DECODABLE NEED NOT BE REWARD-REACHABLE: A CONTROLLED STUDY OF DIRECT-ANSWER LEARNING IN RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) depends on both the answers a policy samples and the update that uses their rewards. Guaranteeing correct answers in training removes a coverage problem, but leaves open whether the update can learn an answer that supervision can teach. We isolate this distinction with execution-labeled Yes/No questions about runtime errors in short Python programs. On a list-index task, Qwen3-8B representations support linear label decoding, and same-data supervised fine-tuning (SFT) reaches margin AUC 0.927. Forcing both final answers into every group restores reward variation and a nonzero answer-logit signal, yet the tested centered update reaches only 0.558; the same intervention succeeds on dictionary lookup. Changing the loss or weighting rule improves index readouts under matched sampling. The primary exact-gradient comparison remains inconclusive, while a separate pre-specified error-focused rule learns the direct index answer in all three seeds and meets its local-sufficiency criterion. Reasoning provides a distinct route: strong Qwen answers are already expressed before task-specific training, while direct answers remain weak after reasoning-channel training. The results separate recoverable label information, answers supervision can teach, and answers a specified reward update learns.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) trains a policy from automatically checked outcomes, using sampled responses to decide which outputs to make more likely. Its success depends on both what the policy produces and how the update uses the resulting rewards, so a weak training endpoint can have several explanations, each calling for a different remedy. The policy may rarely sample a correct answer, motivating better coverage (Yue et al., 2025). The model may lack useful information for the task, motivating changes to what it represents (Chen et al., 2026). Or correct answers may be available while the update fails to learn to produce them. Telling these explanations apart requires more than a low reward at the end of training.

We examine a controlled case using three complementary measurements. A linear probe tests label decodability, same-data supervised fine-tuning (SFT) tests whether the direct answer can be learned, and forced final answers guarantee correct and incorrect samples in every training group. We call a readout *reward-reachable* when the tested outcome-reward update learns it under the specified model, task, channel and budget. The central question is whether a decodable, SFT-learnable answer becomes reward-reachable once correct final answers are always available.

Short program execution provides such a setting. The model reads a Python snippet and answers Yes or No to whether it raises a runtime error; execution supplies the label. For example, `vals[8+2-13]` is valid on a five-element list because the computed index -3 selects its third element from the end. On Qwen3-8B, the index label is linearly decodable and SFT learns the short answer, while standard group-relative training leaves its discrimination weak.

That contrast alone leaves open a simpler explanation: the reward update may rarely see a useful comparison. Equal rewards within a group give zero group-relative advantages (Nie et al., 2026); approximately 98% of standard-GRPO index groups have this property. Because there are only two

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.

final answers, we can guarantee a correct and an incorrect answer by forcing one Yes and one No into every group. This restores reward variation and a nonzero answer-logit signal. Yet index reaches margin AUC 0.558, versus 0.927 after SFT, while the same intervention teaches a dictionary-lookup control. Correct final answers are present in every group, yet learning them still depends on the task and update.

With final-answer coverage guaranteed, we ask which changes to the update make the direct answer learnable. Loss swaps and scaling without mean-centering give stronger index readouts under the matched sampling protocol, with scaling also helping string index. A narrower exact-gradient test holds the positive log-probability coefficient fixed while removing incorrect-answer credit, but its primary comparison remains inconclusive. The separate pre-specified follow-up instead changes prompt weights under the fixed forced sampler. Its complete error-focused rule meets the local-sufficiency criterion on index, while changing both relative allocation and total update mass. These results exhibit a successful update in this setting without identifying removal of negative credit as the reason it succeeds.

A different intervention changes the output channel. Before task-specific training, Qwen already expresses strong index answers after chain-of-thought (CoT). Trained checkpoints retain strong reasoned readouts while direct answers remain weak, separating access through reasoning from learning to express the answer directly.

Our contributions.

1. **A controlled learning gap.** A decodable, SFT-learnable answer remains weak under the tested reward update despite guaranteed final-answer coverage (§2.2).
2. **Controlled update comparisons.** Tests under matched sampling separate an inconclusive primary exact-gradient contrast from the local success of a complete error-focused rule (§3).
3. **Access versus direct-answer learning.** Reasoning-channel comparisons distinguish already expressed answers from direct-readout learning; follow-up tasks and models delimit the case (§4.1–4.2).

Figure 1 presents the task and principal contrast.

2 A Learnable Task with Both Final Answers

2.1 Task, answer readout, and comparison

The principal comparison uses Qwen/Qwen3-8B, not the separately named Base checkpoint, on short Python programs followed by a Yes/No question about execution errors. Execution supplies the labels. Index validity depends on a computed list index, including valid negative indices; key selects a dictionary key with a conditional expression before lookup. Key is an intervention positive control, not a difficulty-matched task. Answer-only prompts are raw task text without a chat template. The controlled coverage, RAFT and exact-update experiments read and sample from the two Yes/No logits; the four-token allowance belongs to the generative answer-only protocols. Other task families and recorded examples are defined in Appendix A.

The *answer readout* is the model's own Yes/No output. Margin AUC measures whether the Yes-minus-No logit separates Yes-labeled from No-labeled examples; 0.5 denotes chance ranking; values below 0.5 indicate an inverted margin. AUC is distinct from realized answer accuracy. A linear probe is a separately fitted hidden-state classifier. SFT tests whether training can teach the model's answer readout. An update *installs* a readout when it meets that experiment's success criterion; if the tested outcome-reward update does so, the readout is *reward-reachable* under the specified update rule, model, task, channel and budget.

Controlled final-answer coverage. Each group of eight contains one forced Yes, one forced No and six policy draws. Both final actions therefore occur and binary reward varies in every group. Correct parsed answers earn 1, incorrect answers 0, and the generative parser assigns unparseable completions -0.5. The two-action controls cannot produce unparseable completions. Forced answers enter at unit weight, deliberately biasing the action distribution; this is not an unbiased esti-

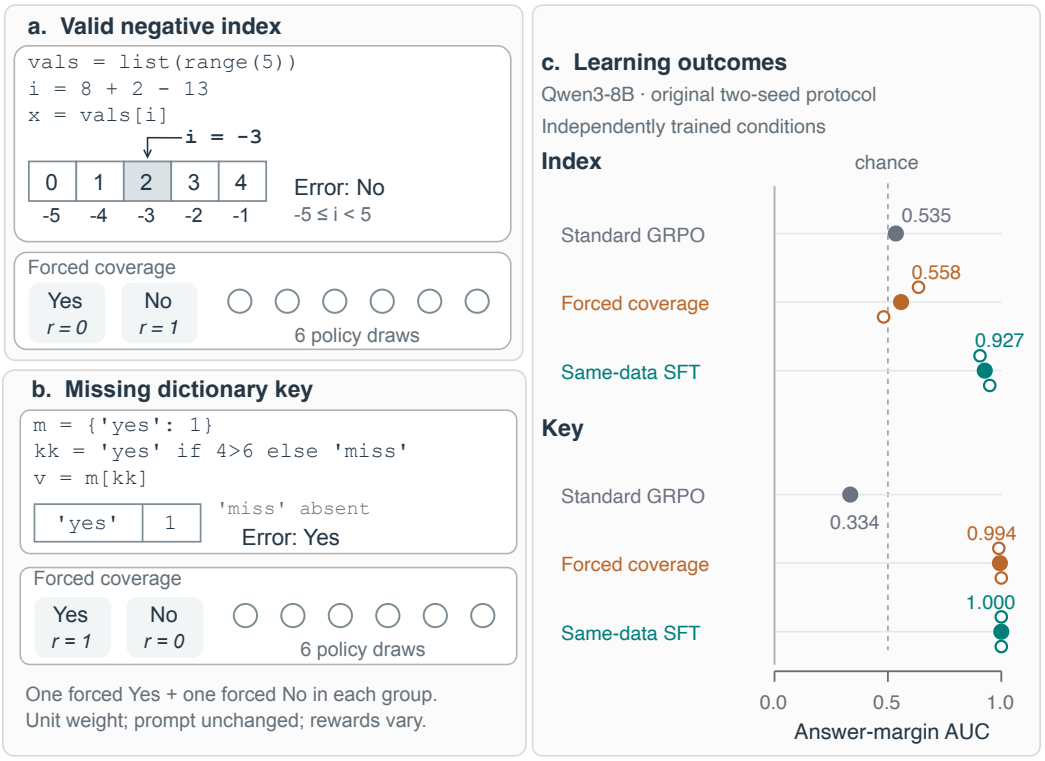


Figure 1: **Answer coverage helps the two tasks differently.** (a,b) Illustrative execution-labeled programs and forced-coverage groups. (c) Answer-margin AUC under the original Qwen3-8B two-seed protocol (Tables 10–11), separate from later three-seed studies. Filled markers show reported means; hollow markers show seeds 42 above and 7 below. GRPO has means only. Coverage succeeds on key, an intervention positive control rather than a difficulty-matched task, but leaves index far below same-data SFT.

mator of ordinary on-policy GRPO. Prompts and answer positions are unchanged. The intervention covers final answers, not intermediate reasoning trajectories.

Same-data SFT tests learnability, not equal-feedback efficiency. Appendix A.4 specifies arm matching and optimizer differences; Appendix A gives each experiment's implementation, precision and measurements.

2.2 Coverage leaves a supervised–reward gap

Same-data SFT learns the Qwen3-8B index readout at margin AUC 0.927. Forced coverage tests whether supplying both final answers suffices for reward training: it eliminates zero-reward-variance groups and raises the mean absolute gold-answer-logit derivative from 0.010 to 0.33 (not a parameter-gradient norm). Yet index reaches only 0.558 across two seeds ($0.635/0.481$), while key reaches $0.989/1.000$ under the same intervention (Tables 10–11; Appendix C.1).

A separate three-seed protocol also leaves sampled forced coverage below index's joint installation rule while learning key (Table 2; protocol in §3.3). The remaining index gap motivates examining how the update uses the available answers.

Other controls. Det3mod remains weak under answer-only SFT (0.537). The inverted mf_heap_pop_empty control reaches 1.000 under coverage, 0.994 under weak importance sampling, and 1.000 under RAFT: a mis-signed readout alone is insufficient to explain the gap. On index, importance sampling restores neither the answer-logit statistic (0.0008) nor the readout (0.482), making it a weaker coverage control (Appendix C.1).

