IF YOU CAN'T BEAT IT, COPY IT: HINT RELIANCE AND LEARNING SIGNALS IN RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) checks only the final answer. When a prompt contains a candidate solution, a policy can earn the same reward by solving the problem or by copying the candidate. We study which route RLVR strengthens, using Countdown problems with hints of controlled reliability. Checkpoints trained with always-correct hints reach near-perfect correct-hint accuracy yet range from 0.00 to 0.60 in wrong-hint accuracy. In Countdown with Qwen2.5-7B and Llama-3.1-8B-Instruct and in an arithmetic task with Qwen2.5-Math-7B, copying grows when the hint is more reliable than the accuracy the model reaches without hints at the same budget. A model that cannot beat the hint learns to copy it, and the updates spent copying are lost to learning. Copying persists because a saturated copier gives every response to a prompt the same reward, which leaves group-relative RL no contrast within a prompt. Adding no-hint questions restores this contrast. The first response token can choose between the routes a model has. On held-out questions, replacing one hidden state lowers wrong-hint copying from 0.97 to 0.35 and from 0.80 to 0.28, and forcing only the first token lowers it to 0.32 and 0.31. The same control applied during training creates reward contrast only while it is on. RLVR learns from the alternatives that the policy itself samples.

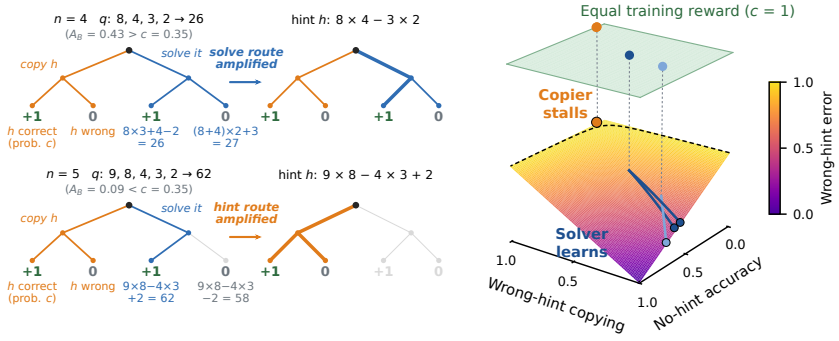

Figure 1: **Same reward, different routes. Left:** Countdown examples at $c = 0.35$. Training amplifies the solving route when $A_B > c$ (four numbers) and the hint route when $A_B < c$ (five numbers). **Right:** checkpoints trained at $c = 1$ earn the same reward (upper sheet) but differ in measured copying and no-hint accuracy. Three base runs (orange) and two solver starts (dark and light blue) are shown. Continued at $c \approx 0.21$, the solvers move into the low-error basin, while a saturated copier formed at $c \approx 0.81$ (large orange marker) stays on the plateau.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) trains language models on answers that a program can check (Shao et al., 2024; Guo et al., 2025a). The check looks only at the final answer. When the prompt also contains a candidate solution, two policies can earn the same reward. One solves the problem and the other copies the candidate. A correct hint cannot tell them apart, but a

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

wrong hint can. We ask which route RLVR strengthens, why copying sometimes survives a change in the training data, and whether the route can be moved directly.

We study these questions in Countdown, where the verifier is exact and we set the hint reliability c, the fraction of training hints that are correct. We compare each hinted run with the budget-matched no-hint reference A_B, the accuracy that the same model reaches when it trains on the same questions without hints (Equation 1), and evaluate every checkpoint on correct-hint, wrong-hint and no-hint versions of held-out questions. We test the same rule on a second, arithmetic task in which the prompt recommends one of two printed answers.

Across two tasks and three models, copying grows when the hint is more reliable than A_B and does not grow when it is less reliable (Figure 2). The same hint with $c = 0.35$ leaves excess copying almost unchanged on four-number problems, where $A_B = 0.43$, and raises it by more than 0.80 on five-number problems, where $A_B = 0.09$. The outcome reward pays for a correct answer however it was produced, so copying maximizes reward whenever the hint beats the model's own route. This is the rule in our title. A model that cannot beat the hint learns to copy it.

We then examine what copying does to training and show that (1) equal reward hides the learning that copying costs, (2) copying becomes persistent once it saturates, because uniform rewards leave group-relative RL no contrast within a prompt, (3) the first response token can choose between the routes a model has, and (4) RLVR learns only from alternatives that the policy samples itself, so control from outside creates contrast only while it is applied.

Our findings highlight a gap between what RLVR rewards and what it trains. A reward curve cannot tell a solver from a copier, but a wrong-hint and a no-hint evaluation can, and the fraction of mixed-reward groups in the training logs shows when training has stopped comparing alternatives. Fixes work when they change what the policy itself samples. No-hint mixing does so, and a product reward over hint pairs does not.

2 Related work

What RL selects. GRPO with rule-based rewards is the standard recipe for reasoning training (Shao et al., 2024; Guo et al., 2025a). Whether it adds reasoning ability or selects behavior the base model already has is debated (Yue et al., 2025; Wen et al., 2026; Liu et al., 2025a), and several analyses place most of the change on a few critical tokens as entropy collapses (Chen et al., 2026; Meng et al., 2026; Cui et al., 2025). In deep RL, early experience shapes later learning (Nikishin et al., 2022; Lyle et al., 2023). We study selection between two routes that earn the same reward. The first response token is such a critical token, and a saturated copier is such a case of path dependence.

Hints as cues and as training aids. Networks adopt the simplest rule their training signal allows (Geirhos et al., 2020), and language models follow hints, user opinions and conflicting context, often without saying so (Turpin et al., 2023; Lanham et al., 2023; Chen et al., 2025b; Sharma et al., 2023; Xie et al., 2023; Zhou et al., 2023). RLVR pipelines add hints, expert traces or partial solutions on purpose to guide exploration (Yan et al., 2025; Zhang et al., 2026a; 2025; Agashe et al., 2026; Li et al., 2025b; Xia et al., 2026; Deng et al., 2026; Cao et al., 2026; Li & Gao, 2025; Xie et al., 2026; Zhang et al., 2026b), restore within-prompt contrast by filtering or conditioning rollouts (Yu et al., 2025; Zhong et al., 2026; He et al., 2026), or reward the process instead of the answer (Lightman et al., 2024; Guo et al., 2025b; Wang et al., 2025; Yuan et al., 2026; Pronesti et al., 2026; Ye et al., 2025; Liu et al., 2025b). These works use hints and contrast as tools. We ask when an outcome reward turns a hint into a shortcut and show that the answer depends on the reachable alternative A_B.

Auditing rewards and representations. An outcome reward does not certify the intended mechanism. Agents game literal objectives, RL reasoning training increases such gaming, and learned reward seeking can generalize (Krakovna et al., 2020; Pan et al., 2022; Nishimura-Gasparian et al., 2026; Helff et al., 2026; Wu & Tang, 2026; Denison et al., 2024; Taylor et al., 2025). Random or incorrect rewards can still raise accuracy by amplifying behavior already in the model (Shao et al., 2026; Yan et al., 2026), while verified-incorrect annotations harm RLVR (Zhu & Kang, 2026). Causal readouts, monitorability analyses, activation steering and representation finetuning test and change what a model computes (Yu et al., 2026; Yang et al., 2025b; Xiong et al., 2026; Turner et al., 2023; Rimsky et al., 2024; Zou et al., 2023; Wu et al., 2024; Poupart et al., 2026; Arditi et al., 2024; Huang

et al., 2026; Li et al., 2025a; Wang et al., 2026; Wu et al., 2026). Our wrong-hint test flips only the hint, and our state replacement uses the model's own no-hint state to locate where the route is chosen. Appendix G discusses these lines in detail.

3 SETUP

Task and hints. A Countdown problem gives n numbers and a target. A valid answer uses each number once and evaluates to the target. A fixed parser and verifier assign a binary reward $R(x, y) \in \{0, 1\}$. A training prompt may include a candidate expression h. The candidate is correct on a fraction c of training questions and wrong on the rest, and every response to a prompt sees the same candidate. The no-hint version removes the candidate.

Models and training. Our main model is Qwen2.5-7B (Yang et al., 2025a). We repeat the reliability experiments with Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and run them on an arithmetic task with Qwen2.5-Math-7B (Appendix B). Countdown training uses a group-normalized outcome reward with eight responses per prompt and 64 prompts per update, without a KL penalty, as in DAPO (Yu et al., 2025). A stage lasts 24 updates. Continuation stages start from the previous weights with a fresh optimizer. Appendix A gives the remaining settings. For a starting policy π_0, training procedure $\mathcal{A}$ and budget B, we define the no-hint reference

$$A_B(\pi_0, \mathcal{A}) = \mathrm{Acc}^{\varnothing}(\mathcal{A}(\pi_0, \mathcal{D}_{\varnothing}, B)), \tag{1}$$

the no-hint accuracy after training on the same questions without hints. A_B is a property of a model, task and budget, measured by an actual training run.

Measurements. We evaluate each checkpoint on correct-hint, wrong-hint and no-hint versions of the same 256 held-out questions, with eight samples per version. The three accuracies are Acc^{+}, Acc^{-} and $\mathrm{Acc}^{\varnothing}$. Let $C(y, h)$ indicate that the final expression in response y matches the printed candidate h after a fixed normalization. Wrong-hint copying at checkpoint k is

$$\mathrm{Copy}_k^{-} = \mathbb{E}_x \mathbb{E}_{y \sim \pi_k(\cdot|x,h^-)} C(y, h^-). \tag{2}$$

Copying follows the printed string. When a different wrong expression is printed for the same question, the copying rate barely changes and the expression no longer printed never appears. Counting algebraically equivalent rewrites as copies changes Copy^{-} by at most 0.07 in any run (Appendix A). A model can also produce the wrong candidate without seeing it. We subtract that rate to get excess copying:

$$G_{10}(k) = \mathrm{Copy}_k^{-} - \mathbb{E}_x \mathbb{E}_{y \sim \pi_k(\cdot|x,\varnothing)} C(y, h^-), \qquad \Delta G_{10} = G_{10}(24) - G_{10}(0). \tag{3}$$

Copy^{-} describes a checkpoint and ΔG_{10} describes what training did to it. Intervals are 95% question-level paired bootstrap intervals with 10,000 resamples. On four-number problems, the settings labeled 0.20 and 0.80 have realized reliabilities of about 0.21 and 0.81, which we use elsewhere. For the inference-time interventions in Section 6, we choose all settings on 128 fitting and 64 calibration questions and then evaluate on 256 separate held-out questions without further tuning.

4 IF THE MODEL CAN'T BEAT THE HINT, IT COPIES

Table 1 follows Qwen2.5-7B on Countdown through training histories, and Figure 2 tests the rule in our title on two tasks and three models. A solver was trained for 24 updates without hints, a selective start copies few wrong hints ($\mathrm{Copy}^{-} \leq 0.19$), a blind copier copies most of them ($\mathrm{Copy}^{-} \approx 0.93$), and a saturated copier copies every one ($\mathrm{Copy}^{-} \geq 0.998$). Each start is the endpoint of a 24-update run from the base model, with or without hints.

4.1 EQUAL SCORES HIDE DIFFERENT BEHAVIOR

Correct hints make the seven Qwen checkpoints trained at $c = 1$ look the same. Their correct-hint accuracy lies between 0.996 and 1.000, and all of them end training with reward near 1. Wrong hints separate them (Table 1). The three runs started from the base model copy the wrong candidate

Table 1: **Main results: training histories** of Qwen2.5-7B on four-number Countdown. Each row trains the start checkpoint for 24 updates at hint reliability c with a fraction q of questions shown without hints. †48 further updates after a first stage at the same c from the base model. ‡48 updates. Mean$_{\pm\text{std}}$ over runs, single runs without std. Background color scales with Copy$^-$.

