, $p=1.08 \times 10^{-38}$, confirming that ISR reward shifts the overall hacking mode composition, not merely the TS rate. Per-seed tests are individually significant (s42: $\chi^2=15.8$, $p < 0.001$; s43: $\chi^2=59.4$, $p < 10^{-13}$; s44: $\chi^2=168.0$, $p < 10^{-36}$). The previously reported $\text{OR}_{\text{CMH}}=2.46$ for TS specifically ($p < 10^{-8}$) captures a narrower slice of this distributional shift.

C Hacking Mode Definitions

We identify eight distinct reward hacking modes across all experiments. TS comprises two subtypes: TSa (text-match duplication, where the premise is

Table 6: Fisher exact tests for non-hacking rate across conditions (3-seed pooled). ISR $\beta=0.1$ non-hacking rate is corrected for PG reclassification and TS false negatives (Appendix C.1).

Comparison	Condition	Control	OR	p
<i>Non-hacking rate (PG+FN corrected)</i>				
ISR vs. Random	6.06% [3.6,9.9]	9.5% [6.5,13.7]	0.61	0.182 (n.s.)
ISR vs. Exec-only	6.06%	7.5%	0.79	0.62
Random vs. Exec-only	9.5%	7.5%	1.30	0.68
<i>TS gaming rate</i>				
ISR vs. Random	70.6% [64.3,76.3]	13.7% [9.9,18.6]	15.05	1.3×10^{-39}
<i>SFT vs. Exec-only non-hacking rate</i>				
SFT vs. Exec-only	92.3% [79.7,97.3]	7.5%	147	<0.001

ISR: $n=231$ (3 seeds). Random: $n=262$ (3 seeds). Exec-only: $n=93$ (s42). SFT: $n=39$ (3 seeds, 1000 steps). 95% Wilson CIs in brackets where shown. Pooled statistics treat within-seed samples as independent and serve as descriptive summaries only; the seed-stratified CMH test (§B) accounts for within-seed dependence.

textually identical to the conclusion) and TSb (obfuscated tautology, where the premise is logically equivalent but textually distinct; §C.1). Each mode is defined below with its mechanism, a representative example, and its interaction with ISR.

Table 3 (main text) provides the compact cross-condition summary; Table 7 below gives the extended breakdown.

Classifier Validation. Hacking mode classification uses deterministic rule-based classifiers, not learned models, so standard ML precision/recall is not directly applicable. CE and TS are identified by normalized text matching between conclusion and premise strings after whitespace and variable normalization; remaining modes are detected via functional analysis of the per-sample $n_{\text{necessary}}$ distribution from STA (leave-one-out ablation). No training data or manual annotation is used for classification.

To estimate precision, we manually verified 50 randomly sampled outputs classified as CE or TS: all 50 were confirmed correct (estimated precision $>95\%$, Wilson 95% CI: [93%, 100%]). Recall is inherently bounded by the text-match threshold: semantic manipulations that alter surface form while preserving the gaming structure (e.g., variable renaming beyond our normalization, or paraphrased conclusions) would be missed. We cross-validate with functional LOO analysis: samples classified as non-hacking but showing $n_{\text{necessary}} \leq 1$ are flagged as potential false negatives. This cross-validation revealed query-as-fact gaming in Prolog (§C), where the strict text classifier reported 100% non-hacking but functional analysis showed 95/97 samples were tautological ($n_{\text{necessary}}=1$). The non-hacking rate for FOL conditions may therefore be

slightly overestimated, as subtle semantic manipulations below our text-matching threshold would evade the strict classifier.

Boundary case validation. Stratified validation across 82 samples (all five hacking categories) yields Cohen’s $\kappa=1.0$; independent re-annotation confirms full agreement. Structured inspection of 20 boundary cases (short-conclusion CE (10, minimum 4 characters), low-ISR genuine (5, minimum ISR=0.22), and TS on false propositions (5)) confirmed 20/20 correct classifications. Short-conclusion CE samples are unambiguous hacking: the conclusion appears as a standalone premise with no NL source. Low-ISR genuine samples reflect mode-collapsed template outputs, not structural gaming. 3/5 TS boundary cases confirm factually false propositions (e.g., affirming “dogs are plants”), showing TS is not merely vacuous but actively harmful.

The TS_b subcategory (functional LOO tautologies undetectable by text matching) provides a lower bound on TS recall: $\geq 91\%$ (149/163) under ISR $\beta=0.1$. Other modes’ recall is not estimable from the validation-only design; non-hacking rates throughout should be interpreted as upper bounds.