2.3 WHAT THE PROBE ESTABLISHES

A separate probe-control protocol finds label AUC 0.885, or 0.853 after regressing out the declared format covariates. Residualizing the computation's operands reduces probe AUC to 0.524, while a linear classifier using those operands alone reaches 0.935; erasing a probe direction does not block SFT (Appendix B.3). These controls establish linear decodability without showing that the base model has precomputed the answer or identifying the route used by SFT. A nearly constant short answer can coexist with decodable labels, so this dissociation and the inconclusive checkpoint intervention (Appendix B.4) do not identify active suppression or a causal probe route.

3 LEARNING UNDER ALTERNATIVE UPDATES

With both final answers available in every group, the remaining question is how the update uses that comparison. We first vary the loss or weights under matched sampling, then examine incorrect-answer credit and prompt-level weighting separately.

3.1 LEARNING UNDER THE SAME SAMPLING PROTOCOL

Correct-completion cross-entropy. Can a different loss learn under the same final-answer sampling protocol? We apply cross-entropy to correct completions, matching prompt order, forced Yes/No slots and group budget to coverage training. The other six answers are sampled from each arm's current policy, so shared random numbers do not fix realized rollouts once policies diverge. In this binary task, correct completions are gold answers. Table 1 reports all seven families' mean and seed-level AUCs.

Table 1: **Correct-completion loss on the Qwen3-8B matched forced-sampling protocol.** The RAFT-style loss (Dong et al., 2023) uses cross-entropy on correct completions. Prompt data and group budget are fixed at $2{,}400$ groups ($19{,}200$ sampled answers) per seed. Entries are reported margin AUCs; RAFT gives the mean followed by per-seed values (three seeds for index, key, Regex match and Call raises; two elsewhere). †strindex was screened after pre-registration under pre-declared criteria. The post-hoc and pre-registered family sets are distinguished in the text.

Family	Standard GRPO	Coverage	RAFT mean	RAFT per-seed AUCs	SFT
index	0.535	0.558	0.782	(0.96/0.57/0.82)	0.927
key	0.334	0.994	1.000	(1.0/1.0/1.0)	1.000
Regex match	0.499	0.651	0.765	(0.69/0.74/0.86)	0.722
Call raises	0.798	0.740	0.851	(0.87/0.85/0.83)	0.830
Return type	0.961	1.000	0.997	(0.99/1.0)	1.000
String index†	0.419	0.598	0.746	(0.93/0.56)	0.869
Determinant mod	0.599	0.459	0.600	(0.59/0.61)	0.537

The post-hoc mean across index, strindex and regexgroup is 0.764, with some seeds still weak; regexgroup's coverage AUC 0.651 lies just above the 0.65 cutoff. The pre-registered four-family set (index, key, regexgroup and runtime_prop) reaches 0.849, including key, which coverage already solves. Correct-completion loss thus gives higher mean AUC for these sets under the matched sampling protocol.

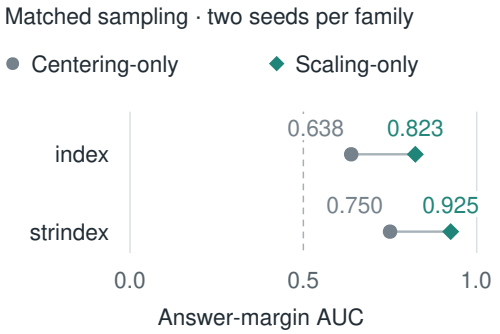
Changing the weighting rule. Scaling-only divides reward by its within-group standard deviation, giving parsed incorrect answers zero weight. Centering-only subtracts the mean without that division, giving below-mean answers negative weights (Appendix G.2). Under the same sampling protocol, scaling-only reaches mean AUCs $0.823/0.925$ on index/strindex with all four family–seed cells reaching AUC ≥ 0.78. Centering-only reaches lower means of $0.638/0.750$, with heterogeneous strindex endpoints of $0.585/0.914$. Both incorrect-sample treatment and correct-sample weights change, so this is not a one-term ablation (Figure 2). The RLOO centering control also leaves index weak, with family-dependent endpoints (Appendix C.3). These earlier AUC-only criteria differ from the joint behavioral rule below.

	Correct ($r = 1$)	Incorrect ($r = 0$)
Centering	$1 - \mu$	$-\mu$
Scaling	$1 / \sigma$	0

μ: group mean; σ: group std.

Parsed binary (0/1) rewards.

(a) Full weighting rules



(b) Reported endpoints

Figure 2: **Weighting rules under matched sampling on Qwen3-8B.** (a) For binary rewards, centering uses $r - \mu$ and scaling uses r/σ, with within-group mean μ and standard deviation σ; both correct- and incorrect-answer weights change. (b) Two-seed mean answer-margin AUCs under the matched sampling protocol, distinct from Figure 1 and the exact-gradient matrix.

3.2 From answer-level signs to prompt-level weighting

The weighting-rule contrast motivates distinguishing the sign of credit on an answer from the weighting of updates across prompts. In the binary-softmax abstraction, let p be the correct-answer probability and Δ the correct-minus-incorrect log-probability margin. Then

$$\nabla_\theta \log p = (1 - p)\nabla_\theta \Delta, \qquad -\nabla_\theta \log(1 - p) = p\nabla_\theta \Delta.$$

Within a prompt, rewarding the correct answer and penalizing the incorrect answer yield updates along the same margin gradient, with different probability-dependent coefficients. Across prompts, changing these coefficients can change the aggregate update; the identity does not determine the training endpoint (Appendix G.1).

A separate weighting-matched base-state test holds correct-sample weights fixed. Adding the incorrect-answer negative-advantage term reduces cosine alignment with the supervised update from approximately 0.92 to 0.32–0.64 on Qwen3-4B, and by 0.21–0.30 on Qwen3-14B from a lower baseline (Appendix G.3). These local measurements motivate the following Qwen3-8B tests of learned readouts.

3.3 Exact-gradient primary comparison

Does removing incorrect-answer credit change learning when its positive coefficient is held fixed? We compare five Qwen3-8B updates on index and key, with three seeds and 300 steps each, using 4-bit NF4 and bf16 computation. Exact IID centering averages over independent policy draws and the random retained-group denominator; exact forced sampling includes both forced answers. Fixed-denominator centering replaces the random normalization by its fixed batch size. Matched correct-only retains the IID update's detached positive log-probability coefficient. Sampled forced coverage supplies the corresponding empirical reference. Appendix D.1 defines all five losses and settings. We measure both margin AUC and two-answer gold probability.

A seed meets the behavioral criterion only if its Clopper–Pearson 95% accuracy lower bound and mean two-answer gold probability both exceed 0.5. A condition is installed with at least two qualifying seeds and mean margin AUC ≥ 0.80; it is not installed with at most one qualifying seed and mean AUC ≤ 0.65; otherwise it is inconclusive.

The primary comparison remains inconclusive. Exact IID centering does not install index ($0/3$ behavioral seeds; AUC 0.489). Matched correct-only credit meets the behavioral criterion in $2/3$ seeds, but AUC 0.756 falls below the 0.80 installation threshold, so it remains *inconclusive* (Table 2). The pre-specified reference checks pass. Key also fails under exact IID and fixed-denominator centering, so failure under these updates is not specific to index. The comparison does not identify the weighting mechanism or establish equivalence between updates.

Table 2: Exact-gradient comparison on Qwen3-8B at step 300. Seeds is the number, out of three, meeting the behavioral criterion; AUC is the mean answer-margin AUC. Outcome applies the joint rule above. The sampled-coverage row uses this 30-cell protocol, not Table 10's two-seed protocol.

	index			key		
Update	Seeds	AUC	Outcome	Seeds	AUC	Outcome
Exact IID centered	0/3	0.489	not installed	0/3	0.397	not installed
Exact forced-sampler	0/3	0.702	inconclusive	3/3	0.996	installed
Sampled forced coverage	0/3	0.620	not installed	3/3	1.000	installed
Matched correct-only	2/3	0.756	inconclusive	3/3	0.998	installed
Fixed-denominator centered	0/3	0.472	not installed	0/3	0.406	not installed

3.4 Separate prompt-allocation follow-up

The exact-gradient primary comparison leaves open whether a different probability-dependent weighting can learn index under the fixed forced sampler. The follow-up changes the multiplier applied to each prompt's exact forced-sampler coefficient:

$$w_\alpha(p) = e_F(p)[1 + \alpha(1 - 2p)], \qquad \alpha \in \{-1, 0, +1\}.$$

Here p is the current two-answer probability of the execution-correct answer and $e_F(p)$ the exact forced-sampler coefficient (Appendix D.2). The multipliers $2p$, 1, and $2(1-p)$ define correctness-focused, reference, and error-focused weighting, respectively. Their registered labels are reverse, even, and SFT-signed. The reference is not uniform across prompts because $e_F(p)$ varies; error-focused weighting is not ordinary SFT.

The detached coefficient multiplies the live correct-minus-incorrect log-probability margin. All weights are nonnegative: $\alpha = -1$ changes allocation across prompts, not an individual gradient's direction. Weights are not normalized to hold total update mass fixed. Initialization, optimizer, examples, data order and arm-invariant RNG are held fixed. Six $\alpha = 0$ conditions reuse the verified-identical exact forced-sampler runs; twelve $\alpha = \pm 1$ conditions are additional runs.

Table 3: Allocation follow-up on Qwen3-8B at step 300. Seeds counts runs meeting the behavioral criterion; AUC is the seed mean and Outcome applies Table 2's joint rule. The reference row ($\alpha = 0$) reuses that table's exact forced-sampler runs. Seed-level metrics are in Table 13.

	index			key		
Allocation α	Seeds	AUC	Outcome	Seeds	AUC	Outcome
-1	0/3	0.475	not installed	0/3	0.543	not installed
0	0/3	0.702	inconclusive	3/3	0.996	installed
$+1$	3/3	0.928	installed	3/3	1.000	installed

Local sufficiency. The complete error-focused rule learns the index readout in all three seeds, whereas the reference is inconclusive and the correctness-focused rule does not install it (Table 3). Key installs under both reference and error-focused weighting. The registered local criterion also requires error-focused minus reference to be consistently positive, and reference minus correctness-focused to be nonnegative, for both final mean gold-answer probability and margin AUC; both requirements hold (criteria in Table 13). This establishes local sufficiency of the complete rule, whose unnormalized weights change relative prompt allocation and total update mass together. The index greedy accuracies are $0.865/0.830/0.695$ (Table 13), so installation does not imply near-perfect accuracy.