	Training		Held-out evaluation		
Start	c	q	Copy$^-$ ↓	Acc$^-$ ↑	Acc$^\varnothing$ ↑
Base	1.00	0.00	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.08_{\pm 0.01}$
Selective	1.00	0.00	$0.11_{\pm 0.10}$	$0.51_{\pm 0.14}$	$0.34_{\pm 0.05}$
Solver	1.00	0.00	$0.35_{\pm 0.20}$	$0.34_{\pm 0.10}$	$0.45_{\pm 0.04}$
Saturated copier	≈0.21	0.00	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.09_{\pm 0.00}$
Blind copier	≈0.21	0.00	$0.01_{\pm 0.00}$	$0.76_{\pm 0.00}$	$0.46_{\pm 0.00}$
Solver after $c = 1$	≈0.21	0.00	$0.01_{\pm 0.01}$	$0.81_{\pm 0.04}$	$0.63_{\pm 0.08}$
Blind copier†	≈0.81	0.00	$0.02_{\pm 0.01}$	$0.78_{\pm 0.05}$	$0.43_{\pm 0.01}$
Selective†	0.65	0.00	0.00	0.85	0.50
Saturated copier†	0.65	0.00	1.00	0.00	0.09
Saturated copier	≈0.21	0.25	$0.18_{\pm 0.06}$	$0.43_{\pm 0.05}$	$0.42_{\pm 0.07}$
Saturated copier	≈0.21	0.50	$0.24_{\pm 0.09}$	$0.39_{\pm 0.11}$	$0.46_{\pm 0.03}$
Saturated copier	≈0.21	0.75	$0.51_{\pm 0.21}$	$0.24_{\pm 0.08}$	$0.53_{\pm 0.02}$
Saturated copier	n/a	1.00	$0.88_{\pm 0.12}$	$0.07_{\pm 0.07}$	$0.53_{\pm 0.01}$
Blind copier	n/a	1.00	$0.38_{\pm 0.04}$	$0.38_{\pm 0.00}$	$0.64_{\pm 0.01}$
Base‡	n/a	1.00	$0.03_{\pm 0.01}$	$0.72_{\pm 0.05}$	$0.78_{\pm 0.04}$

almost every time (Copy$^- \geq 0.998$, Equation 2) and score 0. The two selective starts and the two solver starts keep wrong-hint accuracy between 0.27 and 0.60. The training reward cannot see this difference because training at $c = 1$ never shows a wrong hint. How far a solver moves toward copying depends on the dose. Starting from copying of 0.13 and 0.17, two solvers end 24 updates at 0.02 and 0.04 when $c \approx 0.81$, at 0.13 and 0.16 when $c = 0.95$, and at 0.21 and 0.49 when $c = 1$.

4.2 Reliability counts relative to the reachable alternative

A one-line calculation shows when copying pays. Suppose a policy copies the hint with probability x and otherwise solves with success rate y. Its expected reward at reliability c is

$$\mathbb{E}[R] = cx + (1 - x)y, \qquad \frac{\partial\, \mathbb{E}[R]}{\partial x} = c - y. \tag{4}$$

More copying earns more reward exactly when the hint is more reliable than the model's own route. Since y grows during training, we compare c with the accuracy the policy can reach, A_B.

The Countdown runs follow Equation 4 (Figure 2). At $c = 0.35$, excess copying (Equation 3) barely moves on four-number problems and rises by 0.82–0.93 on five-number problems, whose no-hint references 0.43 and 0.09 lie on opposite sides of 0.35. Across the full Qwen sweep, **in every setting outside the shaded band** $|c - A_B| \leq 0.05$**, copying grew when** $c > A_B$ **and stayed within** ± 0.05 **when** $c < A_B$ (Appendix Table 3). Above the boundary, runs can still diverge, and at $c = 0.65$ on four numbers one run became a saturated copier while the other two kept copying at 0.19 and 0.35. Llama-3.1-8B-Instruct moves in the same direction. Copying grows in all seven Llama runs with $c > A_B$, by 0.66 at $c = 1$ and by 0.025 to 0.105 at $c = 0.95$ and at $c \approx 0.81$ on five numbers, and a $c \approx 0.81$ hint that sits at its four-number reference barely changes copying. Qwen2.5-14B ($A_B = 0.64$) copies more at $c = 1$ ($\Delta G_{10} = 0.83$) and less at $c \approx 0.21$ (-0.16; Appendix Table 6).

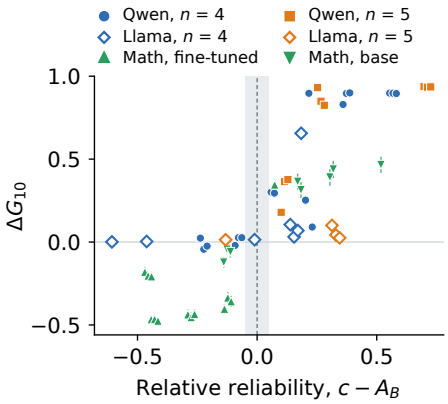

Figure 2: **Copying follows** $c - A_B$**.** One run per marker, 24 updates, 95% intervals. Math: Qwen2.5-Math-7B.

The rule holds on a second task. In the arithmetic task, each problem prints two candidate answers, and an answer key recommends one of them that is correct with probability c. We train Qwen2.5-Math-7B from three starts. A checkpoint fine-tuned on easier problems, with half of its demonstrations copying the recommendation, reaches $A_B = 0.93$ without a recommendation, and the base model reaches 0.48. The same rule separates the runs (triangles in Figure 2). From this start, recommendations of reliability 0.50 to 0.80 all fall below A_B, and copying of a wrong recommendation drops by 0.34 to 0.48 although this start was taught to copy. Only a perfect recommendation is copied more. From the base model, every recommendation above 0.65 is copied more and a 0.36 recommendation is not, so a recommendation near 0.80 is abandoned in one setting and adopted in the other. A third start, fine-tuned only on derivations, solves 0.40 of the questions before training and reaches $A_B = 0.95$. At $c = 0.50$, above where it starts but below A_B, copying drops in all three runs. Of the 21 hinted runs, 19 move beyond ± 0.05 in the direction of the rule and none moves the other way (Appendix Table 7).

4.3 Copying costs learning

Copying is also cheaper per response. At the end of training, responses that copy a wrong hint have a median length of 39 tokens, against 495 for correct responses that do not copy. The reward does not charge a copier for the skills it fails to build, and the cost appears once the hint is removed (Table 1, rows with $q = 1$). After 48 updates, no-hint accuracy is 0.78 for Qwen trained without hints throughout, 0.64 when the first 24 updates at $c \approx 0.81$ made it a blind copier and 0.53 when the same hints made it a saturated copier. Over the same updates, the two copier lineages generate only 0.42 to 0.58 as many response tokens as the lineage trained without hints. Copiers still learn once the hint is removed, but they start from behind. An 8,192-token limit changes these accuracies by at most 0.02 (Appendix A). The gap also appears within a single stage (Table 2). After 24 updates at $c = 1$, no-hint accuracy is 0.08 for Qwen and 0.32 for Llama, against 0.43 and 0.82 after no-hint training. Hinted training lowers no-hint accuracy even where copying stays flat, and most where copying grows.

Table 2: **No-hint accuracy** after 24 hinted updates, $n = 4$. Shading marks the drop below A_B.

c	Qwen	Llama
A_B	*0.43*	*0.82*
≈0.21	$0.32_{\pm 0.05}$	0.78
0.35	$0.29_{\pm 0.04}$	0.78
≈0.81	$0.10_{\pm 0.02}$	0.55
1.00	$0.08_{\pm 0.01}$	0.32

5 Copying persists when groups lose contrast

5.1 Only saturated copiers stay trapped

We continue several checkpoints at $c \approx 0.21$, where the hint is wrong on 79% of training questions (Table 1). Three runs from two solver checkpoints trained at $c = 1$ drop wrong-hint copying to 0.013, 0.002 and 0.002 within 24 updates, and their wrong-hint accuracy rises to 0.78–0.85. Blind copiers that still produce some non-copying answers, with copying at 0.93, fall to 0.01. **Two runs from a saturated copier do not move.** Their copying stays at 1.00 and their wrong-hint accuracy at 0 (Figure 3a). Longer training does not change this. Over 72 updates at $c = 0.65$, a selective line ends at 0.003 copying while a saturated line stays at 1.000, and blind copiers trained at $c \approx 0.81$ end at 0.008 and 0.023. A solver lineage trained at $c = 1$ shows a third outcome. Its copying rises from 0.17 to about 0.5 and then stays there through 72 updates (Figure 3b).

5.2 Uniform groups carry no comparison

GRPO normalizes each reward against the other responses to the same prompt (Shao et al., 2024). If all G responses receive the same reward, the group gives no comparison between outcomes. This happens both when every response fails and when every response succeeds. Mean reward does not show how often it happens. For a question with success probability p_x, a group of G independent samples contains both outcomes with probability

$$F_G(p_x) = 1 - p_x^G - (1 - p_x)^G. \tag{5}$$

At a fixed mean reward $\mu = \mathbb{E}_x p_x$,

$$0 \le \mathbb{E}_x F_G(p_x) \le F_G(\mu), \tag{6}$$

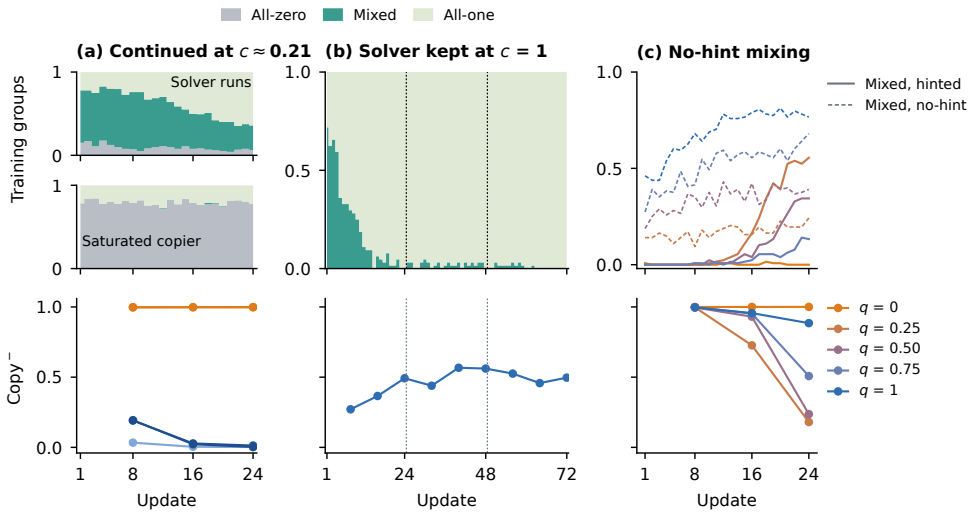

Figure 3: **Group composition and copying during training.** Top: composition of hinted training groups (a, b) and mixed groups as fractions of all groups (c, mean of two runs). Bottom: wrong-hint copying. (a) Continuation at $c \approx 0.21$. (b) A solver kept at $c = 1$ for three stages, with dotted lines at stage boundaries. (c) Saturated copiers trained with a fraction q of no-hint questions.

with both bounds attained (Appendix C). At $\mu = 0.5$ and $G = 8$, the expected fraction of mixed groups can be 0, if half the questions always succeed and half always fail, or $127/128$, if every question succeeds half the time. Figure 1 (right) applies this to the policy of Equation 4. Its lower surface is the wrong-hint error $1 - (1 - x)y$, and beyond the dashed line the expected fraction of mixed wrong-hint groups, $F_8\big((1 - x)y\big)$ from Equation 5, is below 0.05.