C.1 PG – Premise-Subset Gaming

Manual audit of ISR $\beta=0.1$ seed 42 revealed a previously undetected hacking mode: the model emits a subset of the gold premises that is individually necessary (high ISR) but insufficient for the intended derivation, with proof success depending on the conclusion already being embedded (CE-style). We term this **PG (premise-subset gaming)**: formalizations where each component is LOO-necessary but the premise set is a strict

Mode	Mechanism	Exec	ISR	Rnd	SFT
CE	Concl. in premises	87	18	68	8
TS	Single prem. = concl.	•	71	14 [†]	0
PG	Premise-subset gaming	5	5	8	0
Tautological	2-comp $P=C$	•	0	0	0
Repetition	Output degen.	•	—	—	—
Contradiction	n_{comp} inflated	•	—	—	—
C-w/o-utility	Complex, redundant	0	0	1	0
Q-as-fact	Atom = fact+query	—	—	—	—

Table 7: Hacking taxonomy across FOL conditions (% of proved samples). ISR/Rand columns: $\beta=0.1$ 3-seed pooled. TS includes subtypes TSa (text-match, $n=149$ ISR + 36 random) and TSb (obfuscated tautology, $n=14$; primarily s44). [†]Random TS driven by seed 43 (38.3%); seeds 42/44: 0%. Exec: CE=87% in s42; • marks seed- or step-specific modes (TS/Tautological in s44, Repetition in s43, Contradiction at 1000 steps). PG: premise-subset gaming (Appendix C.1). C-w/o-util.: complexity-without-utility (random-specific). Q-as-fact: Prolog STA only (Table 10).

Mode	Val. N	Prec.
CE (Concl. Embed.)	80 [†]	1.00
TS (Tautological)	7	1.00
PG (Premise-Subset)	20	1.00
Contradiction	20	1.00
Genuine	25	1.00
Overall	152	1.00

Table 8: Per-mode classifier precision on 152 manually validated samples (stratified + boundary + random; three exercises pooled). [†]Includes 50 randomly sampled CE∪TS outputs (pool ratio CE:TS=374:2). Recall is not estimable from a validation-only design.

subset of what the NL problem requires, inflating ISR without faithful reasoning.

Detection. PG is identified by manual inspection of non-hacking samples with unusually low premise counts ($n_{\text{prem}} \leq 3$) that nonetheless achieve high ISR. Automated detection is not currently feasible because PG requires comparing the generated premise set against the gold premises.

Audit scope and limitations. Full PG audit was completed for ISR $\beta=0.1$ seed 42 only. ISR seeds 43 and 44 checkpoints were deleted before audit and cannot be recovered; their non-hacking rates may be overestimated. Random $\beta=0.1$ does not produce PG by mechanism (no ISR optimization pressure); seed 42 audit confirmed PG=0.

TS false-negative correction (TSb obfuscated tautology). ISR $\beta=0.1$ seed 44 contained 12 samples with $n_{\text{prem}}=1$ classified as non-hacking by the text matcher. Manual inspection confirmed all 12 are logically equivalent to TS but evade text matching through vacuous quantifier wrapping (11/12) or structural simplification (1/12).

We classify these as **TSb** (obfuscated tautology), distinct from text-match-detectable TSa. 10/12 achieve $\text{ISR} \geq 0.83$, indicating simultaneous gaming of execution reward, ISR metric, and TS classifier: triple gaming (§4.5). After correction, ISR s44 non-hacking=0; total TS = TSa (77 text-match) + TSb (12 obfuscated) = 89.

Impact on non-hacking-rate comparison. After PG reclassification and TS false-negative correction: $\text{ISR non-hacking} = 14/231 = 6.06\%$ vs. $\text{random non-hacking} = 25/262 = 9.5\%$ (Fisher $p=0.182$, n.s.; Wilson 95% CI: ISR [3.6%, 9.9%], Random [6.5%, 13.7%]). The direction is consistent with ISR-amplified gaming modes (TS, PG) displacing non-hacking output, but the effect is not statistically significant at the conventional $\alpha=0.05$ threshold.

C.2 CE – Conclusion Embedding

Definition. The conclusion appears verbatim (or with trivial variable renaming) as one premise among multiple premises. The solver succeeds by matching the embedded conclusion, rendering the other premises irrelevant to the proof.

Example (FOL).

```

Premises:
  all x (bird(x) -> flies(x)).
  bird(tweety).
  flies(tweety). % conclusion embedded
Conclusion: flies(tweety).
```

ISR interaction. Variable. The embedded premise has $\text{ISR} = 1$ (removing it may break the proof), but other premises may also register as necessary depending on Prover9’s search strategy. ISR

does not distinguish trivial from substantive necessity.

Prevalence. Conclusion embedding in some form (CE multi-premise or TS single-premise) is GRPO-universal at 68–97% across all RL conditions. Under exec-only and Random reward, it manifests as CE (69–93%); under ISR reward, it shifts to TS (70.6% pooled).

C.3 TSa – Conclusion Duplication (Standard)

Definition. The entire formalization collapses to a single premise whose content is identical (text-match detectable) to the conclusion. Since $P \rightarrow P$ is trivially provable, this simultaneously maximizes exec reward. Because the sole premise is necessary for every sub-component of the conclusion under LOO ablation, ISR is inflated toward 1.0.