The stricter criterion is unmet. The separate *strict-trajectory* criterion requires consistently positive adjacent allocation contrasts for both final endpoints. The reference and correctness-focused allocations finish at the same mean gold-answer probability, 0.5, despite different AUCs, so this criterion is not met. Figure 3 shows the distinction across the four retained checkpoints. The criterion concerns allocation ordering at the final endpoint, not monotonic progress at every training step. The original exact-gradient primary comparison remains inconclusive.

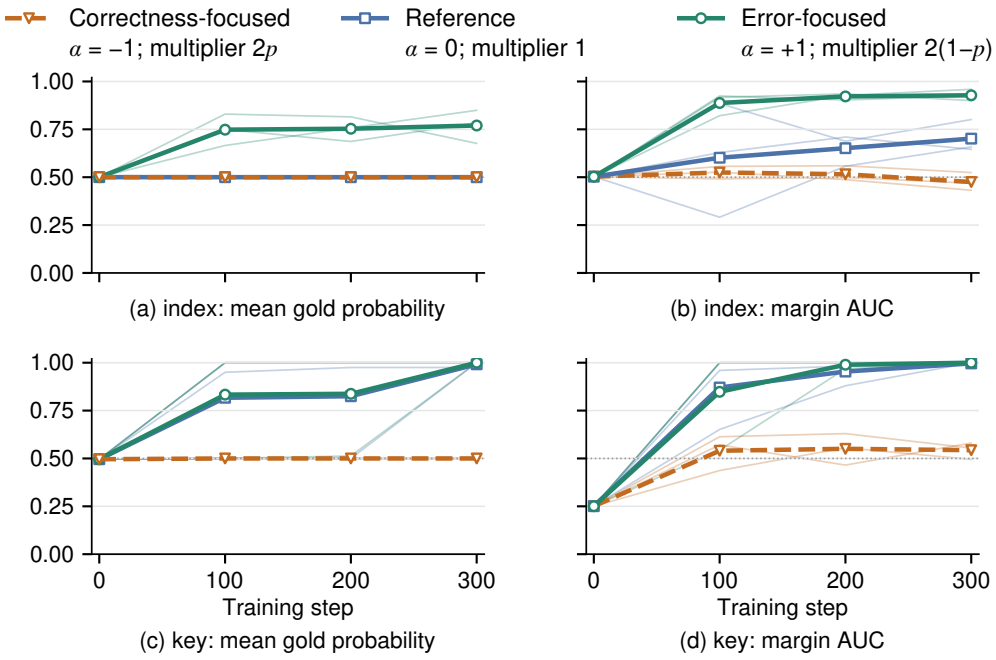


Figure 3: **Recorded checkpoints for the allocation follow-up.** The panels show all 18 index/key runs of Table 13. Thin lines show the three seeds and thick lines their means (correctness-focused means dashed) at steps 0, 100, 200, and 300; lines connect only these stored checkpoints. The legend shows each allocation's multiplier $w_\alpha(p)/e_F(p)$.

4 Reasoning-Channel Access and Task Scope

4.1 Answers expressed before task-specific training

The reasoning-channel comparison asks a different question from the answer-only update tests: can the model express an answer through a longer response when its direct readout remains weak? Before task-specific training, Qwen3-8B already has reported post-CoT AUC 0.998 on index and 0.996 on det3mod. The CoT arm adds a reasoning instruction and up to 512 reasoning tokens while retaining the outcome reward; the answer-only arm uses raw task text without a chat template and permits four tokens. This comparison changes format and budget together.

Table 4 reports starting readouts and trained endpoints with each protocol's estimators. Llama's pre-specified installation rule requires post-CoT AUC ≥ 0.93 and direct AUC < 0.65.

The trained Qwen checkpoints have high post-CoT readouts but low direct readouts. Det3mod is particularly informative: its trained post-CoT AUC is 0.984 and its direct AUC is 0.549, while the separate answer-only SFT recipe reaches 0.537. Because a strong post-CoT readout is already observed before task-specific training, these results demonstrate channel-specific access rather than the creation of that ability by RL. They do not estimate a paired training gain across the baseline and trained protocols. The same $k = 8$ evaluation of the trained index checkpoints also gives sampled accuracies $1.000/0.989$, providing behavioral evidence alongside the ranking measurements (Table 15); these are completion-level accuracies, whereas AUC in that evaluation is prompt-level.

The Llama follow-up does not meet its installation rule: regexgroup reaches post-CoT AUC 0.500 on both seeds and runtime_prop reaches 0.781 (Appendix E.2). These negative results bound the Qwen finding. The protocol-specific estimators in Table 4 do not support an architecture ranking.

Trace content and causal dependence. All 2,560 parsed index/strindex completions across two seeds per family state a computed index (Appendix E.4). After replacing the stated index, we tested whether 320 continuations followed the edit: 24 adopted the implanted verdict, all in the strindex

Table 4: CoT starting readouts and trained endpoints (AUC). Paired Qwen entries are seeds $42/7$; Llama pairs keep their reported order, and single entries are single runs. Trained post-CoT AUCs use each prompt's mean parsed Yes/No prediction over eight sampled completions; direct AUCs use the answer-logit margin. Unadapted baselines use their source protocols. These are neither paired training-effect estimates nor architecture rankings. Full controls are in Appendix E.1.

Family	Unadapted post-CoT	Trained post-CoT	Trained direct
Qwen3-8B (trained: eight sampled completions per prompt)			
index	0.998	1.000/1.000	0.440/0.524
strindex	0.999	1.000/1.000	0.608/0.549
regexgroup	0.906	1.000/1.000	0.208/0.134
runtime_prop	0.953	0.944/0.955	0.426/0.372
det3mod	0.996	0.984	0.549
Llama-3.1-8B (trained: eight-sample mean per prompt)			
regexgroup	0.474	0.500/0.500	0.516/0.628
runtime_prop	0.736	0.781	0.629

seed-42 invalid-to-in-range cell; 295 returned to the original verdict and one did neither. This does not establish the stated scalar as a unique causal conduit. Verdict-flipping edits had a larger pooled margin effect than verdict-preserving edits, but the contrast had the required sign in only two of four family–seed conditions, below the pre-registered three-of-four gate. No semantic-mediation claim follows (Appendix E.5).

4.2 Dependence on initial state, model, and task

Dependence on the initial readout. A separate SFT-seeding study changes the answer readout before RL. All four index chains whose post-seed AUC was below approximately 0.60 declined to 0.443–0.501. All seven chains starting at or above 0.79 finished at or above 0.788, including three slight declines; the remaining chain moved $0.629 \rightarrow 0.639$. At the same $K = 50$ supervised optimizer steps, seeds showed erosion or retention, so step count alone does not characterize this finite sweep. Table 12 lists all twelve chains. This finite sweep concerns the starting readout, not a universal threshold; its protocol is separate from the exact-gradient comparison in §3.3.

Other models. The broader evidence separates descriptive readouts from paired learning tests. On six unadapted 3B–70B checkpoints from Qwen, Llama and Gemma, index probe accuracy is 66–78% while realized answers remain near chance (Table 18). In paired training on Qwen2.5-3B, Qwen3-4B, Llama-3.1-8B and Qwen3-14B, same-data SFT reaches index margin AUC 0.96–0.98, versus 0.49–0.72 for standard answer-only GRPO without forced coverage (Table 17). Second RL seeds on Qwen3-4B and Llama-3.1-8B remain below SFT. The original Qwen2.5-3B testbed gives SFT 0.963 against three RLVR seeds at 0.48–0.52 (Appendix B).

Forced coverage on Llama-3.1-8B tests the scope of the Qwen finding. Regexgroup and runtime_prop remain below SFT in both seeds, but SFT itself is seed-fragile on index and key. The Qwen positive-control contrast therefore cannot be formed under this native-chat protocol; Table 19 retains every result. The probe sites and evaluation protocols differ across these studies (Appendix A), so the descriptive breadth and paired training outcomes are interpreted separately.

Additional task tests. Neither GSM8K answer verification nor execution-labeled MBPP runtime-error prediction meets the pre-specified high-probe, chance-margin entry criterion: base answer-margin AUC is already 0.776 and 0.748, respectively. The multiclass and arithmetic checks also find no second clean instance under their stated criteria. A separate MBPP training check asked whether outcome reward strengthens an already expressed answer readout, but too few groups had reward variance: its pre-specified degeneracy rule prevents counting the observed margin increases as validation. These checks delimit the demonstrated settings, not the frequency or absence of such regimes on natural tasks (Appendices F.4, F.5.1, F.5.2, and F.7).

The probe–answer gap is associated with the supervised advantage on the 15 analyzed families, but the association weakens on 28 fresh families. Shared-baseline dependence may contribute to

the original relationship. It remains a conditional descriptive indicator, not an independent causal explanation or a validated predictor of new-task outcomes (Appendix H).

5 Related Work

Sampling support. Several studies find that tested RLVR recipes sharpen behavior already sampleable by the base model: pass@k coverage can narrow rather than expand (Yue et al., 2025), and training can contract empirical answer support (Wu et al., 2025). These accounts concern complete solution trajectories. Our binary task saturates support over final-answer actions, establishing that this narrower coverage is insufficient on index. Teacher-guided curricula that introduce patterns before RL consolidates them (Cai et al., 2026) address a different intervention.

Zero-variance groups and credit. Nie et al. (2026) diagnose binary-reward GRPO failures through zero-variance groups and study fixed-reference credit alongside coverage and group-size remedies. We examine what happens after every group has reward variance: the key control is learned while index remains weak. Other studies analyze cancellation along shared rollout directions (Cheng et al., 2026; Chang et al., 2026), positive-only training (Xu & Fang, 2026), and sign-preserving token-credit allocation (He et al., 2026). Here positive-only and uncentered credit serve as matched-sampling diagnostics; the main question is how the tested updates learn one answer readout.