The training logs reach both ends. Copying removes its own training signal as it forms. When the base model trains at $c = 1$, 0.69 of hinted groups are mixed at the first update and almost none by the last, while copying rises from 0.10 to 1.00. With a KL penalty toward the base model, copying still reaches 1.00. Saturated copiers continued at $c \approx 0.21$ or $c = 0.65$ keep only 0.001–0.005 of hinted groups mixed on average, and in 18 to 22 of their 24 updates every token receives zero advantage (Appendix C). Wrong-hint groups are all-zero and correct-hint groups are all-one, so the reward separates prompts but never separates responses to the same prompt. Neither filtering uniform groups, as DAPO does (Yu et al., 2025), nor larger groups supplies the missing comparisons. With DAPO's dynamic sampling, a saturated copier finds 7 mixed groups among 4,608 sampled prompts and skips 18 of 24 updates, and with 16 samples per prompt or a sampling temperature of 1.2 its wrong-hint copying stays above 0.99 (Appendix C). What it lacks is any gradient. A KL penalty toward the copier itself, in loss form with the k1 estimator, rewards nothing and its gradient has zero expectation, but the gradient is nonzero in every update, and with it copying falls to 0.006 and 0.16 in two runs. The $c = 1$ solver lineage reaches the same state through success. Its mixed fraction is 0.72 at the first update and averages 0.020 and 0.008 in the second and third stages, as almost every group becomes all-one. Its copying stops changing at the same point.

5.3 No-hint questions restore contrast

Mixing in no-hint questions changes this. Starting from a saturated copier, we remove the hint from a fraction q of training questions. In one run at $q = 0.25$, the fraction of mixed hinted groups is 0 at updates 1 and 8, 0.216 at update 16 and 0.756 at update 24, and wrong-hint copying falls from 0.999 to 0.223 over the same period. The contrast appears first on no-hint questions, where the policy learns to solve (Figure 3c). Its solutions then start to appear on hinted prompts, and those groups become mixed. All six runs with $q \in \{0.25, 0.50, 0.75\}$ end with 0.69–0.82 of hinted groups mixed and copying between 0.14 and 0.65, against 1.00 at $q = 0$. The effect is not monotone in q. The smallest mixture gave the lowest copying in both replicates, and $q = 1$, which removes hints entirely, leaves copying at 0.80 and 0.97 when a wrong hint appears at test time.

5.4 A STRICTER REWARD NEEDS SUCCESSES TO ACT ON

A natural alternative is to reward consistency directly. We present each training question twice, once with a correct and once with a wrong hint, and give the i-th responses to the two versions the product reward $r_i^{\text{pair}} = r_i^+ r_i^-$. A pair earns credit only if both responses succeed. The control uses the same paired data but scores the two versions independently. The product reward helps copiers that still produce some correct answers. From a start with copying 0.93, both runs end at 0.05–0.06 copying and 0.62 wrong-hint accuracy. It does nothing for the saturated copier (Appendix Table 22). Every product reward is zero for all 24 updates, and the checkpoint does not change. On solver starts it is slightly worse than independent scoring of the same data. The paired data, which puts wrong hints into training, does the work, and multiplying the rewards adds nothing.

6 THE FIRST TOKEN CHOOSES BETWEEN EXISTING ROUTES

Sections 4 and 5 change the training data and read out the result. We now freeze the weights and act on the forward pass. We use three checkpoints. S1 and S2 are saturated copiers that were then trained for 24 updates without hints. They solve about half of no-hint questions but still copy most wrong hints. SAT0 is a saturated copier with no-hint accuracy 0.09.

6.1 ONE HIDDEN STATE REDIRECTS THE RESPONSE

At the last prompt position, we replace the output of decoder block 19 with the state that the same model computes for the same question without the hint. We chose the block and position on fitting and calibration questions and froze them before the held-out evaluation (Figure 4a; other blocks and positions in Appendix D). S1 copying falls from 0.970 to 0.348 and its accuracy rises from 0.017 to 0.364. S2 moves from 0.799 to 0.276 copying and from 0.117 to 0.413 accuracy. A random donor state of the same norm changes copying by at most 0.015. The replaced state carries a way of responding, and two observations show that it does not carry a solution. A donor state taken from another checkpoint works about equally well whether it is computed on the same question or on a different one (S1 copying 0.337 and 0.359 on calibration questions), and with a correct hint the model still adopts the printed candidate, now after writing a short derivation. On calibration questions, wrong-hint responses grow from about 50 tokens to about 700 as the model works the problem out. The replacement can only point the model toward a route it already has. SAT0, which rarely solves without a hint, stays at 0.993 copying.

6.2 THE FIRST TOKEN CARRIES MOST OF THE EFFECT

The replacement acts at one position, so its most direct output is the distribution of the first response token. On held-out questions, S1 starts 96% of wrong-hint responses with *The*, as in "The answer is…", and nearly all of these responses copy. Without a hint it starts with *To*, as in "To solve this…". Forcing S1's own no-hint first token, with no change to any hidden state, cuts copying from 0.970 to 0.321 and raises accuracy from 0.017 to 0.385. Forcing the single token *To* on every question does the same (0.320 and 0.386), and forcing *The* pushes copying back to 0.996. S2 falls from 0.799 to 0.307 with *To* and rises to 0.975 with *The*. The effect runs in both directions. C1, a former copier that no-hint mixing moved to solving, starts with *To*, and forcing *The* raises its copying from 0.29 to 0.68 (Appendix D).

A 2×2 test separates the first token from the rest of the replaced forward pass (Figure 4b, procedure in Appendix Figure 7). We sample the first token from the native distribution p_0 or the replaced distribution p_1, and continue decoding with the native prompt cache C_0 or the replaced cache C_1. Changing only the first-token distribution brings copying from 0.969 to 0.395 in S1 and from 0.801 to 0.254 in S2. Changing only the cache leaves S1 at 0.965 and moves S2 to 0.676. With both changes, copying is 0.297 in S1 and 0.254 in S2. **The first token accounts for most of the effect.** The 256 held-out questions confirm that the first token accounts for most of the effect (Appendix Table 16). Like the replacement, a forced token selects a route but cannot create one. In copiers that rarely solve, forcing *To* lowers copying by only 0.09–0.19 (Appendix Table 17).

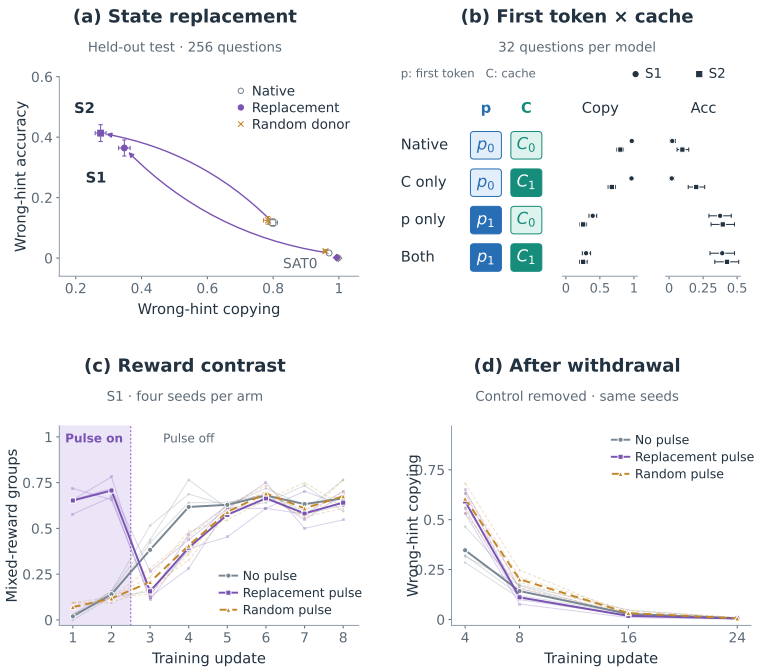

Figure 4: **The first token can set the route, and training follows the policy's own samples.** (a) Block-19 state replacement on 256 held-out questions. (b) First-token × cache test on 32 questions per model (256 held-out questions in Appendix Table 16). (c) Mixed hinted groups when the replacement is applied during updates 1–2. (d) Wrong-hint copying after the control is removed. Thin lines are individual runs and thick lines are means over four runs.

6.3 Learning follows the policy's own samples

If copying persists because the policy's own groups lack contrast, a control that supplies solving responses should help only while it is applied. We test this on S1 at $c \approx 0.21$.

First, we apply the state replacement to every rollout during the first two updates of a 24-update continuation and then remove it. We compare against training without control and against a random pulse of the same norm, with four runs per arm. While the control is on, 0.65 and 0.71 of hinted groups are mixed, against 0.02 and 0.14 without control and 0.07 and 0.12 with the random pulse (Figure 4c). One update after removal, the fraction falls to 0.16, below the 0.38 of ordinary training. At update 4, copying is higher in the pulsed runs than in ordinary training in all four runs, and the random pulse causes a similar delay (Figure 4d). By update 8 every run has copying below 0.25, and at update 24 all three arms are near 0.005. The contrast existed only in responses that the control produced, and it left with the control.

Second, we leave the trained policy uncontrolled and use the control only to generate data. Each group of eight has four native responses and four controlled ones, and the uncontrolled policy is updated on all eight with truncated importance weights (Appendix E). On wrong-hint prompts the success rate of the controlled half exceeds that of the native half by 0.29–0.37, while a temperature control matched in first-token KL gains only 0.04–0.07. Little of this credit reaches the policy. Under the uncontrolled policy, the first token of a controlled response has a median probability ratio of 0.04–0.09, so the update gives it little weight. After two updates, the uncontrolled policy's probability of opening a wrong-hint calibration response with a solving token rose less than in native-only training with the same trainer, by 0.19 and 0.09 in two runs.

7 DISCUSSION

A model that cannot beat a reliable hint is paid to copy it, and the results above follow from one constraint on what RLVR can reinforce, which is only what the policy samples. A_B measures how reachable the solving route is, uniform groups mark the point where the policy stops sampling alternatives, and an alternative supplied from outside does not make the uncontrolled policy sample it more than ordinary training does.

Mathematical reasoning. The same picture appears on MATH. On 252 MATH questions, the Qwen2.5-Math-7B checkpoints that Shao et al. (2026) trained for 150 steps gain 29 points with incorrect rewards and 31 with ground-truth rewards. Without any training, a step-by-step prefix recovers 58% and 55% of these gains in the base model and an oracle state direction 69% and 66%, while a direction built from the ground-truth reward gradient recovers none (Appendix Table 9). For both checkpoints, most of the gain is behavior that the base model can already produce.

Scope. Our hints print candidate answers, and our models have 7–14B parameters. Partial hints are also reused. Shown only the first operation of a wrong candidate, trained checkpoints build 0.43–0.88 of their answers on it, against 0.05–0.07 for a neutral line (Appendix B.3). The mechanism needs only two routes with equal reward and a policy that samples one far more often, so implicit cues in real training data are the natural next test.

8 CONCLUSION

A reward curve does not show how a policy earns its reward. Across two tasks and three models, a model that cannot beat the hint learns to copy it, and in Countdown the updates it spends copying are lost to learning. Copying persists once training groups stop containing both outcomes. The first response token can choose between the routes a model has, and lasting change comes from changing what the policy samples on its own.

REPRODUCIBILITY STATEMENT

Appendix A gives the training and evaluation settings, data splits and truncation handling. Appendix B gives the Llama results and the other tasks. Appendix C proves Equation 6 and describes the group-reward audit. Appendices D and E give the intervention and training-control details, and Appendix F gives the reward-pairing results.

AI USE STATEMENT

In this work, we used generative AI tools only for tasks for which the ICLR 2027 AI Policy recommends disclosure. We used them to organize our ideas, to search for and identify relevant literature, and to edit the language of the paper for readability. We did not use generative AI tools for any task that requires disclosure, including developing the conceptual framework, formulating or proving mathematical claims, proposing hypotheses, designing experiments, implementing methods, generating or processing data, analyzing data and interpreting results. The authors read every suggested reference in its original source and checked every edited passage against the intended meaning. We take responsibility for the final content of this work, including all text, claims and artifacts.