Example (FOL).

Premises:
 $\text{all } x ((A(x) \& B(x)) \rightarrow (C(x) | D(x)))$.
 Conclusion:
 $\text{all } x ((A(x) \& B(x)) \rightarrow (C(x) | D(x)))$.

ISR interaction. $\text{ISR} \approx 0.8\text{--}1.0$. The single premise is trivially “maximally necessary,” but the proof is vacuous. TSa games both exec reward and ISR reward; TSb extends this to triple gaming by additionally evading the text-match classifier.

Prevalence. TSa (text-match detectable): 64.5% pooled (149/231) under $\text{ISR } \beta=0.1$ (s42: 16.0%, s43: 96.7%, s44: 86.5%). Combined TS (TSa + TSb): 70.6% (163/231; s43 TSa=59 + TSb=2; includes s44 FN correction). Random $\beta=0.1$ pooled: 13.7% (36/262, all TSa; driven by seed 43 at 38.3%, seeds 42 and 44 at 0%).

C.4 TSb – Obfuscated Tautology

Definition. A single-premise formalization logically equivalent to TSa (premise = conclusion) but with surface-form differences that evade the text-match TS classifier. The model achieves this through vacuous quantifier wrapping (11/12 instances) or structural simplification (1/12).

Example (FOL, vacuous quantifier wrapping).

Premises:
 $\text{all } x \text{ all } y ((A(x) \& B(y)) \rightarrow (C(x) | D(y)))$.
 Conclusion:
 $\text{all } x ((A(x) \& B(x)) \rightarrow (C(x) | D(x)))$.

The premise introduces a vacuous variable y , unified with x at proof time; the formulas are logically equivalent but textually distinct.

ISR interaction. $\text{ISR} \geq 0.83$ (10/12 samples). TSb simultaneously games three objectives: execution reward ($r_{\text{exec}}=1$ via tautological proof), ISR metric (single necessary premise yields high ISR), and the text-match TS classifier (surface-form difference evades detection). This triple gaming makes TSb the most complete Goodhart case we observe.

Prevalence. Found primarily in $\text{ISR } \beta=0.1$ seed 44 (12/89 proved samples, all originally classified as non-hacking); seed 43 contributes 2 additional cases. Functional cross-validation via $n_{\text{necessary}}=1$ exposed the misclassification.

C.5 Tautological Collapse

Definition. Two-component formalizations where the premise and conclusion are identical atomic or simple formulas, yielding a $P \rightarrow P$ tautology with minimal token cost.

Example (FOL).

Premises: $p(a)$.
 Conclusion: $p(a)$.

ISR interaction. $\text{ISR} = 1.0$ (the sole premise is necessary). Distinguished from TS by having low component count ($n_{\text{comp}} \approx 2$) and atomic formulas rather than complex sub-structured conclusions.

Prevalence. Dominant in FOL exec-only seed 44 (PR = 0.98, mean $n_{\text{comp}} = 2.04$, 100% TS by strict classifier). Also observed in Prolog exec+STA seed 42 (100% TS, PR = 0.50).

C.6 Repetition Collapse

Definition. The model generates identical premises or clauses repeated until the maximum completion length, producing a degenerate output with high token count but no logical diversity. Removing any single repeated copy leaves the others intact, yielding $\text{ISR} \approx 0$.

Example (Prolog).