Decodability and use. Representation-deficit accounts locate learning failures in missing or ill-formed features (Chen et al., 2026). Probes recover task-relevant information, but recovery does not establish causal use (Belinkov, 2022; Hewitt & Manning, 2019; Sudheendra & Srivastava, 2026). We use a probe to measure decodability and same-data SFT to test learnability. Where SFT also fails, as on det3mod, the cause remains unresolved.

Longer output channels. Process-aware rewards and token-level credit distribute feedback across longer outputs (Li et al., 2025; Uesato et al., 2022; Lightman et al., 2023). Our channel comparison instead asks whether an answer already expressed after reasoning remains accessible through that channel when direct answering is not learned. Appendix I develops the full comparison with probing, sampling-support and credit accounts.

6 Discussion and Conclusion

Guaranteed coverage supplies both final answers and restores reward variation, yet the tested centered update remains below SFT on index. The same intervention succeeds on dictionary lookup. This contrast separates answer availability from learning under the tested update.

The binary identities in Section 3.2 show that rewarding the correct answer and penalizing the wrong answer can share a margin-gradient direction. Their coefficients nevertheless weight prompts differently in shared parameters. The complete error-focused rule meets the joint installation criterion; because its weights are not normalized, it changes relative allocation and total update mass together. The exact-gradient primary comparison remains inconclusive about removing negative credit.

The reference index run improves AUC while greedy accuracy and mean gold probability stay at 0.5 (Figure 3). Reasoning exposes pre-existing answers. These measurements establish a controlled dissociation, not its prevalence or a unique mechanism.

Reproducibility Statement

Tasks, measurements, data separation, arm matching, and selected compute settings are specified in §2 and Appendices A, A.4, and A.6. The supplement provides the core deterministic generator, parser, supplied splits, and checkpoint summaries with code for checking endpoint classifications and paired contrasts. It does not provide a complete training rerun, trained weights, or the per-example records and software needed to reconstruct all reported AUCs and probabilities. Checks made from retained per-example evaluations, including the exact-gradient and allocation analyses, are documented separately from the publicly reproducible summary checks in the release scope. Reuse of the six allocation reference conditions is documented in Appendix D.2. The runtime-error programs are instantiated from deterministic templates and labeled by execution; the RL arms use the evaluated models' own completions as specified in the methods.

AI Use Statement

The research questions, methods, experimental design and scientific conclusions were developed by the authors. Generative AI tools assisted the organization and presentation of research materials, language editing and translation, figure preparation, and checks of experimental and analytical rigor, including related supporting code. These tools were used under author direction; scientific judgments and responsibility for the final manuscript remain with the authors.

Ethics Statement

The controlled runtime-error programs are generated from code templates, and the natural-task checks use the public GSM8K and MBPP benchmarks. This work uses no personal data and reports no human-subject research.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Yuntao Bai, Andy Jones, Kamal Ndousse, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1), 2022.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems*, 2023.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Pengxiang Cai, Tianchen Fang, Xiaohan Li, Qingyuan Zeng, Guocong Li, and Jintai Chen. Curriculum reinforcement learning can incentivize reasoning capacity in LLMs beyond the base model. *arXiv preprint arXiv:2606.22317*, 2026.

Powei Chang, Jinpeng Zhang, Chaoqun Sun, MiniWell Tsao, Lianrui Li, Jianxiang Xiang, Chenyu Wang, Yukang Gao, and Dongying Kong. SALT: When more rollouts don't help in group-based policy optimization and how to make them matter. *arXiv preprint arXiv:2606.05800*, 2026.

Yulin Chen, He He, and Chen Zhao. The unlearnability phenomenon in RLVR for language models. *arXiv preprint arXiv:2605.16787*, 2026.

Tianhao Cheng, Zeyu Huang, Zihan Qiu, Yu Cheng, Edoardo Ponti, Yinghui Xu, Ivan Titov, and Zenglin Xu. The cancellation hypothesis in critic-free rl: From outcome rewards to token credits. *arXiv preprint arXiv:2605.08666*, 2026.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics (TACL)*, 9, 2021.

Nelson Elhage, Neel Nanda, Catherine Olsson, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yuhang He, Haodong Wu, Siyi Liu, Hongyu Ge, Hange Zhou, Keyi Wu, Zhuo Zheng, Qihong Lin, Zixin Zhong, and Yongqi Zhang. Where hindsight credit can reside: A signed-capacity view of token updates in RLVR. *arXiv preprint arXiv:2604.11056*, 2026.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *NAACL*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *ICLR Workshop: Deep RL Meets Structured Prediction*, 2019.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yang Li, Zhichen Dong, Yuhan Sun, Weixun Wang, Shaopan Xiong, Yijia Luo, Jiashun Liu, Han Lu, Jiamang Wang, Wenbo Su, Bo Zheng, and Junchi Yan. Attention illuminates LLM reasoning: The preplan-and-anchor rhythm enables fine-grained policy optimization. *arXiv preprint arXiv:2510.13554*, 2025.

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. *arXiv preprint arXiv:2305.20050*, 2023.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *ICLR*, 2023.

Phuc Minh Nguyen, Chinh D. La, Duy M. H. Nguyen, Nitesh V. Chawla, Binh T. Nguyen, and Khoa D. Doan. The reasoning boundary paradox: How reinforcement learning constrains language models. *arXiv preprint arXiv:2510.02230*, 2025.

Wenhua Nie, Jianan Wu, Junlin Liu, Ziwei Li, Zheng Lin, Zijian Zhang, Yilong Fan, Haoran Zheng, and Jyh-Shing Roger Jang. Gradient starvation in binary-reward GRPO: Why group-mean centering fails and why the simplest fix works. *arXiv preprint arXiv:2605.07689*, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld et al. RL's razor: Why on-policy reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

Smitha Muthya Sudheendra and Jaideep Srivastava. When decodability is not enough: Logical validity representations, behavioral dissociation, and causal tests in language models, 2026. URL https://arxiv.org/abs/2609.02438v1.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *ACL*, 2019.

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 *NeurIPS*, 2023.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why RLVR may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

Mingwei Xu and Hao Fang. Beyond negative rollouts: Positive-only policy optimization with implicit negative gradients. *arXiv preprint arXiv:2605.06650*, 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 D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 57654–57689. Curran Associates, Inc., 2025. doi: 10.52202/085713-1933. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/537d5aa768c2d534016a4d06f87bc8fb-Paper-Conference.pdf.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A TASKS, MEASUREMENTS, AND TRAINING PROTOCOLS

This appendix specifies the common task and protocol definitions supporting §2; experiment-specific criteria remain with their results.

A.1 TASKS, LABELS, AND SAMPLES

The task is runtime-error prediction for short Python programs: given a snippet, answer Yes/No to "will running this raise a runtime error," with the ground-truth label obtained by actually executing the snippet. The core testbed uses six template families. Two are *computed* (labels depend on computed relations): *index* (list index in/out of range) and *zerodiv* (division by a value that may be zero). Four expose surface or partly surfaced cues: *type*, *value*, *attr*, and *key*. Additional experiments use *strindex* (string index in/out of range), *regexgroup* (full-match success before regex-group extraction), *runtime_prop* (whether a supplied function call raises an exception), *det3mod* (matrix-determinant congruence modulo a stated modulus), and *return_type* (whether a function call returns the proposed type). Generative answer-only arms generate at most 4 tokens and emit the short Yes/No answer without reasoning tokens. The split is generated with disjoint eval-first sampling (train 2780, eval 1170; computed-only train subset 1200 = index 800 + zerodiv 400).

Recorded task examples. The following examples are recorded rows 1 and 372 of the released core evaluation set, shown with normalized whitespace; they illustrate semantics, not a claim that all 8B comparisons share these evaluation items. Both ask: *Will running this Python code raise a runtime error? Answer Yes or No.*

index: recorded label No

```
vals = list(range(5))
a, b, c = 8, 2, 13
i = a + b - c
x = vals[i]
print(x)
```

key: recorded label Yes

```
x, y = 4, 6
m = {'yes': 1}
kk = 'yes' if x > y else 'miss'
v = m[kk]
print(v)
```

Here $i = -3$ is valid for a length-five list; the conditional lookup selects the absent key `miss`. The labels come from the released execution records.

A.2 CHECKPOINTS, INPUT FORMATS, AND PRECISION

The original answer-only recipe uses GRPO (group-relative policy optimization) in TRL with LoRA ($r = 16$; Hu et al., 2022) and eight samples per group. Controlled coverage and exact/allocation comparisons use experiment-specific update implementations.

The retained cross-family SFT/RLVR and margin-evaluation implementations load their base models in bfloat16. The locked exact-gradient and allocation protocols and the retained Qwen3-8B/Llama CoT training scripts use 4-bit NF4 with double quantization and bfloat16 computation. Software versions are recorded by experiment: the exact-gradient environment lock specifies PyTorch 2.5.1 (CUDA 12.4), Transformers 4.52.4, and PEFT 0.18.1; the Llama CoT launch record specifies TRL 0.24.0. The Qwen3-8B arms use `Qwen/Qwen3-8B`, not the Base checkpoint; they tokenize the task prompt as raw text without applying the chat template and therefore do not invoke an `enable_thinking` switch. The long-CoT arm instead appends its explicit reasoning instruction, while the generative answer-only protocols retain a four-token cap. The controlled two-action updates instead restrict the readout to the Yes/No logits. Direct answer margins use a single forward pass. Trained Qwen CoT evaluation samples $k = 8$ completions per prompt at temperature 1.0, top-$p = 0.95$ and a 512-token cap. Post-CoT AUC uses the prompt-level mean parsed Yes/No prediction; accuracy pools parsed completions (Appendix E.3).

Table 5: Measurement protocols by experiment and checkpoint.

Experiment / checkpoint	Input and measurement site	Fit/evaluation and statistic	Retained source scope
Core Qwen2.5-3B-Instruct	Raw task; decision-token residual at $\lfloor 0.7L \rfloor$	Deduplicated evaluation corpus; stratified five-fold probe; $C = 0.05$	Core evaluator excerpt and setup above
Qwen/Qwen3-8B coverage	Raw text, no chat; final hidden state at final prompt position	Reserved probe fit, distinct 400 update items, separate evaluation; $C = 0.5$	Coverage code excerpt and matching audit
Qwen3-8B probe controls	index; layer25	Residualization/erasure protocols and AUC in Appendix B.3	Protocol-specific report; no assumed common split/site
Llama-3.1-8B-Instruct	Native chat	Family/seed coverage comparison; CoT uses prompt means of eight samples	Separate reported protocols; no inferred common probe layer
Selected 3B–70B models	Base/Instruct variants as listed	Descriptive five-fold probes; realized answer accuracy	Table 18 and its stated corpus

Table 6: Training controls and comparison conditions.