REFERENCES

Saaket Agashe, Jayanth Srinivasa, Gaowen Liu, Ramana Kompella, and Xin Eric Wang. Context bootstrapped reinforcement learning. *arXiv preprint arXiv:2603.18953*, 2026.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.11717.

Zhiyu Cao, Kaixin Wu, Mingjie Zhong, Peifeng Li, Can Ye, and Qiaoming Zhu. Hint-guided diversified policy optimization for LLM reasoning. *arXiv preprint arXiv:2606.03021*, 2026.

Xingwu Chen, Tianle Li, and Difan Zou. Reshaping reasoning in LLMs: A theoretical analysis of RL training dynamics through pattern selection. In *International Conference on Learning Representations*, 2026.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9487–9499, 2025a.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025b.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Qiyuan Deng, Kehai Chen, Min Zhang, and Zhongwen Xu. HiPO: Self-hint policy optimization for RLVR. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rcb20pHmT1.

Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, Ethan Perez, and Evan Hubinger. Sycophancy to subterfuge: Investigating reward-tampering in large language models. *arXiv preprint arXiv:2406.10162*, 2024.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025a. doi: 10.1038/s41586-025-09422-z.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in RL for large language models. *arXiv preprint arXiv:2505.23564*, 2025b.

Xixiang He, Qiyao Sun, Ao Cheng, Xingming Li, Xuanyu Ji, Hailun Lu, Runke Huang, and Qingyong Hu. Advantage collapse in group relative policy optimization: Diagnosis and mitigation. *arXiv preprint arXiv:2605.21125*, 2026.

Lukas Helff, Quentin Delfosse, David Steinmann, Ruben Härle, Hikaru Shindo, Patrick Schramowski, Wolfgang Stammer, Kristian Kersting, and Felix Friedrich. LLMs gaming verifiers: RLVR can lead to reward hacking. *arXiv preprint arXiv:2604.15149*, 2026.

Fanding Huang, Guanbo Huang, Xiao Fan, Yi He, Xiao Liang, Xiao Chen, Qinting Jiang, Faisal Nadeem Khan, Jingyan Jiang, and Zhi Wang. Semantic-space exploration and exploitation in RLVR for LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38402–38449, 2026. doi: 10.18653/v1/2026.findings-acl.1915. URL https://aclanthology.org/2026.findings-acl.1915/.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024.

Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. Specification gaming: The flip side of AI ingenuity. Google DeepMind Blog, April 2020. URL https://deepmind.google/blog/specification-gaming-the-flip-side-of-ai-ingenuity/.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Melody Zixuan Li, Kumar Krishna Agrawal, Arna Ghosh, Komal Kumar Teru, Adam Santoro, Guillaume Lajoie, and Blake Aaron Richards. Tracing the representation geometry of language models from pretraining to post-training. In *Advances in Neural Information Processing Systems*, volume 38, pp. 54691–54724, 2025a.

Qizhen Li and Hui Gao. Leveraging process hints to expand capability boundaries in LLM reasoning. In *2025 22nd International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 1–4. IEEE, 2025. doi: 10.1109/ICCWAMTIP68645.2025.11352663.

Ziheng Li, Zexu Sun, Jinman Zhao, Erxue Min, Yongcheng Zeng, Hui Wu, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Xu Chen, and Zhi-Hong Deng. Staying in the sweet spot: Responsive reasoning evolution via capability-adaptive hint scaffolding. *arXiv preprint arXiv:2509.06923*, 2025b.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *Advances in Neural Information Processing Systems*, 2025a.

Xiaoyuan Liu, Tian Liang, Zhiwei He, Jiahao Xu, Wenxuan Wang, Pinjia He, Zhaopeng Tu, Haitao Mi, and Dong Yu. Trust, but verify: A self-verification approach to reinforcement learning with verifiable rewards. In *Advances in Neural Information Processing Systems*, 2025b. arXiv:2505.13445.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, 2023.

Haoming Meng, Kexin Huang, Shaohang Wei, Chiyu Ma, Shuo Yang, Xue Wang, Guoyin Wang, Bolin Ding, and Jingren Zhou. Sparse but critical: A token-level analysis of distributional shifts in RLVR fine-tuning of LLMs. In *International Conference on Learning Representations*, 2026.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, 2022.

Kei Nishimura-Gasparian, Robert McCarthy, and David Lindner. Towards understanding specification gaming in reasoning models. *arXiv preprint arXiv:2605.02269*, 2026.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022.

Yoann Poupart, Aurélie Beynier, and Nicolas Maudet. Policy gradient steering: Interventions from behavioral objectives. *arXiv preprint arXiv:2607.27574*, 2026.

Massimiliano Pronesti, Anya Belz, and Yufang Hou. Beyond outcome verification: Verifiable process reward models for structured reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32187–32202, 2026. doi: 10.18653/v1/2026.findings-acl.1611. URL https://aclanthology.org/2026.findings-acl.1611/.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. URL https://aclanthology.org/2024.acl-long.828/.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in RLVR. *arXiv preprint arXiv:2506.10947v2*, 2026.

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

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in RLHF. In *Conference on Language Modeling*, 2024.

Mia Taylor, James Chua, Jan Betley, Johannes Treutlein, and Owain Evans. School of reward hacks: Hacking harmless tasks generalizes to misaligned behavior in LLMs. *arXiv preprint arXiv:2508.17511*, 2025.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, 2023.

Zhen Wang, Zhifeng Gao, and Guolin Ke. Masked-and-reordered self-supervision for reinforcement learning from verifiable rewards. *arXiv preprint arXiv:2511.17473*, 2025.

Ziyue Wang, Aomufei Yuan, Yongfu Zhu, Shuai Dong, Wenpu Liu, Yiran Yao, Weichu Xie, Yuqi Xu, Caoyuan Ma, Wenqi Shao, Xiaoying Zhang, Nan Duan, and Jiaqi Wang. Right makes might: Aligning verified hidden states empowers RL reasoning. *arXiv preprint arXiv:2606.03234*, 2026.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *International Conference on Learning Representations*, 2026.

Jianghao Wu, Jianfei Cai, Weiqiang Wang, Jin Ye, Daniel F. Schmidt, and Yasmeen George. Single-rollout hidden-state dynamics for training-free RLVR data selection. In *International Conference on Machine Learning*, 2026.

Rui Wu and Ruixiang Tang. From rebound to remedy: Understanding and mitigating reward hacking via representation engineering. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2604.01476.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 63908–63962, 2024.

Yu Xia, Canwen Xu, Zhewei Yao, Julian McAuley, and Yuxiong He. Learning to hint for reinforcement learning. *arXiv preprint arXiv:2604.00698*, 2026.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. *arXiv preprint arXiv:2305.13300*, 2023.

Pei-Xi Xie, Che-Yu Lin, and Cheng-Lin Yang. Mitigating distribution sharpening in math RLVR via distribution-aligned hint synthesis and backward hint annealing. *arXiv preprint arXiv:2604.07747*, 2026.

Zidi Xiong, Shan Chen, and Himabindu Lakkaraju. Monitorability as a free gift: How RLVR spontaneously aligns reasoning. *arXiv preprint arXiv:2602.03978*, 2026.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *Advances in Neural Information Processing Systems*, 2025.

Lecheng Yan, Ruizhe Li, Guanhua Chen, Qing Li, Jiahui Geng, Wenxi Li, Longyue Wang, and Chenyang Lyu. Spurious rewards paradox: Mechanistically understanding how RLVR activates memorization shortcuts in LLMs. *arXiv preprint arXiv:2601.11061*, 2026.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025a. URL https://arxiv.org/abs/2412.15115v2.

Yiran Yang, Haifeng Sun, Jingyu Wang, Qi Qi, Zirui Zhuang, Huazheng Wang, Pengfei Ren, Jing Wang, and Jianxin Liao. Unveiling internal reasoning modes in LLMs: A deep dive into latent reasoning vs. factual shortcuts with attribute rate ratio. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2186–2206, 2025b. doi: 10.18653/v1/2025.emnlp-main.111. URL https://aclanthology.org/2025.emnlp-main.111/.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through RL training. *arXiv preprint arXiv:2509.03403*, 2025.

Qinan Yu, Alexa Tartaglini, Peter Hase, Carlos Guestrin, and Christopher Potts. Outcome rewards do not guarantee verifiable or causally important reasoning. *arXiv preprint arXiv:2604.22074*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Huining Yuan, Zelai Xu, Huaijie Wang, Xiangmin Yi, Jiaxuan Gao, Xiao-Ping Zhang, Yu Wang, Chao Yu, and Yi Wu. Verifiable process rewards for agentic reasoning. *arXiv preprint arXiv:2605.10325*, 2026.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=4OsgYD7em5.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. StepHint: Multi-level stepwise hints enhance reinforcement learning to reason. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37846–37864, 2026a. doi: 10.18653/v1/2026.acl-long.1755. URL https://aclanthology.org/2026.acl-long.1755/.

Wenyuan Zhang, Shuaiyi Nie, Zhengyang Ai, Chengguang Tang, Xinghua Zhang, Yi Liu, Tingwen Liu, and Pinyan Lu. A survey on hint-based RLVR: Overcoming zero-advantage failures with external textual signals. *Preprints.org*, 2026b. doi: 10.20944/preprints202606.1050.v1. Preprint 202606.1050, not peer-reviewed.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-GRPO: Scaffolded group relative policy optimization for enhancing LLM reasoning. *arXiv preprint arXiv:2510.19807*, 2025.

Haitian Zhong, Jixiu Zhai, Lei Song, Jiang Bian, Qiang Liu, and Tieniu Tan. RC-GRPO: Reward-conditioned group relative policy optimization for multi-turn tool calling agents. *arXiv preprint arXiv:2602.03025*, 2026.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. Context-faithful prompting for large language models. *arXiv preprint arXiv:2303.11315*, 2023.

Yuxuan Zhu and Daniel Kang. Noisy data is destructive to reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2603.16140*, 2026.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A EXPERIMENTAL PROTOCOL

A.1 TRAINING AND EVALUATION

All Countdown runs use four- or five-number problems with the candidate expression, when present, added to the task prompt. One update consists of sampling 64 prompts with eight responses each, followed by four optimizer steps on the resulting batch. The learning rate is constant at 5×10^{-7} and there is no KL penalty. Continuation stages start from the previous weights with a fresh optimizer, so a 72-update lineage consists of three 24-update stages. Over the 48 updates of the $q = 1$ lineages in Table 1, training samples 18.8M and 19.9M response tokens without hints throughout, 10.6M and 11.1M after a blind-copier stage, and 8.2M and 8.4M after a saturated-copier stage. The solver-B lineage at $c = 1$ has 72 updates at $c = 1$ after 24 earlier no-hint updates. Paired-reward runs use 32 questions per update, each with a correct-hint and a wrong-hint version; the paired set has 768 questions and 1,536 prompts, and the two versions of a question stay in the same batch.

Evaluation uses 256 held-out questions, each with correct-hint, wrong-hint and no-hint versions. We draw eight responses per version at temperature 1 with a 3,072-token limit. Evaluation always uses the task verifier; matching the printed candidate earns nothing extra. Bootstrap resampling draws questions, and each resampled question keeps all its versions and responses. In the tables, gray rows are baselines, blue rows mark conditions that move the model toward solving, and orange rows mark conditions under which it copies.

A.2 COPY MATCHING AND HINT VERSIONS

A response is correct if its final expression uses each number once and reaches the target. A response copies if its final expression matches the printed candidate after normalization; matching the target value alone does not count. G_{10} compares this rate with how often the model writes the same wrong expression when no hint is shown.

The copy measure follows the printed string. Across eight checkpoints, printing a different wrong expression for the same question barely changes the copying rate, for example 0.997 against 0.994 on five-number questions at $c = 0.35$ and 0.915 against 0.924 on four-number questions at $c \approx 0.81$, and the expression that is no longer printed never appears in the output. With a correct hint, printing a second correct expression, available for 58 of the 64 calibration questions, moves the final answer to that expression in 0.998 of responses from S1 and S2 (Section 6), and the standard form of the first expression never appears. The model adopts the candidate as written.