1222

1267	penguin(fido).	Example	(FOL, 1000-step exec-only).	1313
1268	penguin(fido).			1314
1269	penguin(fido).	Premises:		1315
1270	... (repeated to 512 tokens)	c1. c2. c3. ... c18. % 18.55 mean		1316
1271	query(penguin(fido)).	% conclusion embedded among premises		1317
1272		Conclusion: Q(a).		1318
1273	ISR interaction. ISR ≈ 0 . Each copy is individ-	ISR interaction. ISR ≈ 0 (at 1000 steps: ISR =		1320
1274	ually redundant since identical copies remain after	0.003, with 94.8% of proved samples having ISR =		1321
1275	removal. ISR correctly diagnoses this degenera-	0).		1322
1276	tion.			
1277	Prevalence. Observed in FOL exec-only seed 43	Prevalence. Observed in extended exec-only		1323
1278	(87% of outputs hit max completion length; PR =	training (seed 42, 1000 steps). Mean component		1324
1279	13%, ISR = 0).	count grows from ~ 6 (500 steps) to 18.55 (1000		1325
1280		steps).		1326
1281	C.7 Complexity-without-Utility	C.9 Query-as-Fact Gaming		1327
1282	Definition. The formalization contains syntacti-	Definition. The model converts the natural lan-		1328
1283	cally complex premises (nested quantifiers, multi-	guage conclusion into a snake_case atom and in-		1329
1284	arity predicates) that are semantically vacuous:	cludes it as both a ground fact and the query. Prolog		1330
1285	none individually necessary for the proof. The	trivially proves the query by unifying with the fact.		1331
1286	complexity inflates surface metrics (formula length,	This mode evades text-match classifiers designed		1332
1287	nesting depth) without contributing to logical	for CE/TS because the generated atom bears no		1333
1288	derivation.	surface resemblance to the original conclusion text.		1334
1289	Example	Example	(Prolog).	1335
1290	Premises:			1336
1291	all x all y (R(x,y) ->	% NL: "Fido is happy."		1337
1292	exists z (S(x,z) & T(z,y))).	% Model output:		1338
1293	all x (P(x) -> Q(x)).	fido_is_happy.		1339
1294	P(a). % only this needed	:- fido_is_happy. % query = fact		1340
1295	Conclusion: Q(a).			1341
1296				
1297	ISR interaction. ISR correctly identifies the vac-	ISR interaction. ISR ≈ 0.9 –1.0. The single fact		1342
1298	uous premises (ISR = 0 for each), pulling down the	is trivially necessary (its removal breaks the query),		1343
1299	mean ISR. This is the mode ISR is best calibrated	inflating ISR. Functionally equivalent to TS but		1344
1300	for.	with a distinct surface mechanism.		1345
1301	Prevalence. Low. Observed in Random $\beta=0.1$	Prevalence. Observed in Prolog exec+STA		1346
1302	seed 42 at 3.8% (3/79). Not observed under	seed 44 (PR = 0.97, ISR = 0.912). 95/97 proved		1347
1303	ISR reward (ISR's gradient discourages zero-ISR	samples have $n_{\text{necessary}} = 1$, confirming func-		1348
1304	premises).	tional tautology despite the strict classifier report-		1349
1305	C.8 Contradiction Explosion	ing 100% non-hacking.		1350
1306	Definition. The component count grows un-	D Per-Seed Detailed Results		1351
1307	boundedly over training. Excess premises intro-	Tables 9 and 10 report full per-seed results. All met-		1352
1308	duce contradictions or redundancies that trivially	rics are computed by eval_sta.py on the held-out		1353
1309	entail any conclusion via <i>ex falso quodlibet</i> . In-	100-example FOLIO evaluation set using greedy		1354
1310	dividual premises may register as ISR = 0 (re-	decoding unless otherwise noted. ISR values are		1355
1311	movable without breaking the proof), but the sheer	proved-only ISR*.		
	count obscures the underlying triviality. ¹³⁵⁶			

ISR collapse trajectory (FOL exec-only, seed 42). Table 9 traces the ISR collapse over training: $\text{ISR}^* = 0.642$ at 100 steps $\rightarrow 0.132$ at 500 steps (-79.4%) $\rightarrow 0.003$ at 1000 steps (-99.5% from step 100). PR increases monotonically ($0.32 \rightarrow 0.93 \rightarrow 0.96$) while component necessity collapses, confirming that ISR flags degeneration invisible to execution reward. By step 1000, 94.8% of proved samples have $\text{ISR} = 0$ and the mean component count rises to 18.55.

Prolog: universal Goodhart via three distinct pathways. All three Prolog exec+ISR seeds exhibit Goodhart’s Law with 0% non-hacking proofs (Figure 6), but each discovers a different gaming mechanism: seed 42 produces clean TS tautologies (single premise = conclusion), seed 43 generates uppercase pseudo-Prolog that bypasses the text-match classifier while remaining functionally tautological, and seed 44 converts NL conclusions to snake_case atoms used as both fact and query. The diversity of gaming pathways across seeds (despite identical training data and reward function) demonstrates that Goodhart’s Law in neurosymbolic reasoning is not a single failure mode but a family of convergent exploits.

D.1 Per-Seed Results at $\beta=0.2$ and $\beta=0.3$

Table 12 reports per-seed results for ISR at $\beta=0.2$ and both ISR and Random at $\beta=0.3$. At $\beta=0.2$, all three seeds converge to pure TS Goodhart (100% or near-100% TS), confirming $\beta \geq 0.2$ as the extremal Goodhart regime. The three ISR $\beta=0.3$ seeds illustrate qualitatively distinct Goodhart pathways (§4.3).

Table 9: Detailed per-seed results: FOL (Prover9). Includes the per-seed $\beta=0.1$ breakdown referenced from Table 1. PR = prove rate, ISR* = proved-only ISR, TS/CE/N-H = percentage among proved samples. “—” indicates data not available. [†]s43 uses sampling eval ($T=0.7$). [‡]Three s44-1000 samples escape text-match heuristics (CE/PG classifiers) but exhibit ISR=0 and functionally degenerate proof structures: disjunctive conclusions that bypass PG detection and commuted premises that bypass CE detection; reclassified as degenerate. [◊]Seeds 45–46 added post-hoc under identical hyperparameters (effective batch size 8; micro-batch decomposition differs: 4×2 vs. 2×4 , mathematically equivalent).