Comparison	Changed object	Matching and scope
SFT / RLVR	Label-supervised / outcome loss	Model, data, adapter, evaluation; not equal feedback or efficiency
Forced coverage	Final-answer sampling	Same prompts/positions; unit-weight biased intervention
Loss / weighting	Loss or full weight rule	Matched prompts/sampler/budget; realized draws can differ
Local gradients	Incorrect-sample negative term	Base state and correct-sample weights; 4B/14B only
Exact updates	Sampler expectation / denominator / credit	Five arms, two families, three seeds; separate protocol
Allocation	w_α, $\alpha = -1, 0, +1$	Fixed process; six $\alpha = 0$ references reused
CoT	Instruction, token allowance and channel	Same outcome reward; not equal token compute

A.3 Probe and evaluation protocols by experiment

The probe is an ℓ_2-regularized logistic classifier (C=0.05) on the residual activation at the decision token, taken at layer $\lfloor 0.7\,L \rfloor$, scored by stratified 5-fold cross-validation over the deduplicated exact-program evaluation set, so identical programs cannot cross folds; the answer signal is the realized Yes/No accuracy and, for the sharper metric, the $\mathrm{logit}(\texttt{ Yes}) - \mathrm{logit}(\texttt{ No})$ margin AUC against the executed label, read at the final "Answer:" position.

The implementation supplement supplies source excerpts and hashes of the complete original files; it does not establish every historical runtime setting or global absence of leakage. Unspecified sites or split details are not borrowed from another experiment. Trained CoT AUC uses prompt-level means of sampled Yes/No predictions, whereas direct AUC uses answer-logit margins; realized CoT accuracy is completion-level (Appendices E.2, E.3).

A.4 ARM MATCHING AND FORCED COVERAGE

The SFT and RLVR arms are matched on base, data, adapter, split, evaluation, and effective update count; the remaining objective and per-arm optimizer differences are disclosed below, and we do not treat the two arms as equal supervision. The arm-matching comparison considers six dimensions: identical base-model weights, same LoRA adapter (r=16, α=32, dropout 0.05, all attention and MLP projections), identical disjoint model-training/evaluation splits, matched *effective update budget* (SFT uses $2{,}400$ labeled update items; RLVR uses $2{,}400$ sampled groups), same evaluation protocol (margin AUC at the answer position), and matched optimizer family. The three-way probe-fit / ground-truth-update / evaluation separation belongs specifically to the Qwen3-8B coverage-control audit: the probe is fitted on a reserved training subset, the update items come from a different training subset, and both are separate from evaluation. This differs from the core Qwen2.5-3B descriptive probe protocol above, which uses held-out folds within the evaluation corpus; it is not a three-way split of that corpus. The arms differ in the objective and in standard per-arm optimizer settings (e.g. SFT learning rate $1\mathrm{e}{-4}$ vs. RL learning rate $2\mathrm{e}{-5}$). The rate asymmetry cannot carry the result: at the 8B stage, the loss-swap and scaling-only arms run at the *same* learning rate, budget, forced-sampling rule and prompt schedule as the coverage arm ($2\mathrm{e}{-4}$ throughout), and reach higher mean AUC (0.782/0.823 vs. 0.558; 0.782 is the index-only RAFT mean, with heterogeneous seeds).

Two asymmetries are *by design*. (i) **Feedback quantity.** The RL arm receives *more* outcome evaluations than SFT receives labels—$19{,}200$ verifier calls (2400 groups $\times$ 8 generations) versus 2400 gold labels. Raw feedback count alone therefore cannot explain the asymmetry and is not equivalent to dense token supervision. (ii) **Skipped groups.** Under standard GRPO the locked family skips almost every group for zero reward variance. The forced-coverage arm removes this confound (skip rate $\to 0$) and shows that failure can persist on this forced-coverage rollout stream; it does not identify zero-variance skipping as the unique mechanism. Table 7 lists selected experiment settings.

Forced-coverage definition The forced-coverage arm intervenes at the sampling step only: within every GRPO group of G=8, both answer actions are guaranteed to appear—one Yes and one No are always included, and the remaining $G-2$ draws follow the current policy over the answer token—with the group's draws entering the group-relative update at unit weight. The prompt, the answer-token position, and the answer format are untouched; nothing is added to the context. The arm is therefore a forced-coverage intervention on the *action* distribution, not a prompt modification: the input distribution is untouched, while the action distribution is deliberately altered. Under forced coverage the number of Yes samples is $K = 1 + \mathrm{Binomial}(G-2, p)$ rather than $K \sim \mathrm{Binomial}(G, p)$, and the forced actions receive unit weight, so IID on-policy expectation identities do not apply to this arm.

A.5 PUBLIC RELEASE AND RETAINED VERIFICATION RECORDS

The public release and the retained records support different levels of verification. The protocol is described within this manuscript, in §2 and Appendix A: task-family semantics, execution-verified labels, disjoint split sizes, the probe protocol, and the margin-AUC metric. Loading precision and recorded software environments are distinguished by experiment in Appendix A. Appendix A.6 and Table 7 give selected training settings; Appendix A.4 gives the SFT-versus-RLVR arm-matching audit. Seeds are stated per experiment. The supplementary material includes the deterministic generator and template grammar, the answer parser and binary reward, the exact train/evaluation splits used in the reported analyses, and a deterministic computed-family subset. Its tests reproduce all three supplied data files exactly. It does not include complete configurations for every training condition, the scripts needed to rerun the principal training and probe analyses, evaluation data and software for the remaining conditions, model weights, or trained checkpoints. The release therefore supports interpretation and independent verification of endpoint classifications and paired contrasts from the released checkpoint summaries, not a full training rerun. The authors report that every number was independently checked or recomputed from retained evaluation data. For all 30 exact-gradient conditions and all 18 allocation follow-up conditions, retained per-example evaluations cover steps 0, 100, 200, and 300; six follow-up conditions reuse the α=0 runs after exact identity was verified. Two scripts requiring only the Python standard library recompute the exact-gradient conclusion (inconclusive under the pre-specified rule) and the follow-up conclusion (local sufficiency only). The reported independent checks used retained per-example evaluation data

rather than rerunning training. The supplementary material provides public checkpoint summaries and code that recomputes endpoint classifications and paired contrasts from those summaries; it does not include the per-example records or the code needed to reconstruct their AUCs and probabilities. The included tests pass in a standard Python environment.

Evidence surface	Capability and boundary
Public data and summaries	Deterministic generator, parser/reward, supplied splits, public checkpoint summaries, endpoint classification and paired-contrast code.
Retained per-example evaluations	Historical independent checks of AUC, probability and record order at steps0/100/200/300; these are not included in the public summary release.
Not supplied for full rerun	Complete per-arm configurations, principal training/probe entrypoints, all remaining-arm data/harnesses, weights and trained checkpoints.

The archived evidence index records provenance. The public release supports data/reward reconstruction and endpoint-summary checking, not a complete training rerun.

A.6 Selected compute and settings

Seeds include 42, 7, and 123; the exact seeds for each experiment are stated in its table or paragraph. The table records settings reported in this manuscript; it does not provide every setting needed to reproduce the listed experiments.

Table 7: Selected per-experiment compute and hyperparameters reported in this manuscript.

Experiment	Models	Key settings
Probe and RLVR on the core testbed	Qwen2.5-3B	GRPO LoRA r=16, α=32; 8 generations per group; learning rate 2e$-$5; 600 steps
SFT install arm	(per family)	LoRA answer-token CE; 3 epochs; learning rate 1e$-$4; batch 8
Forced-coverage control	Qwen3-8B, Llama-3.1-8B	both answers in every group; mean absolute gold-logit derivative on Qwen3-8B: 0.010→0.33
Cross-scale probe	Qwen3-4B/14B, Llama-3.1-8B/70B, Gemma-2-27B	probe: 5-fold CV logistic C=0.05 at layer $\lfloor 0.7L \rfloor$
Exact-gradient and allocation follow-up	Qwen3-8B	2 families $\times$ 3 seeds $\times$ 5 arms, 300 steps/cell; triggered 3-arm follow-up reuses six $\alpha = 0$ and adds twelve $\alpha = \pm 1$ cells

B Baseline Readouts and Probe Controls

These measurements support the descriptive and installability distinctions in §2.2; protocols are not assumed identical across models.

B.1 The six-family Qwen2.5-3B comparison

Table 8 contrasts answer-readout patterns in computed and surface-cued families across the six-family baseline.

Appendix Table 8 makes the dissociation concrete. On the computed families, the internal signal is linearly decodable: probes decode *index* at 70% and *zerodiv* at 66%. But the short-answer behavior does not express that information. Base answer-margin AUC is only 0.474 and 0.537, and after computed-only RLVR it remains flat at $0.478/0.486/0.524$ and $0.522/0.421/0.558$ across seeds. By contrast, the surface or partly surfaced families already leak usable answer-channel structure: *type* is perfectly ranked but threshold-stuck, *value* is nearly perfectly inverted rather than absent, and *attr/key* carry partial signal that RLVR can recalibrate. This asymmetry co-occurs with almost total reward-variance collapse in this regime: in $119/120$ logged steps every GRPO group had zero reward variance, with mean $\mathrm{frac}_{\sigma=0} = 0.9992$. These runs distinguish a descriptive internal

Table 8: Probe decodability versus answer-channel behavior on the Qwen2.5-3B testbed.

Family	Probe acc.	Base margin AUC	Reported answer-channel outcome
type (surface)	100%	1.000	binary 50%→100% after 100 training steps; threshold repair only
value (surface)	100%	0.017	AUC 0.970 after 100 training steps; binary reaches 83% and later returns to 50% under threshold drift
attr (held-out)	92%	0.113	AUC 0.504/0.459/0.749 after 100/300/600 training steps
key	98%	0.282	AUC 0.521/0.468/0.662 after 100/300/600 training steps
index (computed)	70%	0.474	computed-only RLVR stays flat at 0.478/0.486/0.524 across 3 seeds
zerodiv (computed)	66%	0.537	computed-only RLVR stays flat at 0.522/0.421/0.558; binary sits at the 59% all-No baseline

readout from the behavioral readout seen by the sampled update. Section 2.2 then tests forced answer coverage; removing zero reward variance does not by itself install index.