Looser copy criteria change little. Counting a final expression as a copy when it is algebraically equivalent to the printed candidate, with the same numbers, adds 0.013–0.029 to Copy^- for the four selective and solver checkpoints trained at $c = 1$ (Table 1), less than 0.001 for the three that copy every wrong hint, and 0.005–0.044 for the Llama runs. Counting any expression with the candidate's value adds 0.014–0.031 for the same four Qwen checkpoints. Over all 94 evaluated runs, algebraic equivalence changes Copy^- by at most 0.07.

In Figure 1 (left), $8 \times 3 + 4 - 2 = 26$ is a correct four-number solution and $(8 + 4) \times 2 + 3 = 27$ is wrong. For five numbers, $9 \times 8 - 4 \times 3 + 2 = 62$ is correct and $9 \times 8 - 4 \times 3 - 2 = 58$ is wrong. Wrong candidates are checked against the verifier before use.

The intervention questions have up to six versions: no hint, two correct candidates, two wrong candidates and a more distant wrong candidate. A second correct expression exists for 58 of the 64 calibration questions. The wrong-hint results in the main text average the two wrong candidates within each question.

A.3 QUESTION SPLITS

The intervention questions form four disjoint sets: 128 fitting, 64 calibration and 256 held-out four-number questions, plus 128 five-number questions. None of their task keys appears in any training or evaluation set used elsewhere, and prompts use the same templates. These questions come from the official training pool, whereas the checkpoint evaluations use the official test pool, so intervention results are compared only with baselines evaluated on the same questions.

A.4 TRUNCATION

A response that reaches the length limit without a final answer counts as incorrect and as not copied, in the same way that it receives zero reward in training. We report truncation rates where they bear on a comparison. Copied answers are short and derivations are long. Across the step-24 evaluations of all training runs (about 92,000 wrong-hint responses), responses that copy have a median length of 39 tokens, correct non-copying responses 495 tokens, and other incorrect responses 647 tokens. Truncation therefore rises whenever a policy moves from copying to solving. Truncated responses would rarely have been correct. Re-evaluating the no-hint version with an 8,192-token limit, for the six checkpoints with $q = 1$ in Table 1, removes almost all truncation (from 0.09–0.13 to at most 0.001) and changes no-hint accuracy by -0.002 to $+0.015$, all within the 95% intervals. Responses longer than 3,072 tokens are correct in at most 0.017 of cases.

B RELIABILITY SWEEPS

All Countdown sweep runs start from the base model and train for 24 updates. Table 3 gives the Qwen2.5-7B sweep with three runs per setting, except two at $c = 0.50$.

Table 3: Qwen2.5-7B Countdown sweep after 24 updates. Mean$_{\pm\text{std}}$ over runs. Orange rows are settings in which excess copying grew.

n	c	$c - A_B$	Copy^{-}	Acc$^{\varnothing}$	ΔG_{10}
4	No hint	–	$0.147_{\pm 0.021}$	$0.432_{\pm 0.036}$	$0.048_{\pm 0.020}$
4	≈0.21	−0.22	$0.084_{\pm 0.034}$	$0.320_{\pm 0.047}$	$-0.015_{\pm 0.034}$
4	0.35	−0.08	$0.110_{\pm 0.027}$	$0.293_{\pm 0.038}$	$0.011_{\pm 0.027}$
4	0.50	0.07	$0.397_{\pm 0.006}$	$0.196_{\pm 0.023}$	$0.298_{\pm 0.004}$
4	0.65	0.22	$0.514_{\pm 0.428}$	$0.207_{\pm 0.118}$	$0.413_{\pm 0.427}$
4	≈0.81	0.37	$0.976_{\pm 0.040}$	$0.100_{\pm 0.024}$	$0.875_{\pm 0.039}$
4	1.00	0.57	$0.999_{\pm 0.001}$	$0.085_{\pm 0.007}$	$0.897_{\pm 0.001}$
5	No hint	–	$0.165_{\pm 0.016}$	$0.089_{\pm 0.038}$	$0.102_{\pm 0.016}$
5	0.20	0.11	$0.369_{\pm 0.110}$	$0.046_{\pm 0.007}$	$0.307_{\pm 0.111}$
5	0.35	0.27	$0.931_{\pm 0.056}$	$0.024_{\pm 0.001}$	$0.868_{\pm 0.056}$
5	0.80	0.71	$0.998_{\pm 0.002}$	$0.022_{\pm 0.010}$	$0.936_{\pm 0.002}$

B.1 LLAMA

Untrained Llama-3.1-8B-Instruct writes a boxed expression in 6.7% of responses and solves 1.2% of no-hint questions, about the same success rate as the Qwen base model (1.1%). Training without hints raises format validity to 0.86 within 24 updates. We train nine conditions (Table 4) and repeat the four-number $c = 0.95$ and five-number $c \approx 0.81$ settings with two further seeds (Table 5). All six runs raise excess copying, with intervals above zero. The four-number run at $c \approx 0.81$ lies within 0.01 of its reference; its copying rises to 0.195 at update 8 and falls back to 0.020 by update 24. At $c = 1$, copying rises steadily to 0.322, 0.539 and 0.664 at updates 8, 16 and 24.

Llama truncates 0.29–0.43 of responses before training and 0.00–0.33 after, with higher rates on five-number tasks. To check that truncation does not drive ΔG_{10}, we recompute it after assigning each truncated response the copy rate observed among that run's completed responses longer than 2,000 tokens (upper 95% bound). The intervals become $[0.066, 0.155]$ at four-number $c = 0.95$, $[0.614, 0.705]$ at $c = 1$, $[0.072, 0.198]$ at five-number $c \approx 0.81$ and $[-0.014, 0.038]$ at five-number $c = 0.35$, so the direction of ΔG_{10} is unchanged in these runs. The four-number $c \approx 0.81$ run, which sits at its reference, becomes $[-0.002, 0.055]$. The further seeds truncate 0.21–0.27 (four numbers) and 0.24–0.29 (five numbers) of no-hint responses.

Qwen2.5-14B uses the four-number Countdown setting and training recipe of Qwen2.5-7B (Table 6). Before training it writes a valid expression in 0.50 of no-hint responses and solves 0.036 of them. Truncation at update 24 is at most 0.10, and counting every truncated response as a copy leaves both changes on the same side of zero.

Table 4: Llama-3.1-8B-Instruct after 24 updates. The last column is the 95% paired bootstrap interval of ΔG_{10}.

n	Training	Copy^{-}	Acc^{-}	Acc$^{\varnothing}$	ΔG_{10} interval
4	No hint	0.007	0.792	0.816	$[-0.001, 0.007]$
4	$c \approx 0.21$	0.005	0.796	0.777	$[-0.003, 0.005]$
4	$c = 0.35$	0.008	0.844	0.783	$[-0.002, 0.009]$
4	$c \approx 0.81$	0.020	0.777	0.546	$[0.007, 0.022]$
4	$c = 0.95$	0.113	0.538	0.574	$[0.089, 0.121]$
4	$c = 1$	0.664	0.088	0.316	$[0.635, 0.678]$
5	No hint	0.027	0.390	0.488	$[0.017, 0.032]$
5	$c = 0.35$	0.016	0.438	0.316	$[0.007, 0.021]$
5	$c \approx 0.81$	0.103	0.337	0.326	$[0.085, 0.116]$

Table 5: Three Llama runs in two settings after 24 updates. The first run of each setting is the one in Table 4.

Setting	ΔG_{10}	95% interval	Acc$^{\varnothing}$
$n = 4, c = 0.95$	0.105	$[0.089, 0.121]$	0.574
$n = 4, c = 0.95$	0.031	$[0.021, 0.042]$	0.390
$n = 4, c = 0.95$	0.069	$[0.054, 0.084]$	0.530
$n = 5, c \approx 0.81$	0.100	$[0.085, 0.116]$	0.326
$n = 5, c \approx 0.81$	0.043	$[0.031, 0.058]$	0.125
$n = 5, c \approx 0.81$	0.025	$[0.018, 0.034]$	0.319

Table 6: Qwen2.5-14B on four-number Countdown after 24 updates.

Training	ΔG_{10}	95% interval	Acc$^{\varnothing}$
No hint	-0.110	$[-0.129, -0.091]$	0.637
$c \approx 0.21$	-0.158	$[-0.175, -0.141]$	0.501
$c = 1$	0.828	$[0.811, 0.844]$	0.186

B.2 Other tasks

In the arithmetic task, each prompt prints two candidate answers and states that an answer key recommends one of them. The recommendation is correct with probability c, and the copy measure counts final answers equal to the recommended wrong candidate. Table 7 reports all runs. The no-hint reference trains on prompts that print both candidates without a recommendation. The mixed start is Qwen2.5-Math-7B supervised for 256 steps on 8,192 problems of a lower level of the same family, all shown with a correct recommendation. Half of the demonstrations copy the recommendation and half derive the answer while ignoring it. The derivation-only start follows the same recipe with derivation demonstrations only; before RL it solves 0.40 of the questions without a recommendation. For this start, the value $c = 0.50$, the three seeds and the ± 0.05 criterion were fixed before training. Runs from both fine-tuned starts train on harder problems of the form $(((a \times b) + c) \times d + e \times f) - g$, and runs from the base model on problems of another level. All 21 hinted runs lie outside the band around A_B. For nineteen, the 95% interval of ΔG_{10} lies beyond ± 0.05 on the side the rule gives, and the other two, both at $c = 0.36$, move the same way without clearing it. From the base model, the no-hint reference of 0.48 is close to choosing one of the two candidates at random, and no-hint accuracy after training is 0.48–0.50 in all recorded runs at this setting. When the model answers with one of the two candidates, which it does in 0.92–0.93 of responses, that candidate is correct in 0.52 of cases, with 95% intervals that contain 0.50.

A rule-induction task gives the opposite case, a shortcut that pays less than solving. The base model uses an enumeration shortcut in 0.4% of responses. Training raises accuracy by 34 points (95% CI $[28.3, 40.7]$) and the shortcut does not grow. On Countdown, Llama-3.2-3B writes the required

Table 7: Arithmetic runs with Qwen2.5-Math-7B after 24 updates. $A_B = 0.93$ from the mixed start, 0.95 from the derivation-only start and 0.48 from the base model, and $c - A_B$ uses unrounded values. Mean$_{\pm\text{std}}$ over three runs, two at $c = 0.66$ and 0.79 and for the base-model and derivation-only references, single runs at $c = 1$.

Start	c	$c - A_B$	ΔG_{10}
Mixed	No recommendation	–	$-0.099_{\pm 0.026}$
Mixed	0.50	-0.43	$-0.470_{\pm 0.005}$
Mixed	0.65	-0.28	$-0.442_{\pm 0.010}$
Mixed	0.80	-0.12	$-0.366_{\pm 0.035}$
Mixed	1.00	0.07	0.345
Derivation-only	No recommendation	–	$0.021_{\pm 0.007}$
Derivation-only	0.50	-0.46	$-0.198_{\pm 0.015}$
Base	No recommendation	–	$-0.032_{\pm 0.011}$
Base	0.36	-0.13	$-0.075_{\pm 0.042}$
Base	0.66	0.18	$0.340_{\pm 0.036}$
Base	0.79	0.31	$0.415_{\pm 0.034}$
Base	1.00	0.52	0.466

answer format in only 7.2% of responses, receives almost no reward, and does not change during training.

B.3 Partial hints

We replace the full wrong candidate with its first operation ("A legacy solver's hint: start with a op b") or with a neutral line of the same length, on the 256 Countdown test questions. Reuse counts final expressions that contain the hinted operation (Table 8). The hinted operation can still be completed to a correct answer on 0.379 of questions, and reuse stays at 0.31–0.86 for the trained checkpoints on the remaining questions.