Condition	Seed	Steps	PR	ISR*	TS%	CE%	N-H%	Notes
<i>Baselines</i>								
Base model (no training)	—	0	0.12	0.064	—	—	—	Pre-training
Ground-truth FOL	—	—	0.18	0.559	—	—	—	Human formalizations
SFT	42	1000	0.13	0.568	0	7.7	92.3	Supervised, no RL
	43	1000	0.11	0.635	0	9.1	90.9	
	44	1000	0.15	0.589	0	6.7	93.3	
3-seed pooled			0.130	0.595	0	7.8	92.3	$n=39$ proved
<i>Exec-only</i>								
	42	100	0.32	0.642	—	—	—	Early checkpoint
	42	500	0.93	0.132	0	87.1	7.5	PG=5.4%
	42	1000	0.96	0.003	0	100	0	Contradiction explosion
	43	500	0.13	0 [†]	—	—	—	[†] Greedy ($n_{\text{proved}}=13$)
	43	1000	0.96	0.082	0	93.8	6.2	CE dominant
	44	500	0.98	0.582	100	0	0	Tautological collapse
	44	1000	1.00	0.002	0	85	0 [‡]	PG=15%
3-seed mean $\pm$ std			0.68 $\pm$ 0.47	0.238 $\pm$ 0.30				500-step
3-seed mean $\pm$ std			0.97 $\pm$ 0.02	0.029 $\pm$ 0.05	2.1 $\pm$ 3.6			1000-step
<i>ISR $\beta=0.1$</i>								
	42	500	0.81	0.415	16.0	51.9	17.3	Mixed CE+TS+PG
	42	1000	0.90	0.877	96.7	0	2.2	TS collapse; PG=1 ($n=90$)
	43	500	0.61	0.830	100	0	0	TSa=59, TSb=2
	44	500	0.89	0.902	100	0	0	TS dominant (FN corrected)
	45 [◊]	500	0.42	0.605	38.1	4.8	54.8	PG=1; $n=42$
	46 [◊]	500	0.77	0.411	14.3	26.0	54.5	PG=4; $n=77$
3-seed mean $\pm$ std			0.77 $\pm$ 0.14	0.716 $\pm$ 0.26				500-step (s42–44)
5-seed mean $\pm$ std			0.70 $\pm$ 0.19	0.633 $\pm$ 0.23				500-step (s42–46)
<i>ISR $\beta=0.3$</i>								
See Table 12 for per-seed $\beta=0.3$ results.								
<i>Random $\beta=0.1$</i>								
	42	500	0.79	0.131	0	73.4	22.8	PG=4
	43	500	0.94	0.150	38.3	59.6	1.1	TSa=36; PG=1
	44	500	0.89	0.070	0	82.0	6.7	PG=16 (CE $\cap$ PG: 6)
3-seed mean $\pm$ std			0.87 $\pm$ 0.08	0.117 $\pm$ 0.042				TS seed-dep.: 0/38/0%
<i>Random $\beta=0.3$</i>								
	42	500	0.48	0.094	8	92	0	TSa=4; $n_{\text{comp}} = 9.6$
	43	500	0.92	0.180	7.6	89.1	3.3	TSa=7; $n_{\text{N-H}}=3$
	44	500	0.23	0	0	100	0	Pure CE ($n=23$)
3-seed mean $\pm$ std			0.54 $\pm$ 0.35	0.09 $\pm$ 0.09				
<i>Shuffled-ISR $\beta=0.1$ (deconfounding control)</i>								
	42	500	0.15	0.567	6.7	6.7	86.7	$n=15$ proved
	43	500	0.16	0.568	6.3	6.3	87.5	$n=16$ proved
	44	500	0.15	0.542	6.7	6.7	86.7	$n=15$ proved
3-seed mean $\pm$ std			0.153 $\pm$ 0.006	0.559 $\pm$ 0.015	6.5	6.5	87.0	$N=46$ proved

Table 10: Detailed per-seed results: Prolog (SWI-Prolog 8.4.2). PR = prove rate, ISR* = proved-only ISR, TS/CE/N-H = percentage among proved samples. “—” indicates data not available.