B.2 Same-data supervision and the missing-positive caveat

This comparison tests supervised installation on the original 3B computed-family data; the zerodiv training-coverage caveat remains part of its interpretation.

Table 9: On-policy RLVR versus SFT on identical computed-family data (Qwen2.5-3B testbed). Values are margin AUC; binary accuracy is given in parentheses.

Family	RLVR seed 1	RLVR seed 2	RLVR seed 3	SFT
index	0.48	0.49	0.52	0.963 (binary 86%)
zerodiv	0.52	0.42	0.56	0.485

The SFT comparison sharpens the interpretation (Appendix Table 9). On *index*, SFT reaches margin AUC 0.963 on disjoint evaluation while the three RLVR seeds remain at $0.48/0.49/0.52$: direct cross-entropy installs the missing answer readout on the same family where on-policy RLVR does not. This is installability within a family, not transfer across families. The *zerodiv* endpoint requires a training-coverage caveat: on Qwen, SFT stays at 0.485. The retained SFT-arm loaders use a training split containing 400 No examples and no Yes examples for this family, so this endpoint does not distinguish SFT resistance from missing positive training coverage.

B.3 Format, operand, and erasure controls

Two pre-registered controls on the *index* family (Qwen3-8B, probe layer 25) separate a decodable feature from a surface correlate the probe might exploit.

(i) Format-residualized probe: decodability persists after removing the specified surface covariates. Regressing the pre-declared surface covariates (prompt length, code length, list-size literal) out of the hidden states leaves the refit probe at 0.853 versus 0.885 unresidualized—the pre-registered criterion (at least 0.60) was met. An aggressive variant additionally residualizes the computation's own integer operands: the probe then collapses (0.524), while a linear classifier on those operands alone reaches AUC 0.935. The residualized quantities are task *inputs* rather than merely format covariates; we report this variant for completeness rather than use it as a decision criterion. Consequently, this probe control cannot distinguish a computation already performed by the base answer-only policy from linearly carried operand information that the probe itself combines; *computed* describes the task construction, not a causal claim about that policy.

(ii) Amnesic erasure: erasing the probe direction does not prevent supervised installation. A LEACE-style projection (Belrose et al., 2023) removes the probe direction at the probed layer (the base probe collapses to 0.513, confirming the eraser works), yet SFT still installs the readout *under* the eraser: margin AUC $0.939/0.902$ across two seeds (mean 0.921, versus 0.928 without erasure; the pre-registered success criterion for a causal block was ≤ 0.60, and it was not met). Specificity controls behave (random-direction erasure 0.952; erasing the *key* direction leaves index install at $0.954/0.945$; the unlocked *return_type* installs at 1.000 under the index eraser). The observed SFT installation under erasure is consistent with redundant encoding or re-derivation during fine-tuning; this control does not distinguish those possibilities. The paper therefore makes *no* causal-necessity claim about the probe direction anywhere: the probe certifies linear decodability, and the causal weight is carried by the training controls.

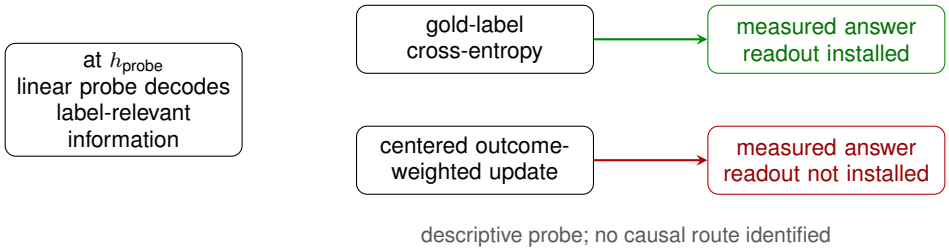


Figure 4: **Decodability and installation are separate measurements.** The probe box denotes descriptive recovery of label-relevant information; the training branches summarize the reported index comparison under the tested answer-only updates. The diagram does not identify a probe site or direction as the causal route used by training. Probe sites and protocols are experiment-specific.

B.4 Checkpoint intervention: inconclusive

In a separate pre-registered inference-only analysis on Qwen2.5-3B, we tested whether replacing the risky-token residual projection along a pre-specified label probe changed endpoint behavior at the base, GRPO-trained, and SFT-trained model states; GRPO on *type* was the positive control. Here the risky token contains the final character of `x=vals[i]` for *index* or `c=a+b` for *type*. The pre-registered matrix completed, and independent recomputation from retained evaluation data confirmed record completeness and the numerical results. The decision rule was nevertheless inconclusive: endpoint and representation criteria were met, while the direction, intervention-control, positive-control, and downstream-isolation criteria were not. In particular, the final-block negative control exceeded its pre-specified tolerance, so the analysis does not isolate the proposed causal route. We draw no mechanism inference from it. It adds no support beyond the evidence already reported and changes neither the inconclusive exact-gradient result nor the allocation-follow-up local-sufficiency result or any main-text claim.

C Coverage, Loss Swaps, and Initialization Controls

These controls expand §2.2, §3, and §4.2; each retains its own rollout, seed and comparison scope. For an overview of the separate answer-only comparisons, see Figure 1.

C.1 Coverage seeds and manipulation checks

The recorded manipulation checks were as follows: in the Qwen2.5-3B testbed, every group had zero reward variance in $119/120$ logged steps and the mean zero-variance fraction was 0.9992. For the Qwen3-8B comparator, the mixed-reward group rate rises from 0.024 (standard GRPO on *index*; 0.001 on *key*) to 1.0 under coverage; the corresponding standard-GRPO zero-variance fraction on *index* is approximately 0.98. The mean absolute derivative at the correct-answer logit rises from 0.010 to 0.33 (≈ 0.33 on every family under coverage).

Table 10: **Qwen3-8B answer-only training with two seeds.** Entries report two-seed mean margin AUC. Index is the supervised-learnable coverage-gap case; key is the coverage-positive control; det3mod remains weak under the tested answer-only SFT recipe. The inverted mf_heap_pop_empty control has base AUC 0.092 and recalibrates under coverage. These roles are descriptive, not classifications under the separate three-seed exact-gradient rule. Seedwise coverage/SFT values are in Table 11.

Family	Standard GRPO	Forced coverage	SFT
index	0.535	0.558	0.927
key	0.334	0.994	1.000
det3mod	0.599	0.459	0.537
mf_heap_pop_empty	0.572	1.000	1.000

Table 11: Seedwise margin AUC for the original two-seed Qwen3-8B coverage/SFT comparison. Only the index/key seed values already reported in the original caption are tabulated; other seed cells are not inferred. Means and det3mod are in Table 10.

Family	Arm	Seed42	Seed7
index	Coverage	0.635	0.481
index	SFT	0.906	0.949
key	Coverage	0.989	1.000
key	SFT	1.000	1.000

For the *index* comparison, this statistic is the absolute scalar derivative of the group's rollout-averaged loss with respect to the gold-answer logit, computed before minibatch reduction. It is averaged over all $2{,}400$ visited groups per seed (skipped groups contribute zero), then equally across the two seeds; it is not a parameter-gradient norm or an optimizer-step magnitude.

Margin-AUC trajectories were recorded at training checkpoints but are not included for these runs in this anonymous supplement. KL and entropy trajectories were not logged in these runs.

The importance-sampling variant (behavior policy mixed toward uniform, importance-weighted updates) is the manipulation check: it does not raise that answer-logit statistic (0.0008 on *index*) and yields a two-seed mean margin AUC of 0.482 (observed range 0.460–0.504).

The inverted mf_heap_pop_empty control has base AUC 0.092 and reaches 1.000 under coverage, 0.994 under importance sampling, and 1.000 under RAFT in the same Qwen3-8B runs. A mis-signed readout alone is insufficient to explain the index gap.

C.2 Correct-completion cross-entropy

The coverage arm leaves one alternative standing: perhaps learning from *self-generated* tokens, rather than the objective, is what fails. RAFT (Dong et al., 2023) applies plain cross-entropy to the correct completions from a matched forced-coverage sampler, prompt order, random-number construction and budget. The six policy-drawn answers are sampled anew from each arm's current policy; the realized responses are not frozen across training arms. On the post-hoc three-family diagnostic set, *index*, *strindex*, and *regexgroup* have coverage AUC 0.56–0.65 and SFT 0.72–0.93. *Regexgroup*'s 0.651 coverage AUC slightly exceeds the 0.65 cutoff.

RAFT reaches $0.782/0.746/0.765$ (mean 0.764; per-seed values in Table 1). The pre-registered four-family criterion included the *key* control fixed by coverage and required mean ≥ 0.8; that mean was 0.849. We report 0.764 because excluding key is the more conservative comparison. The borderline *runtime_prop* (0.740 under coverage) is also excluded. The unlocked *return_type* installs under RAFT (0.997), while SFT-resistant *det3mod* does not (0.600; SFT 0.537). The loss swap improves the family means above but does not overcome the weak det3mod endpoint. Because correct completions equal gold labels in this binary task, RAFT is a localization control, not an independent method or proposed objective.

The strindex family was screened after pre-registration under pre-declared criteria (base margin 0.471, probe AUC 0.852). Adding it to the pre-registered four-family set gives the five-family mean 0.829. Table 1 retains every family and reported seed, including the low endpoints.