Table 8: Full, partial and neutral hints. Accuracy and reuse of the hinted operation, eight responses per question.

	Accuracy			Reuse	
Checkpoint	Full	Partial	Neutral	Partial	Neutral
Base model	0.005	0.018	0.008	0.145	0.012
SAT0	0.000	0.042	0.079	0.878	0.067
Blind copier	0.017	0.092	0.126	0.742	0.065
S1	0.012	0.246	0.519	0.583	0.054
Solver A $\to c = 1$	0.411	0.299	0.478	0.431	0.058
Blind copier, 24 more at $c \approx 0.81$	0.542	0.217	0.281	0.546	0.060

B.4 Mathematical reasoning

Table 9 compares the Qwen2.5-Math-7B base model with two checkpoints that Shao et al. (2026) released after 150 steps of RLVR, one trained with incorrect rewards and one with ground-truth rewards, on 252 MATH questions, and measures how much of each gap simple interventions on the base model recover. The oracle direction is fitted to the incorrect-reward checkpoint and transfers to the ground-truth one. The two checkpoints differ by 1.5 points (95% CI $[-0.6, 3.6]$). The 252 questions are one half of a stratified split of MATH-500, and the other 248 questions fit the directions. Each question gets eight responses at temperature 0.6, top-p 1.0 and the library default top-k of 50, with a 3,072-token limit, and a response is graded on its final answer. The prefix condition adds "Let's solve this step by step." Each direction is added at layer 22, at the last prompt position and the next 32 generated positions, with a norm of 0.2–0.4 times the hidden-state norm.

Table 9: **Accuracy available from a mathematical base model.** Qwen2.5-Math-7B on 252 MATH questions, eight responses each. Gap fraction is $(a - a_{\text{base}})/(a_{\text{RLVR}} - a_{\text{base}})$ for each RLVR checkpoint, computed from the rounded accuracies shown.

		Gap fraction (%)	
Condition	Accuracy (%)	Incorrect reward	Ground truth
Base	45.5	0.0	0.0
RLVR, incorrect reward, step 150	74.8	100.0	95.1
RLVR, ground-truth reward, step 150	76.3	105.1	100.0
Base + step-by-step prefix	62.5	58.0	55.2
Base + oracle state direction	65.7	68.9	65.6
Base + format reward-gradient direction	52.7	24.6	23.4
Base + ground-truth reward-gradient direction	44.2	−4.4	−4.2

C GROUP-REWARD ANALYSIS

C.1 PROOF OF EQUATION 6

For G independent binary rewards on one question, all samples succeed with probability p^G and all fail with probability $(1-p)^G$, so $F_G(p) = 1 - p^G - (1-p)^G$. For $G \geq 2$,

$$F''_G(p) = -G(G-1)\left[p^{G-2} + (1-p)^{G-2}\right] \leq 0. \tag{7}$$

By Jensen's inequality, $\mathbb{E}_x F_G(p_x) \leq F_G(\mathbb{E}_x p_x)$, and $F_G \geq 0$ gives the lower bound. Setting $p_x = \mu$ for every question attains the upper bound. Setting $p_x = 1$ for a fraction μ of questions and $p_x = 0$ for the rest attains the lower bound.

The two cases are ends of one family. Let a fraction ρ of questions be deterministic, with success probability 1 for a share μ of them and 0 for the rest, and let the remaining questions have success probability μ. Then

$$\mathbb{E}_x p_x = \mu, \qquad \mathbb{E}_x F_G(p_x) = (1-\rho) F_G(\mu). \tag{8}$$

At $G = 8$ and $\mu = 0.5$, the mixed fraction ranges from 0 to $127/128$ (Figure 5). At $\mu = 1$ every group is all-one. Checkpoints that all succeed under correct hints therefore have no within-prompt contrast under those hints, whatever their behavior under wrong hints.

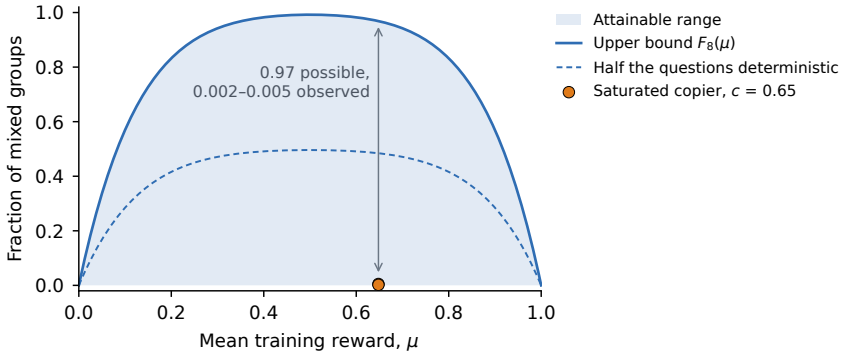

Figure 5: **Mean reward does not fix group contrast.** Expected fraction of mixed groups allowed at each mean reward by Equation 6 ($G = 8$). The dashed curve is the family above with $\rho = 0.5$; the orange marker shows the two overlapping stage means of the saturated copier kept at $c = 0.65$.

C.2 THE TRAINING-LOG AUDIT

We regroup the per-response rewards stored during training by prompt within each update; rescoring with the verifier reproduces every stored reward. We classify each group as all-zero, all-one or mixed. The trainer normalizes rewards within each group and then applies a batch-level normalization, so a uniform group still receives a constant, nonzero advantage after the second step. Such a group

shifts all its responses together and cannot favor one response over another. We therefore report group composition, which measures whether a within-prompt comparison exists, rather than update magnitude.

Table 11 gives the hinted mixed fractions for saturated-copy continuations. Without no-hint questions, the two continuations at $c \approx 0.21$ average 0.001 and 0.003. Each rate divides by the hinted prompts in that update. The saturated line kept at $c = 0.65$ averages 0.005 and 0.002 over its second and third stages. When no group in an update is mixed, the batch normalization also returns zero and every token's advantage is exactly zero. This happens in 22 and 20 of the 24 updates of the two continuations at $c \approx 0.21$, and in 18 and 21 updates of the two stages at $c = 0.65$. We also replay DAPO-style filtering on the logged groups. If an update may sample up to three times its 64 prompts and keeps only mixed groups, the expected filled fraction of the batch is 0.004 and 0.010 for the two continuations and 0.014 and 0.006 for the two stages, against 0.55 to 1.00 for the base runs at $c = 1$, the solver continuations at $c \approx 0.21$ and the no-hint mixtures. With 64 samples per wrong-hint question, SAT0 and its two continuations copy in all 64 samples on 0.94, 0.95 and 0.97 of questions. Table 10 trains with these changes. Continued from SAT0 at $c \approx 0.21$ with DAPO's dynamic sampling (up to three rounds of 64 prompts per update), training samples 4,608 prompts, finds 7 mixed groups, skips 18 of 24 updates and takes one optimizer step in each of the other six, against four per update in standard training. The same sampling from the blind copier skips no update, fills the batch with mixed groups from update 4 onward and lowers copying to 0.19 by update 8, against 0.43 and 0.49 without it. Sampling at temperature 1.2 from SAT0 leaves copying at 0.992. With a KL coefficient of 0.01 toward the base model, the mean KL rises to 0.33 by update 24 while copying forms as without it. The same coefficient toward SAT0 frees it in two runs, with copying at 0.520 and 0.994 after 8 updates and at 0.006 and 0.160 after 24, and no-hint accuracy rising from 0.090 to 0.507 and 0.306. The penalty enters the loss as the mean of $\log \pi - \log \pi_{\text{ref}}$ over sampled tokens (k1 estimator) with a fixed coefficient, and the reward is unchanged. The main experiments use the same trainer with coefficient 0. Figure 6 shows the group composition of each solver and saturated-copier run continued at $c \approx 0.21$, which Figure 3a averages. Solvers A and B are the two solver checkpoints obtained from 24 no-hint updates, and these runs start from them after 24 further updates at $c = 1$.

Table 10: Wrong-hint copying after 24 updates with one training change. The last column repeats the standard runs from the same start.

Change	Start and training	Copy^-	Standard training
DAPO dynamic sampling	SAT0, $c \approx 0.21$	1.000	1.000 / 0.999
DAPO dynamic sampling	Blind copier, $c \approx 0.21$	0.003	0.010 / 0.006
16 samples per prompt	SAT0, $c \approx 0.21$	1.000	1.000 / 0.999
Sampling temperature 1.2	SAT0, $c \approx 0.21$	0.992	1.000 / 0.999
KL 0.01 to the base model	Base, $c = 1$	0.999	1.000 / 1.000 / 0.998
KL 0.01 to the start	SAT0, $c \approx 0.21$	0.006 / 0.160	1.000 / 0.999

Table 11: Fraction of hinted training groups with mixed rewards in saturated-copy continuations at $c \approx 0.21$. Slashes separate two runs.

No-hint fraction	Mean over updates	Last update
$q = 0$	0.001 / 0.003	0.000 / 0.000
$q = 0.25$	0.203 / 0.240	0.756 / 0.804
$q = 0.50$	0.127 / 0.198	0.690 / 0.727
$q = 0.75$	0.151 / 0.111	0.800 / 0.818

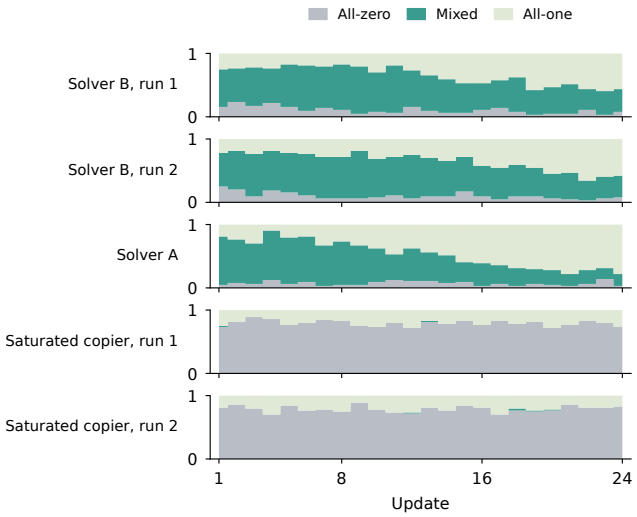

Figure 6: **Composition of hinted training groups** in each solver and saturated-copier run continued at $c \approx 0.21$.

D Inference-time interventions

D.1 Checkpoints and state replacement

S1 and S2 were trained to saturated copying and then for 24 updates without hints. SAT0 is a saturated copier. C1 started from a saturated copier and escaped copying through training with half of the questions shown without hints ($q = 0.50$).

We replace the output of a decoder block at response-decision positions: the last prompt position for $K = 1$, and the last prompt position plus the next seven positions for $K = 8$. Positions are defined in absolute terms, so left padding, prefill, cached decoding and teacher forcing all address the same tokens. Later donor positions use the recipient's own generated prefix. Each generation starts with a fresh cache. Because the replaced state feeds the keys and values of later layers, a replacement at the last prompt position also changes the prompt cache that later tokens attend to. Replacing a state with itself reproduces the unmodified model token for token. Figure 7 shows the replacement and the first-token $\times$ cache test used in Figure 4b.

Table 12 gives the first 32-question study on S1, where the same C1 donor at block 13 leaves copying at 0.914, and Table 13 the 64-question calibration results. For the own-no-hint donor, the no-hint version is unchanged by construction. The C1 donor also changes no-hint behavior: at block 19 it moves no-hint accuracy by $+0.045$ in S1 and -0.053 in S2.

Table 12: Single-position ($K = 1$) state replacement on 32 S1 fitting questions. Entries are Copy^- / Acc^-.