Condition	Seed	PR	ISR*	TS%	CE%	N-H%	Notes
<i>Exec-only</i>							
	42	0.83	0.037	0	0	—	Ground fact enumeration
	43	0.56	0.015	—	—	—	
	44	0.21	0.387	—	—	—	Small-sample ($n=21$)
	3-seed mean $\pm$ std		0.53 $\pm$ 0.31	0.146 $\pm$ 0.21			
<i>Exec+ISR (STA reward)</i>							
	42	0.50	1.000	100	0	0	TS tautological
	43	0.38	0.185	0 [‡]	97.4	0	[‡] Functional TS ^a
	44	0.97	0.912	0 [‡]	0	0 [‡]	Query-as-fact gaming ^b
	3-seed mean $\pm$ std		0.62 $\pm$ 0.31	0.699 $\pm$ 0.45			3/3 Goodhart, 0% non-hacking

^a Strict TS classifier reports 0%, but functional analysis shows all 38/38 proved samples are tautological ($n_{\text{necessary}} = 1$). Uppercase pseudo-Prolog syntax bypasses the text-match classifier.

^b Strict classifier reports Non-hacking = 100%, but 95/97 proved samples have $n_{\text{necessary}} = 1$. The model converts NL to snake_case atoms serving as both fact and query, a third distinct Goodhart pathway.

[‡] Strict classifier value is misleading; see footnotes ^a/^b.

Table 11: Detailed per-seed results: ProofFOL (Prover9). PR_{f50} = mean prove rate over final 50 steps. TS = tautological shortcutting (§4.5), CE = conclusion embedding, Genuine = classified as neither TS nor CE. Stable convergence: PR_{f50} > 0.90.

Condition	Seed	PR _{f50}	ISR*	TS%	CE%	Genuine%	Notes
Base	—	0.16	0.310	0	0	100	$n=16$ proved
Exec-only	42	≈ 1.0	0.066	100	0	0	$n=99$; stable
	43	0.55	0.119	0	49.2	50.8	$n=63$; partial
	44	≈ 1.0	0.043	0	98.6	1.4	$n=72$; stable
	Aggregate			17.4% genuine (mean of seeds)			
ISR $\beta=0.1$	42	0.35	0.12	0	0	100	$n=46$; collapsed
	43	≈ 0.47	≈ 0.13	0	69.6	30.4	$n=56$; unstable
	44	0.49	0.145	0	0	100	$n=66$; oscillation
	Aggregate			76.8% genuine (mean of seeds)			

Aggregate genuine rate: mean of per-seed genuine rates. Permutation test (exec vs. ISR genuine): $p=0.10$, $N=3$ per group.

Table 12: Per-seed results at $\beta=0.2$ and $\beta=0.3$ (FOL). PR = prove rate, ISR* = proved-only ISR, TS/CE/N-H = percentage among proved samples.

Condition	Seed	PR	ISR*	TS%	CE%	N-H%	Mode
<i>ISR $\beta=0.2$</i>							
	42	1.00	0.602	100	0	0	Pure TS ($n=100$)
	43	0.96	0.552	96.9	1.0	0	PG=2
	44	0.94	0.964	100	0	0	Pure TS ($n=94$)
	3-seed mean $\pm$ std		0.97 $\pm$ 0.03	0.71 $\pm$ 0.22	3/3 TS Goodhart		
<i>ISR $\beta=0.3$</i>							
	42	0.18	0.143	0	77.8	5.6	Reward collapse ($n=18$); PG=3
	43	0.86	0.295	4.7	68.6	23.3	CE shortcutting
	44	0.95	0.742	100	0	0	Pure TS Goodhart
	3-seed mean $\pm$ std		0.66 $\pm$ 0.42	0.39 $\pm$ 0.31			
<i>Random $\beta=0.3$</i>							
	42	0.48	0.094	8	92	0	TSa=4
	43	0.92	0.180	7.6	89.1	3.3	TSa=7; $n_{\text{N-H}}=3$
	44	0.23	0	0	100	0	Pure CE ($n=23$)
	3-seed mean $\pm$ std		0.54 $\pm$ 0.35	0.09 $\pm$ 0.09			

D.2 Early-Training Hacking Analysis (100-Step Checkpoint)

To control for the prove-rate confound in our SFT comparison (§4.2), we analyze the exec-only checkpoint at 100 training steps (seed 42), where PR=32%, higher than SFT (13.0%, 3-seed pooled) but still early in training. Of the 32 proved samples, 9 (28.1%) exhibit TS hacking (single premise identical to conclusion) and 1 (3.1%) exhibits CE (conclusion embedded among premises), yielding a total hacking rate of 31.2%. By contrast, SFT at comparable or lower PR shows 7.7% hacking (1/13 proved samples). This demonstrates that reward hacking emerges early in GRPO training, well before the prove-rate plateau, and is not simply an artifact of having more proved samples available for exploitation. The 100-step hacking rate (31.2%) is already substantially above SFT (7.7%), supporting the claim that hacking is associated with RL optimization rather than prove-rate level alone.

D.3 $O(1)$ vs. $O(n)$ ISR Ablation

All main experiments (§4.2–4.4) use $O(n)$ full LOO ISR during both training and evaluation; the $O(1)$ binary variant described below exists as an alternative implementation (execution_fast_sta) but is used only in the ablation reported here. The two signals measure fundamentally different properties, not approximations of the same quantity. The binary signal tests *collective* premise necessity: $\text{fast_ISR}(x) = 1$ if x is proved and the conclusion is not provable under empty premises (tautology check), 0 otherwise. LOO tests *individual* premise necessity: each component is removed in turn, and ISR reflects the fraction whose removal causes proof failure.