C.3 RLOO UNDER FORCED COVERAGE

To test whether the lock belongs to GRPO's particular normalization or to the centered outcome-credit class, we pre-registered an RLOO arm (Ahmadian et al., 2024) under the same forced-coverage design (identical budget: $2{,}400$ groups, $19{,}200$ verifier calls per seed; criterion fixed before any run: family mean margin AUC ≤ 0.65 = still locked, > 0.72 = on the installed side). RLOO's leave-one-out advantage satisfies $r_i - \mathrm{mean}_{j\neq i} r_j = \frac{G}{G-1}(r_i - \bar{r})$, an exact positive rescaling of group centering; Dr.GRPO's advantage (Liu et al., 2025) (dropping the σ and length normalizations) is algebraically identical to our centering-only arm for single-token completions. Results: on *index* the arm stays locked ($0.695/0.528$, mean 0.611), completing the class picture there (GRPO 0.535, coverage 0.558, centering-only 0.638, RLOO 0.611; scaling-only 0.823 exceeds this AUC threshold). On *strindex*—whose centering-only outcomes already differed between seeds ($0.585/0.914$)—the $8/7$ rescaling exceeds the AUC threshold on both seeds ($0.801/0.855$, mean 0.828). We report the class statement family-graded: robustly locked families stay locked across centered variants; marginally locked families tip under magnitude rescaling and seed variance. Across these arms, centered updates leave mean index AUC at or below 0.638, whereas the scaling-only update reaches 0.823. The latter changes correct-sample weights as well as removing the negative-advantage term. These classifications use the AUC-only rule stated above, not the later joint rule of Section 3.3. No retuning was performed.

C.4 SENSITIVITY TO THE INITIAL ANSWER READOUT

Changing the initial readout. In a separate SFT-seeding protocol on index, all four chains with post-seed AUC below approximately 0.60 decline to 0.443–0.501 after RL. The seven chains starting at or above 0.79 include three slight declines (largest drop $0.933 \to 0.911$), and all finish at or above 0.788. The remaining chain, $0.629 \to 0.639$, is preserved. Across the three seeds, the same $K{=}50$ supervised optimizer-step count yields both erosion and retention. This threshold-like, signal-level pattern in the finite sweep is not captured by step count alone and does not establish a universal threshold law (Table 12).

Table 12: SFT-seeded RL dose response on the locked *index* family (post-seed → post-RL margin AUC). Seeds 7 and 123 replicate the sweep at $K \in \{20, 50, 200\}$; K is the number of supervised optimizer steps (batch size 8) in the seeding phase. Across all twelve chains the outcome separates on the *post-seed signal level*: every chain below ≈ 0.60 declines in AUC ($4/4$); the seven chains starting at or above 0.79 include three slight declines (largest drop $0.933 \to 0.911$), and all finish at or above 0.788. The remaining chain, $0.629 \to 0.639$, is preserved. Not run indicates seed/K combinations outside the replicated sweep.

K	Seed 42	Seed 7	Seed 123
20	0.547→0.501	0.530→0.443	0.544→0.485
50	0.629→0.639	0.561→0.495	0.797→0.788
70	0.849→0.844	not run	not run
100	0.861→0.863	not run	not run
200	0.947→0.970	0.933→0.911	0.938→0.957
400	0.969→0.973	not run	not run

The twelve-chain SFT-seeded RL result is summarized in §4.2 and reported seedwise in Table 12; here it serves only as within-testbed localization, not a universal law about what RLVR can learn.

D EXACT-GRADIENT AND ALLOCATION EXPERIMENTS

D.1 EXACT UPDATES, DENOMINATORS, AND REFERENCE CRITERIA

We pre-registered a fixed $2 \times 3 \times 5$ Qwen3-8B matrix to separate the sampler expectation, the skip-renormalized joint-batch denominator, and confidence-dependent prompt weighting. The two families were *index* (primary) and *key* (manipulation control), with seeds $42, 7, 123$ and 300 steps per condition. The five updates were: the exact IID centered update with the true joint-batch denominator; the exact forced-sampler update; the sampled forced-coverage update; the coefficient-matched correct-only update, which used the same detached joint-batch coefficient as the exact IID centered update; and the fixed-denominator exact centered update with a fixed batch-size denominator B. All 30 conditions completed without retirement.

Implemented losses and denominator. For a batch of $B = 8$ prompts and group size $G = 8$, let p_i be the live two-answer probability of the correct answer and Δ_i its correct-minus-incorrect log-probability margin. Write sg for stop-gradient. For IID draws define $K_i \sim \mathrm{Binomial}(G, p_i)$ and

$$Y(k) = \frac{k(G-k)}{G[\sqrt{k(G-k)} + G\varepsilon]}, \quad Y(0) = Y(G) = 0,$$
$$m_i = \mathbb{E}[Y(K_i)], \qquad a_i = 1 - p_i^G - (1-p_i)^G,$$
$$h_i = \int_0^1 \prod_{j \neq i} (1 - a_j + a_j t)\, dt, \qquad \lambda_i = m_i h_i.$$

Here $h_i = \mathbb{E}[(1+M_{-i})^{-1}]$ for the number of other nonzero-variance groups M_{-i}. Thus λ_i includes the random retained-group denominator; it is not $m_i/\mathbb{E}[M]$. The implemented surrogate losses are

$$L_{\mathrm{IID}} = -\sum_i \mathrm{sg}(\lambda_i)\Delta_i,$$
$$L_{\mathrm{forced}} = -B^{-1} \sum_i \mathrm{sg}(e_F(p_i))\Delta_i,$$
$$L_{\mathrm{sampled}} = -(GB)^{-1} \sum_{i,j} \mathrm{sg}(A_{ij}) \log \pi_i(a_{ij}),$$
$$L_{\mathrm{correct}} = -\sum_i \mathrm{sg}(\lambda_i) \log \pi_i(\mathrm{correct}),$$
$$L_{\mathrm{fixed}} = -\sum_i \mathrm{sg}(m_i/B)\Delta_i.$$

For forced sampling, $K_i^F = 1 + \mathrm{Binomial}(G-2, p_i)$ and $e_F(p_i) = \mathbb{E}[Y(K_i^F)]$. In the sampled arm $A_{ij} = (r_{ij} - \bar{r}_i)/(\sigma_i + \varepsilon)$, with $\varepsilon = 10^{-6}$. The correct-only arm matches the positive log-probability coefficient, not the margin coefficient after differentiation. Coefficients and live log-probabilities come from the same training-mode forward pass.

These five arms use Qwen3-8B in 4-bit NF4 with double quantization and bf16 computation, LoRA rank 16, scale 32 and dropout 0.05. AdamW uses learning rate 2×10^{-4}, betas $(0.9, 0.999)$, optimizer epsilon 10^{-8} and weight decay 0.01. There is no learning-rate scheduler, gradient clipping, KL/reference-policy penalty or PPO ratio/clip term. Each arm runs 300 outer steps with seeds 42, 7 and 123. IID/correct-only and fixed-denominator arms skip an optimizer step when every coefficient is zero; forced arms update. These optimizer paths are not claimed to equal the expectation of a stochastic AdamW trajectory.

Each family uses 400 update prompts and 200 disjoint evaluation prompts, with 100 Yes and 100 No evaluation labels. Behavioral accuracy takes the argmax of the two answer logits, choosing Yes on exact ties; it is not unrestricted greedy generation.

The endpoint rule was specified before any outcome inspection. For each seed, a behavioral install required both a Clopper–Pearson 95% accuracy lower bound strictly above 0.5 and mean two-answer expected reward strictly above 0.5. Across three seeds, we classified a condition as installed when at least two seeds met the behavioral criterion and mean margin AUC was at least 0.80; as not installed

when at most one seed met that criterion and mean AUC was at most 0.65; and as inconclusive otherwise. An independent script using only the Python standard library recomputed all metrics at steps 0, 100, 200, and 300 from retained per-example evaluation records and matched the reported summaries for all 30 conditions.

All five arms and both families are reported in main Table 2.

The pre-registered reference criteria were met: the sampled forced-coverage update did not install *index* and installed *key*, while the matched correct-only update installed *key*. The primary exact-centered versus matched-correct-only comparison on *index* was nevertheless inconclusive: the exact centered update did not install *index* ($0/3$, mean AUC 0.489), whereas the correct-only comparator was inconclusive ($2/3$, mean AUC 0.756). This result neither establishes nor rejects a category-changing role for the confidence-dependent sign-asymmetric weighting and does not identify a unique training mechanism. No seed, threshold, update, or claim was changed after inspection; the pre-specified protocol therefore called for the minimal follow-up below.

D.2 Allocation definition and reused references

Here p is the policy probability of the execution-correct answer after a two-answer softmax over the Yes and No token logits; unlike p in Appendix A.4's Yes-count formula, it is label-relative. For each prompt, the implementation applies the detached coefficient to the live margin $\Delta_i = \log \pi(\text{correct}) - \log \pi(\text{wrong})$ through $-\frac{1}{B}\sum_i \mathrm{stopgrad}[w_\alpha(p_i)]\Delta_i$, with no arm-wise coefficient renormalization; $B = 8$. No arm-wise mass/L1/L2 normalization, learning-rate compensation, additional arm-specific clipping, or scale-control arm is introduced. Before inspecting outcomes, we specified Qwen3-8B, families *index* and *key*, seeds $42, 7, 123$, the same initialization/optimizer/examples/data order/arm-invariant RNG as the exact forced-sampler condition, and

$$w_\alpha(p) = e_F(p)\big[1 + \alpha(1 - 2p)\big], \qquad \alpha \in \{-1, 0, +1\}.$$

For the fixed group size $G = 8$, the implementation uses the closed form

$$e_F(p) = \sum_{k=1}^{7} \binom{6}{k-1} p^{k-1}(1-p)^{7-k}\, c_k,$$

$$c_k = \frac{k(8-k)}{8\big[\sqrt{k(8-k)} + 8 \times 10^{-6}\big]}.$$

Thus $\alpha = -1$ only reverses the allocation ($2e_F(p)p$), $\alpha = 0$ is exactly the forced-sampler allocation, and $\alpha = +1$ is the binary-SFT-signed allocation ($2e_F(p)(1-p)$). No-training checks verified pair-sum and affine identities, non-negativity, arm-invariant RNG, and exact $\alpha = 0$ coefficient, loss, delta, and parameter-gradient parity (maximum error 0). Because the six $\alpha = 0$ conditions exactly matched the previously completed forced-sampler runs, we reused those verified-identical runs and trained only the twelve $\alpha = \pm 1$ conditions for 300 steps.

D.3 Seed endpoints and the local and strict rules

Table 13: Allocation follow-up endpoints at step 300: greedy accuracy / mean gold probability / margin AUC for each seed. Aggregate classifications are reported in main Table 3.