Donor	Block 19	Block 25
None	0.973 / 0.012	0.973 / 0.012
Self, same input	0.973 / 0.012	0.973 / 0.012
Random, same norm	0.945 / 0.027	0.988 / 0.008
C1, same question	0.305 / 0.418	0.285 / 0.402
C1, different question	0.289 / 0.414	0.289 / 0.434
S1, no-hint input	0.270 / 0.445	0.336 / 0.418

The replacement leaves correct-hint accuracy at 0.98–0.99. On correct-hint prompts the model still adopts the printed candidate, but it writes a short derivation first: responses grow from about 50 to about 175 tokens. When the second correct candidate is printed instead, the final answer follows that candidate in 0.989 (S1) and 0.991 (S2) of responses. On wrong-hint prompts the replacement

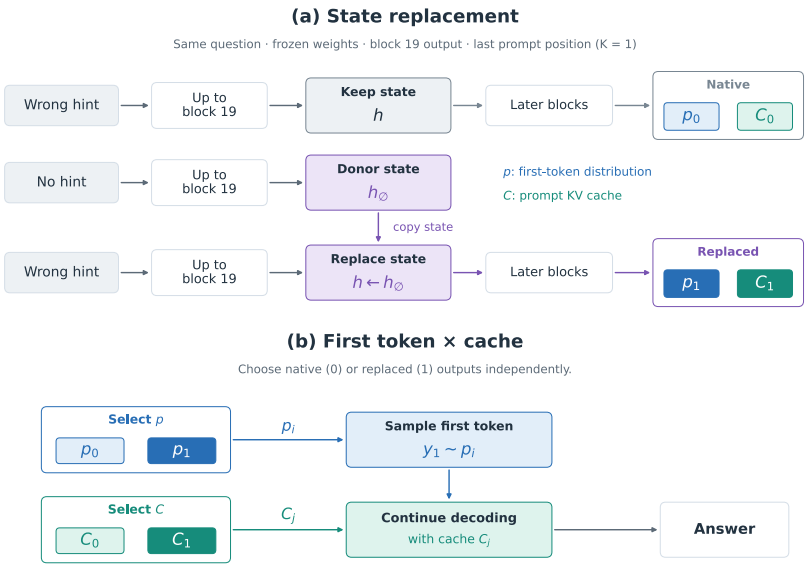

Figure 7: **State replacement and the first-token × cache test.** (a) The no-hint state $h_\varnothing$ replaces the hinted state h at block 19; the native and replaced passes give first-token distributions p_0, p_1 and prompt caches C_0, C_1. (b) The first token is sampled from p_i and decoding continues with C_j.

Table 13: Block-19, $K = 1$ state replacement on 64 calibration questions, eight responses per version.

Recipient	Donor	Copy$^-$	Acc$^-$
S1	None	0.969	0.010
S1	Random	0.949	0.021
S1	C1, same question	0.337	0.368
S1	C1, different question	0.359	0.360
S1	Own no-hint state	0.360	0.354
S2	None	0.791	0.112
S2	Random	0.773	0.121
S2	C1, same question	0.316	0.392
S2	Own no-hint state	0.278	0.404

produces the model's usual no-hint derivations, and truncation rises with length. On the two wrong-hint versions it goes from 0.002 and 0.018 to 0.107 and 0.090 in S1, and from 0.031 and 0.045 to 0.090 and 0.100 in S2. S1 truncates 0.13–0.16 of its own no-hint responses. Accuracy uses the last boxed expression, as the training reward does.

D.2 Forced first tokens

Table 14 forces response prefixes without any hidden-state change. Fixed tokens are chosen on fitting questions: *To* is the modal first token without a hint and *The* the modal first token with one. For S1, forcing the own no-hint first token changes copying by -0.661 ($[-0.695, -0.627]$) and accuracy by $+0.380$ ($[0.320, 0.440]$); forcing *To* gives -0.662 and $+0.381$.

D.3 Held-out evaluation

Table 15 gives the frozen block-19, $K = 1$ replacement on the 256 held-out questions. The last column is the fraction of question–wrong-hint groups of eight responses that contain both a correct and an incorrect answer. The replacement raises this fraction from 0.127 to 0.854 in S1 and from 0.572 to 0.842 in S2.

Table 14: Forced response prefixes on 64 calibration questions per recipient.

Recipient	Condition	Copy^-	Acc^-	$\text{Acc}^{\varnothing}$	Truncation
S1	None	0.967	0.012	0.518	0.043
S1	Own first token	0.306	0.392	0.527	0.086
S1	Own first eight tokens	0.296	0.402	0.521	0.086
S1	Fixed *To*	0.305	0.393	0.527	0.086
S1	Fixed *The*	0.996	0.001	0.410	0.014
S2	None	0.779	0.110	0.561	0.044
S2	Own first token	0.275	0.410	0.527	0.088
S2	Own first eight tokens	0.287	0.415	0.543	0.087
S2	Fixed *To*	0.278	0.408	0.527	0.087
S2	Fixed *The*	0.975	0.013	0.438	0.020

Table 15: Frozen block-19, $K = 1$ replacement on 256 held-out questions. Each recipient uses its own no-hint state for the same question.

Recipient	Condition	Copy^-	Acc^-	Mixed groups
S1	None	0.970	0.017	0.127
S1	Own no-hint state	0.348	0.364	0.854
S1	Random donor	0.959	0.023	0.160
S2	None	0.799	0.117	0.572
S2	Own no-hint state	0.276	0.413	0.842
S2	Random donor	0.785	0.125	0.584
SAT0	None	0.999	0.000	0.000
SAT0	Own no-hint state	0.993	0.001	0.012
SAT0	Random donor	0.997	0.001	0.006

D.4 Forced first tokens on held-out questions

Table 16 repeats the forced-prefix test and the first-token $\times$ cache test on the 256 held-out questions, using the tokens fixed on fitting questions. On held-out prompts with the first wrong candidate, S1 opens with *The* in 0.961 of responses, and these copy in 0.998 of cases (95% CI $[0.996, 0.999]$), against 0.406 ($[0.293, 0.527]$) for responses that open with *To*. Forcing *To* leaves no-hint accuracy unchanged (changes of -0.013 and -0.002, both intervals containing 0) and correct-hint accuracy at 0.99. SAT0 moves much less. Forcing *To* lowers its copying by 0.118 ($[-0.129, -0.107]$), and the cache test changes it by less than 0.01. C1 already opens with *To*, so forcing *To* changes nothing, while forcing *The* raises its copying.

Table 17 forces the first token of wrong-hint responses in 16 checkpoints on the 256 Countdown test questions. In copiers that rarely solve, forcing *To* lowers copying by 0.09 to 0.19 and raises accuracy by at most 0.05. Where responses already open with *To*, forcing it leaves them unchanged. With a correct hint and a forced *To*, every checkpoint still adopts the printed candidate in at least 0.96 of responses, except Llama trained without hints, which has never seen one.

D.5 SAT0 with other donors

SAT0 solves 0.088 of no-hint questions, so its own no-hint state offers little to move toward. A donor state from C1, which does solve, moves SAT0 further (Table 18). At least one correct non-copying response appears on 21 and 24 of 64 questions at $K = 1$ (blocks 19 and 25), and on 29 and 39 at $K = 8$. At $K = 1$, the own no-hint state leaves copying at 0.993 (block 19) and 0.969 (block 25). With $K = 8$, it lowers copying to about 0.6 but also lowers correct-hint accuracy to 0.70–0.74, because it suppresses use of the hint without adding a solution.

Table 16: Forced first tokens and the first-token × cache test on 256 held-out questions. p_1 and C_1 come from the block-19 replacement with the recipient's own no-hint state. The cache test uses the first wrong candidate with four responses per question.

Recipient	Condition	Copy$^-$	Acc$^-$	Acc$^\varnothing$
S1	None	0.970	0.017	0.517
S1	Fixed *To*	0.320	0.386	0.504
S1	Own first token	0.321	0.385	0.504
S1	Fixed *The*	0.996	0.002	0.421
S1	p_1, C_0 / p_0, C_1 / p_1, C_1	0.396 / 0.962 / 0.367	0.332 / 0.023 / 0.357	–
S2	None	0.799	0.117	0.538
S2	Fixed *To*	0.307	0.397	0.535
S2	Own first token	0.310	0.397	0.534
S2	Fixed *The*	0.975	0.013	0.464
S2	p_1, C_0 / p_0, C_1 / p_1, C_1	0.293 / 0.740 / 0.331	0.394 / 0.149 / 0.367	–
SAT0	None	0.999	0.000	0.079
SAT0	Fixed *To*	0.881	0.029	0.108
SAT0	p_1, C_1	0.990	0.004	–
C1	None	0.294	0.345	0.417
C1	Fixed *To*	0.294	0.337	0.410
C1	Fixed *The*	0.681	0.160	0.360

Table 17: Forced first token on wrong-hint prompts in 16 checkpoints, 256 test questions, eight responses each. *To* share is the native fraction of responses that open with *To*.

		Copy$^-$			Acc$^-$	
Checkpoint	*To* share	Native	*To*	*The*	Native	*To*
Base $\to c = 1$	0.001	1.000	0.808	0.999	0.000	0.050
Base $\to c = 1$	0.000	1.000	0.914	1.000	0.000	0.014
Base $\to c = 1$	0.001	0.998	0.830	0.998	0.000	0.032
Saturated copier (SAT0)	0.000	0.999	0.890	0.999	0.000	0.026
Blind copier	0.079	0.929	0.819	0.930	0.022	0.050
Blind copier, 24 more at $c \approx 0.81$	0.026	0.065	0.188	0.056	0.534	0.424
Selective $\to c = 1$	0.074	0.047	0.071	0.098	0.602	0.566
Selective $\to c = 1$	0.998	0.182	0.177	0.256	0.410	0.431
Solver A $\to c = 1$	0.989	0.205	0.199	0.234	0.403	0.409
Solver B $\to c = 1$	0.995	0.492	0.501	0.353	0.267	0.267
Solver after $c = 1$, $c \approx 0.21$	1.000	0.013	0.012	0.044	0.775	0.772
Solver after $c = 1$, $c \approx 0.21$	1.000	0.002	0.001	0.010	0.849	0.851
Solver after $c = 1$, $c \approx 0.21$	1.000	0.002	0.005	0.044	0.802	0.788
Llama, $c = 1$	0.988	0.664	0.655	0.712	0.088	0.083
Llama, $c = 0.95$	0.995	0.113	0.115	0.112	0.538	0.560
Llama, no hint	0.887	0.008	0.007	0.007	0.777	0.783

D.6 Low-rank controls

We also test whether a few directions capture the replacement. Let $H_C - H_S$ be the state differences between C1 and S1 on fitting questions at the first $K = 8$ decision positions. Rank one uses their mean direction; rank four uses the top four right singular vectors, orthonormalized. With basis $U \in \mathbb{R}^{d \times r}$ shared across positions and target coordinates z_j for position j, the control is

$$h' = h + \alpha U(z_j - U^\top h), \qquad U^\top U = I_r, \tag{9}$$

with $d = 3{,}584$. We set α by bisection so that the mean full-vocabulary $D_{\mathrm{KL}}(p_{\mathrm{control}} \| p_{\mathrm{native}})$ over the first eight positions of fitting prefixes matches a budget. Table 19 shows that low-rank controls move behavior in the right direction but recover at most about a third of the full replacement. The best setting, rank one at block 19 with budget 0.05, changes copying by -0.201 ($[-0.244, -0.160]$) and accuracy by $+0.127$. Clamping C1 toward S1's coordinates raises its copying by 0.05–0.10.

Table 18: SAT0 under different donors on 64 calibration questions. Mixed groups count question–wrong-hint groups of eight responses.