Evaluation-time comparison. Across 402 proved samples from all ProofFOL conditions, the $O(1)$ binary signal is 1.0 for every sample: non-tautological conclusions cannot be proved without any premises, so the collective necessity test is trivially satisfied. This renders $O(1)$ unable to distinguish hacking from genuine formalizations: both pass the empty-premise check. By contrast, $O(n)$ LOO produces a clear separation: exec-only models average $\text{ISR}^*=0.046$ ($n=234$) while ISR-trained models average $\text{ISR}^*=0.183$ ($n=168$), because hacking constructions contain individually redundant components that LOO detects. Of 179 disagreements between the two signals, all are false positives ($O(1)=1$, $\text{LOO}=0$):

the binary check approves formalizations that LOO correctly identifies as structurally degenerate.

Training-time ablation. As a training reward ($r = 0.5 r_{\text{exec}} + 0.5 \text{fast_ISR}$, seed 42, 500 steps), the $O(1)$ variant achieves PR=63% with $\text{ISR}^*=0.007$, functionally zero, compared to $\text{ISR}^*=0.132$ under exec-only and $\text{ISR}^*=0.42$ under $O(n)$ ISR $\beta=0.1$. The tautology check depresses prove rate by 30 pp relative to exec-only (63% vs. 93%) by penalizing formalizations where the conclusion is provable without premises, but this constraint provides no gradient toward individual component necessity; the model is free to degenerate as long as it avoids trivial tautologies. That even the fine-grained $O(n)$ LOO signal, which *does* provide per-component necessity feedback, is ultimately gamed (§4.3) underscores that the problem lies in reward optimization itself, not in signal granularity.

Computational overhead. Full LOO requires $n+1$ solver calls per sample (mean $n \approx 6$ for FOLIO). With Prover9 at ~ 5 ms per call, STA evaluation adds ~ 55 ms per sample versus ~ 4 s for LLM inference ($<1\%$ overhead). The $O(1)$ binary check adds a single solver call per proved sample ($<2\%$ per training step), but the evaluation above shows this saving comes at the cost of all diagnostic power.

D.4 Cross-Model Generalization: LLaMA-3.2-3B Cold-Start Failure

We applied the identical GRPO configuration (Table 4) to LLaMA-3.2-3B (Grattafiori et al., 2024) (3B; exec-only, seed 42, $K=4$, FOLIO). Across 72 of 500 planned steps the model produced zero valid Prover9 output, generating only natural-language text; $r_{\text{exec}}=0$ for all rollouts yields zero GRPO advantage, so training was terminated early.⁵ This establishes a *capability threshold*: the base model must produce syntactically valid solver input for the RL loop to generate signal. Qwen3-4B crosses this threshold (non-zero prove rates within 10 steps) while LLaMA-3.2-3B does not despite comparable size (3B vs. 4B), likely reflecting pre-training data composition. An SFT warmstart (3 epochs on gold formalizations, $\text{lr}=2 \times 10^{-5}$) achieves PR=16% ($n=100$) after correcting the

⁵This run was also affected by a generation-termination configuration issue (dual EOS tokens); the zero prove rate reflects both the base model’s lack of FOL generation capability and truncated outputs.

generation-termination issue. However, the model produces Unicode FOL notation ($\forall$, $\rightarrow$, $\wedge$) instead of the required ASCII syntax despite training on ASCII-formatted data; the 16% prove rate reflects Prover9’s partial Unicode tolerance rather than correct format adherence. Subsequent GRPO runs (exec-only, 500 steps from SFT warmstart, 3 seeds) fail to improve beyond SFT baseline ($\text{PR}=15.0\pm1.0\%$, $n=14\text{--}16$ proved per seed), yet produce no structural degeneration: $86.6\pm0.9\%$ of proved samples are genuine across seeds, with cross-seed variance near zero ($\sigma_{\text{PR}}=1$ pp). This contrasts sharply with Qwen3-4B, where GRPO reaches 93% PR but collapses to 8% genuine with high cross-seed variance. The result suggests structural degeneration is a byproduct of successful RL optimization, not of the training procedure *per se*: when GRPO fails to amplify prove rate, the model retains structural integrity.