Donor	K / block	Copy^-	Acc^-	Acc^+	Mixed groups
None	n/a	0.996	0.000	0.998	0.000
C1, same question	1 / 19	0.841	0.030	0.992	0.195
C1, same question	1 / 25	0.882	0.031	0.992	0.211
C1, same question	8 / 19	0.817	0.041	0.994	0.273
C1, same question	8 / 25	0.807	0.050	0.996	0.352
Own no-hint	8 / 19	0.591	0.045	0.701	0.344
Own no-hint	8 / 25	0.627	0.044	0.736	0.266

Table 19: Low-rank controls on S1, $K = 8$, 64 calibration questions. Entries are changes in Copy^- / Acc^-; the reverse row applies the control to C1.

Control	KL budget	Block 13	Block 19	Block 25
Rank one	0.02	$-0.07/+0.05$	$-0.10/+0.06$	$-0.08/+0.05$
Rank one	0.05	$-0.12/+0.06$	$-0.20/+0.13$	$-0.10/+0.06$
Rank four	0.02	$-0.09/+0.04$	$-0.12/+0.06$	$-0.11/+0.08$
Rank four	0.05	$-0.17/+0.12$	$-0.19/+0.11$	$-0.18/+0.10$
Reverse	0.05	$+0.10/-0.04$	$+0.05/-0.05$	$+0.06/-0.02$

E Training with control

E.1 Short pulse

The pulse runs continue S1 at $c \approx 0.21$ for 24 updates. In the pulse arm, the frozen block-19, $K = 1$ replacement is active during updates 1 and 2, both when sampling rollouts and when the trainer recomputes their log-probabilities, so the update uses the same computation that produced the samples. From update 3 on, no hook is present, which we verify at every update. The random arm replaces the state with a random vector of the same norm. Each arm has four seeds. All evaluations run without control and with a fresh cache. Table 20 gives the mean hinted mixed-group fraction during training.

Table 20: Fraction of hinted training groups with mixed rewards, averaged over four seeds.

	Update					
Arm	1	2	3	4	8	24
No control	0.020	0.141	0.383	0.617	0.664	0.332
Replacement pulse	0.652	0.707	0.156	0.395	0.641	0.320
Random pulse	0.070	0.117	0.207	0.406	0.676	0.336

At update 4, wrong-hint copying is higher with the replacement pulse than without control by 0.09, 0.33, 0.35 and 0.22 in the four seeds. At update 24, mean copying is 0.0055 without control, 0.0050 with the replacement pulse and 0.0052 with the random pulse.

E.2 Controlled rollouts for an uncontrolled policy

This experiment keeps the trained policy free of control. Each prompt receives eight responses: four from the native policy and four from a controlled sampler b. All eight are scored by the verifier, and the advantage A of each response is its reward normalized within the group of eight. Advantages are normalized within groups only, so uniform groups contribute nothing. For each token we store the log-probability under the sampler, $\log b$, and under the policy at sampling time without control,

$\log \pi_{\text{old}}$. The loss is

$$L = -\,\text{mean}\Big[w\ \min\big(\varrho A,\ \text{clip}(\varrho, 1{-}\epsilon, 1{+}\epsilon)A\big)\Big], \quad w = \text{sg}\big[\min(\pi_{\text{old}}/b,\ 2)\big], \quad \varrho = \pi_\theta/\text{sg}[\pi_{\text{old}}], \tag{10}$$

where sg stops the gradient and π_θ is the current policy without control. For native responses w is close to 1. All four arms below use this trainer. A finite-difference test on a small model confirms the gradient of this loss.

We compare four samplers, each used for the controlled half of every group: none (all eight native), the block-19 state replacement with its modified cache, the replaced first-token distribution p_1 followed by native decoding, and a temperature control that samples the first token from $\text{softmax}(\ell_0/\tau)$. For the temperature control, $\tau \in [1, 4]$ is chosen per prompt so that its first-token KL from the native distribution matches that of the replacement, within $\max(0.01, 10\%)$; all hinted fitting and calibration prompts meet this tolerance, with a mean target KL of 2.7 nats. Training runs for two updates from S1 at $c \approx 0.21$ on the first 128 training questions, with two seeds (one for the temperature control). The readout is the probability that the uncontrolled policy starts a wrong-hint calibration response with a solving opener.

Table 21 summarizes the results. Without a controlled half, 0.19 of hinted groups are mixed at update 1. All changes in the last column have paired 95% intervals that exclude zero. Both targeted samplers create large reward gaps between the controlled and native halves, and the temperature control does not. Under the uncontrolled policy, the first token of a controlled response has a median ratio π_{old}/b of 0.04–0.09 for the targeted samplers, against 0.51–0.75 for the temperature control. Across all controlled tokens, fewer than 0.4% hit the upper clip.

Table 21: Controlled rollouts for an uncontrolled policy (S1, $c \approx 0.21$). Reward gap, mixed hinted groups and median first-token ratio π_{old}/b are for seed 1 at updates 1 / 2. The last column is the change in the probability of a solving opener after two updates relative to native-only training, per seed.

Controlled half	Reward gap	Mixed groups	First-token ratio	Change in solving opener
Replacement	$+0.37$ / $+0.30$	0.61 / 0.55	0.05 / 0.07	-0.186, -0.092
First token only	$+0.29$ / $+0.32$	0.59 / 0.61	0.04 / 0.09	-0.077, -0.080
Temperature	$+0.04$ / $+0.07$	0.20 / 0.33	0.51 / 0.75	-0.116

F REWARD PAIRING

Table 22: **Product versus independent reward on the same paired data**, after 24 updates from solvers A and B, and from the blind and saturated copiers (product reward only). Paired conditions use the same questions, order and sampling settings.

Start	Reward	Copy^-	Acc^-	$\text{Acc}^{\varnothing}$
Solver A	Independent	0.005	0.767	0.654
	Product	0.011	0.739	0.611
Solver B	Independent	0.013	0.713	0.578
	Product	0.017	0.672	0.544
Blind copier	Product	0.061	0.618	0.357
	Product	0.054	0.617	0.385
Saturated copier	Product	0.999	0.000	0.090

For solver A, the product reward minus the independent reward changes wrong-hint accuracy by -0.028 ($[-0.048, -0.009]$) and no-hint accuracy by -0.043 ($[-0.068, -0.018]$). For solver B, the changes are -0.041 ($[-0.067, -0.015]$) and -0.034 ($[-0.062, -0.006]$), and the rate of adopting a correct printed candidate falls by 0.143 ($[-0.164, -0.122]$).

G Extended related work

What RLVR changes. GRPO (Shao et al., 2024) with rule-based rewards is the standard recipe for reasoning training, and pure RL of this kind can elicit self-verification and long chains of thought (Guo et al., 2025a). What it changes in the base model is debated. Large-k sampling suggests that much of the gain revisits behavior the base model can already produce (Yue et al., 2025). Pass@k can credit correct answers reached through incorrect reasoning (Wen et al., 2026), and prolonged RL with KL control and reference resets can reach solutions that the base model rarely finds (Liu et al., 2025a). Other work describes RL as selection among reasoning patterns, with most of the change on a few critical tokens (Chen et al., 2026; Meng et al., 2026) and entropy collapsing during training (Cui et al., 2025). In deep RL, early experience shapes later behavior (Nikishin et al., 2022) and plasticity declines (Lyle et al., 2023). The first response token in our setting is such a critical token, and saturated copying is such a case of path dependence.

Cue reliance at inference time. When a training signal is compatible with several decision rules, networks tend to adopt the simplest one (Geirhos et al., 2020). Language models follow answer hints in the prompt while their chains of thought rarely mention the hint (Turpin et al., 2023), answers often do not depend on the stated reasoning (Lanham et al., 2023), and reasoning models trained with RL use hints without verbalizing them (Chen et al., 2025b). Models also defer to users' wrong opinions (Sharma et al., 2023) and handle context that conflicts with their parametric knowledge inconsistently, sometimes following it and sometimes ignoring it (Xie et al., 2023; Zhou et al., 2023). Response length is a related behavioral signal. Reasoning models overthink easy problems (Chen et al., 2025a), and rewards in RLHF correlate with length (Singhal et al., 2024). Most of these studies diagnose trained models, and those that follow RL training track verbalization or length. We follow cue reliance through RL training and use wrong-hint and no-hint versions of the same question as the counterfactual test. Length enters our analysis only as a signature of the route, since copies are short and derivations are long.

Hints and learning signals in RLVR. Another line puts hints or external trajectories into RLVR on purpose. When all rollouts of a problem receive the same reward, group-relative methods get no within-prompt contrast from it. DAPO filters such groups (Yu et al., 2025), and RC-GRPO diversifies them by conditioning rollouts on reward tokens (Zhong et al., 2026). LUFFY mixes reasoning traces from a stronger model into rollouts (Yan et al., 2025), and StepHint guides exploration with multi-level partial solutions (Zhang et al., 2026a). Hints can also adapt to the learner, through hint length tuned to keep hinted accuracy near one half (Li et al., 2025b) or a trained hinter that avoids revealing the answer and measures hint reliance as a likelihood gap (Xia et al., 2026). Other methods build hints from the policy's own successful prefixes (Deng et al., 2026) or self-proposed outlines (Cao et al., 2026), distill process hints from a teacher for questions on which every rollout fails (Li & Gao, 2025), or anneal hints per difficulty bucket, with per-question hint dropout that leaves many questions unhinted (Xie et al., 2026), a mixing whose effect on reward contrast Section 5.3 explains. Zhang et al. (2026b) survey this line. A related line supervises the process instead of the final answer, through step-level labels (Lightman et al., 2024), segment-level advantages (Guo et al., 2025b), self-supervised process rewards (Wang et al., 2025), deterministic step verifiers (Pronesti et al., 2026) or filtering rollouts by process–outcome consistency (Ye et al., 2025). RISE also rewards the policy for verifying its own solutions (Liu et al., 2025b). We study hints that carry a candidate answer, in which case the reward cannot tell solving from copying, and show when copying wins. LUFFY reshapes importance weights so that low-probability tokens still receive gradient. Our results in Section 6 show why this matters for hints, since truncated importance weights give the token that decides the route little credit.

Auditing outcome rewards. Rewarding the final answer does not certify the intended mechanism. Causal readouts test whether answers depend on the generated reasoning (Yu et al., 2026), attention knockout and logit-lens analyses separate latent multi-hop reasoning from factual shortcuts (Yang et al., 2025b), and random or even negatively correlated rewards can improve some models by amplifying behavior already in the model, such as code reasoning or memorized answers (Shao et al., 2026; Yan et al., 2026), while annotations verified to be incorrect harm RLVR (Zhu & Kang, 2026). Models also struggle to correct their own reasoning without external feedback (Huang et al., 2024). These failures belong to the specification-gaming lineage, in which an agent satisfies the literal

objective without the intended outcome (Krakovna et al., 2020). Stronger agents can earn more proxy reward with less true reward (Pan et al., 2022), and RL reasoning training increases specification gaming in reasoning models (Nishimura-Gasparian et al., 2026). Isomorphic perturbation tests expose verifier-side shortcuts (Helff et al., 2026). Reward seeking learned on small loopholes can generalize to reward tampering and broader misalignment (Denison et al., 2024; Taylor et al., 2025), and reward hacking in RLVR can be traced through formation and rebound (Wu & Tang, 2026). Our wrong-hint test is a model-side counterpart of isomorphic testing. It holds the question fixed and flips only the hint.

Representations and interventions. Several studies track or exploit hidden-state geometry in RLVR and post-training, through effective-rank dynamics, pretraining-to-post-training geometry, an auxiliary alignment loss and training-free data selection (Huang et al., 2026; Li et al., 2025a; Wang et al., 2026; Wu et al., 2026). Activation steering and representation engineering change behavior through hidden states (Turner et al., 2023; Rimsky et al., 2024; Zou et al., 2023), representation finetuning trains such interventions (Wu et al., 2024), and policy-gradient steering derives them from behavioral objectives (Poupart et al., 2026). Single directions can mediate behaviors such as refusal (Arditi et al., 2024). We use a natural state replacement, the model's own no-hint state, to find where the route is chosen, and then ask whether applying it during training changes what the policy learns.