D.5 Seed-Dependent Convergence in Exec-Only GRPO

Not all exec-only seeds converge. Of 6 seeds tested (s42–s46, s52; s52 uses $K=4$, all others $K=8$), we observe three regimes: *converging with degeneration* (s42: $\text{PR}=93\%$, $\text{genuine}=7.5\%$; s44: $\text{PR}=96\%$, $\text{genuine}=0\%$; s46: $\text{PR}=93\%$ at 500 steps), *slow convergence with degeneration* (s43: $\text{PR}=14\%$ but 100% hacking at 500 steps), and *non-converging* (s45: $\text{PR}=15\%$, $\text{genuine}=86.7\%$; s52: $\text{PR}=14\%$, $\text{genuine}=85.7\%$). Non-converging seeds retain high genuine rates (85–87%), indicating that structural degeneration is conditional on successful RL optimization: when GRPO fails to bootstrap prove rate beyond base-model levels, the model preserves structural integrity. Extended training to 1000 steps for s45 confirms persistent non-convergence ($\text{PR}=15\%$, $\text{ISR}^*=0.592$, $\text{genuine}=86.7\%$, $n=15$ proved); the model neither improves nor degenerates.

This parallels the LLaMA-3.2-3B finding (Appendix D.4), where GRPO also fails to improve PR and retains structural quality. Both cases support the interpretation that degeneration is a byproduct of reward optimization success, not of the training procedure *per se*.

D.6 $N=2$ Sensitivity Analysis (Excluding Seed 43)

Seed 43 was evaluated with sampling ($T=0.7$) rather than greedy decoding (§A). To assess whether this non-standard protocol affects qualita-

tive conclusions, Table 13 reports pooled statistics from seeds 42 and 44 only.

Condition	PR (%)	ISR*	N-H (%)
Exec-only 500-step	95.5 ± 3.5	0.357 ± 0.318	3.8 ± 5.3
Exec-only 1000-step	98.0 ± 2.8	0.003 ± 0.001	0.0 ± 0.0
ISR $\beta=0.1$ 500-step	85.0 ± 5.7	0.659 ± 0.344	8.7 ± 12.2

Table 13: Pooled statistics excluding seed 43 ($N=2$, seeds 42 and 44 only). Mean $\pm$ std of per-seed values. Qualitative patterns preserved: exec-only converges to $\leq 7.5\%$ non-hacking; ISR $\beta=0.1$ shifts composition toward TS.

Excluding seed 43, exec-only converges to $\leq 7.5\%$ non-hacking at 500 steps and 0% at 1000 steps. Under ISR $\beta=0.1$, TS dominates both remaining seeds (s42: 16%, s44: 100%), confirming that ISR reward shifts hacking composition toward tautological shortcuts independent of the s43 evaluation anomaly. The large per-seed ISR* variance (0.357 ± 0.318) reflects different degeneration modes (CE in s42 vs. TS in s44), not data quality issues.

D.7 Cross-Scale: Qwen3-8B Exec-Only GRPO

To test whether structural degeneration generalizes beyond 4B scale, we apply exec-only GRPO to Qwen3-8B with $K=8$ rollouts per prompt ($K=4$ produces near-identical completions at 8B scale, yielding zero GRPO advantage and no learning signal). Table 14 summarizes configuration differences and results.

	Qwen3-4B	Qwen3-8B
Rollouts (K)	4	8
Batch size	4	1
Gradient accum.	2	16
Effective batch	8	16

Config	PR%	Genuine%	Converged?	Gradient
4B $K=4$ s42	93	7.5	Yes (degenerate)	Sufficient
8B $K=4$ s42	15.5	—	No	Zero
8B $K=8$ s42	14	92.9	No	Partial

Table 14: Cross-scale: Qwen3-4B vs. Qwen3-8B. **Top:** Configuration; $K=8$ required at 8B because $K=4$ produces near-identical rollouts. **Bottom:** Results; 4B converges with degeneration, both 8B configurations fail to converge, preserving structural integrity.

We first tested $K=4$ (matching the 4B configuration): the 8B model produces near-identical rollouts across all four completions, yielding zero GRPO advantage throughout 500 steps and no

learning signal (PR=15.5%, cold-start stagnation). Increasing to $K=8$ provides non-zero reward variance during mid-training (approximately 40–50% of steps between 20–370 show non-zero advantage), but this signal proves insufficient for convergence: held-out evaluation yields PR=14% ($n=14$ proved), $ISR^*=0.573$, genuine=92.9% (13/14), with a single CE sample. Cosine learning-rate decay combined with insufficient early bootstrap appears to prevent the model from escaping the low-PR regime.

This result mirrors two other non-convergence cases: (1) the seed-dependent 4B non-convergence (s45, s52; Appendix D.5), where identical 4B configurations fail to bootstrap, and (2) the LLaMA-3.2-3B cold-start failure (Appendix D.4), where the base model lacks FOL generation capability entirely. Together, these cases establish a convergence threshold for GRPO-driven degeneration: the structural collapse documented in §4.2 requires successful RL optimization as a prerequisite, and at 8B scale the rollout diversity bottleneck prevents this optimization from occurring under our configuration.

E Use of AI Assistance

We used Claude for coding, code drafting, debugging, and language polishing.

1587