Family	α/Allocation	Seed 42	Seed 7	Seed 123
index	-1/reverse	0.500/0.500/0.469	0.500/0.500/0.432	0.500/0.500/0.524
index	0/even	0.500/0.500/0.801	0.500/0.500/0.658	0.500/0.500/0.645
index	$+1$/SFT-signed	0.865/0.849/0.959	0.830/0.784/0.924	0.695/0.677/0.900
key	-1/reverse	0.500/0.500/0.581	0.500/0.500/0.495	0.500/0.500/0.554
key	0/even	0.975/0.975/0.989	1.000/1.000/1.000	1.000/1.000/1.000
key	$+1$/SFT-signed	1.000/1.000/1.000	1.000/1.000/1.000	1.000/1.000/1.000

All twelve new conditions completed. Independent recomputation from retained per-example evaluation records reproduced probabilities within 3.65×10^{-8} and matched the reported summaries exactly. The pre-specified local-sufficiency criterion was met: on *index*, the SFT-signed allocation

installed in all three seeds while the reverse allocation did not install the readout, and the *key* control confirmed that the setup could still install a readout. Relative to $\alpha = 0$, both final mean gold probability and AUC increased under $\alpha = +1$ in every index seed. The separately pre-specified strict-trajectory criterion was not met because $\alpha = 0$ and -1 had the same final mean gold-answer probability of 0.5, even though AUC was higher for $\alpha = 0$ in all three seeds. The warranted claim is therefore local sufficiency of the SFT-signed allocation under fixed forced coverage and training process, not strict allocation ordering or unique universal mechanism identification.

Figure 3 displays the retained checkpoints for these same conditions.

For each of the two continuous step-300 endpoints (mean gold-answer probability and margin AUC), a paired contrast is *consistent positive* if at least two of three matched seeds and the mean difference are strictly positive; *nonnegative* uses at least two nonnegative seeds and a nonnegative mean. There is no epsilon, tie tolerance or seed exclusion. Local sufficiency requires key installation at $\alpha = 0, +1$, index installation at $+1$ and non-installation at -1, and, on index, consistent-positive $+1 - 0$ and nonnegative $0 - (-1)$ for both endpoints. All these conditions are met. Reverse key is reported but need not install. The strict criterion additionally requires both adjacent contrasts to be consistent positive for both final endpoints; it is unmet because even and reverse have equal final gold probability. It is not a rule of monotonicity at every training step and cannot upgrade the local verdict.

D.4 Endpoint-rule sensitivity and record scope

Using only the retained Qwen3-8B step-300 per-example evaluation records and CPU arithmetic—without rerunning training or inference—we varied the rule's installation AUC threshold over 0.85–0.97 and its chance width over 0.10–0.225. The exact-gradient comparison remained inconclusive at every swept installation threshold. The allocation follow-up met the local-sufficiency criterion through 0.91 and was inconclusive from 0.93; its mean *index* endpoint AUC of 0.9278 is an observed boundary, not a newly registered threshold. At chance width 0.10, sampled-coverage index no longer met the "not installed" classification, so the exact reference criterion was not met although the primary exact-gradient comparison remained inconclusive. This post-hoc check does not tune, change, or re-register any pre-specified threshold.

E Reasoning-Channel Tests

E.1 Qwen starting readouts and trained endpoints

Holding the outcome reward fixed while opening an answer channel tests whether the readout becomes behaviorally available under the tested recipe; this is a channel-access test, not identification of a unique credit-assignment mechanism. We train long-CoT GRPO on Qwen3-8B with the same execution-verified reward and up to 512 reasoning tokens allowed before the final answer.

The readout becomes behaviorally available through the reasoning channel on all four counted families (Table 14): the post-hoc three-family diagnostic set reaches post-CoT prediction AUC 1.000 (two seeds each). Within the set, *regexgroup*'s coverage AUC is 0.651, just above the 0.65 cutoff, and the borderline *runtime_prop* (already partly installable under coverage, 0.740) reaches $0.944/0.955$ across two seeds; realized accuracy tracks the prediction AUC (0.86–1.00 sampled correct, against the 50% constant-class baseline when locked; Table 15, appendix); evaluated answer-only, the same trained models stay far lower (direct margin AUC 0.13–0.61). The controls behave (Table 14): *key* and the unlocked *return_type* install as expected (not counted). Most importantly, *det3mod* is behaviorally available through CoT at 0.984 while the same trained model remains at 0.549 answer-only and answer-only SFT reaches 0.537; because the untrained post-CoT baseline is already 0.996, this contrast supports channel-specific access rather than RL-induced installation (§6). The traces state the decisive computation—every parsed completion states the computed index before answering ($2560/2560$), and removing the reasoning channel collapses the readout (index, seed 42: direct 0.44 vs. post-CoT 1.00; Appendix E.4)—but the splice test does not establish the stated scalar as a unique causal conduit. After a preregistered contradictory splice, $24/320$ continuations adopted the implanted verdict and $295/320$ returned to the original one; $293/295$ of those recoveries contain prespecified recalculation or range-check text, consistent with the open trace serving as a redundant repair/recomputation workspace rather than a faithful scalar conduit (Appendix E.5).

Table 14: Long-CoT GRPO channel opening on Qwen3-8B: post-CoT prediction AUC versus the same trained model evaluated answer-only ("direct"), per family (seeds $42/7$ where two values are shown). The post-hoc three-family diagnostic set and the borderline *runtime_prop* are all behaviorally available through the reasoning channel while the answer-only readout stays low. Within that diagnostic set, *regexgroup*'s coverage AUC is 0.651, just above the 0.65 cutoff; this is access to signal already carried by the base Qwen reasoning policy, not evidence that RL created the computation. Controls (below the rule) are not counted in the tally: *key* is fixed by coverage; *return_type* is unlocked (direct already high); *det3mod* is SFT-resistant, where even answer-only SFT fails (0.537), yet it is behaviorally available through the reasoning channel. The base post-CoT AUCs are $0.906/0.953/0.998/0.999/0.996$ for *regexgroup*/*runtime_prop*/*index*/*strindex*/*det3mod*. Trained post-CoT AUCs and Table 15 use the same eight-sample evaluations; direct AUC uses a forward pass on the answer-only prompt.

Family	Status	Post-CoT AUC	Direct AUC
index	post-hoc diagnostic set	1.000 / 1.000	0.440 / 0.524
strindex	post-hoc diagnostic set	1.000 / 1.000	0.608 / 0.549
regexgroup	boundary case (coverage 0.651)	1.000 / 1.000	0.208 / 0.134
runtime_prop	borderline	0.944 / 0.955	0.426 / 0.372
key	positive control (fixed by coverage)	1.000 / 1.000	0.251 / 0.211
det3mod	SFT-resistant (answer-only SFT fails)	0.984	0.549
return_type	unlocked control	1.000	0.776

E.2 Cross-architecture CoT channel opening (negative)

We repeated the long-CoT GRPO channel opening on Llama-3.1-8B-Instruct under the matched reward, split and optimizer budget, using the Llama native chat template rather than Qwen's raw-text format, pre-registering the same install bar (post-CoT AUC over per-prompt mean sampled predictions ≥ 0.93, with answer-only direct AUC < 0.65) before any trained family–seed condition was observed. We ran the two families that pass Llama's per-seed coverage criterion: *regexgroup* (both seeds) and *runtime_prop* (one seed). The reported pre-task-training baselines, with their estimator distinction stated below, already show different starting readouts in these settings: the Qwen3-8B reasoning channel expresses the signal on these families (reported base post-CoT AUC *regexgroup* 0.906, *runtime_prop* 0.953, *index* 0.998, *strindex* 0.999, *det3mod* 0.996, answer-only direct AUC 0.26–0.54; $80/80$ parsed), whereas the untrained Llama channel has post-CoT AUC 0.474 and direct AUC 0.365 on *regexgroup*. These separately reported baseline values do not establish a paired training effect or a matched comparison of task difficulty. After training, no family–seed condition installs. *Regexgroup* has post-CoT AUC $0.500/0.500$ and direct AUC $0.516/0.628$ on the two seeds, and *runtime_prop* has a reported unadapted value of 0.736 and a trained endpoint of 0.781 (direct 0.629), all below the 0.93 bar (each family–seed condition $80/80$ parsed, k=8). Per the preregistration we report this as a negative without retuning learning rate, steps, or template; supervised installation is markedly seed-fragile on Llama (Table 19), so a null consistent with seed fragility here is consistent with the coverage arm. The Qwen channel-opening claim is scoped accordingly, and no Qwen result changes. Read constructively, the contrast makes the bypass's measured starting condition visible: higher base post-CoT AUC.

For trained Qwen, each family–seed evaluation uses 80 prompts and eight sampled completions per prompt (temperature 1, top-$p = 0.95$, 512-token cap). The parser takes the final `Answer: Yes/No`; unparsed completions are omitted. With k_i parsed predictions $a_{ij} \in \{0, 1\}$, the post-CoT score is $k_i^{-1} \sum_j a_{ij}$, and AUC compares these prompt scores with execution labels, assigning ties half credit. It is not a CoT-end logit margin. Direct AUC uses the Yes-minus-No logit on the original answer-only prompt. All 80 prompts have parsed answers. Runtime_prop seed 7 and return_type each parse 639 of 640 completions; the remaining reported checkpoints parse 640. Completion accuracy uses these parsed denominators. The two Qwen tables therefore report complementary metrics from the same evaluation, not independent greedy and sampled AUCs. Unadapted baselines retain their source-protocol estimates and are not paired training-effect estimates.

E.3 Prediction AUC and realized behavior move together

Post-CoT prediction AUC ranks the prompt-level parsed-answer scores; to confirm that the CoT access result reflects behavior rather than a ranking artifact, we pair it with realized accuracy—the fraction of *sampled* completions that are correct against the executed label—from the same trained checkpoints (k=8, temperature 1.0).

Realized accuracy tracks post-CoT prediction AUC: it is 0.86–1.00 where the readout is behaviorally available, versus the 50% constant-class baseline in the locked answer-only regime. The access claim therefore concerns what the model *answers*, not only how it ranks.

Table 15: Post-CoT prediction AUC and sampled accuracy from the same eight-sample evaluations (training seeds 42/7 where paired). AUC uses each prompt's mean parsed Yes/No prediction; accuracy pools correct parsed completions. These are complementary metrics from the same responses.

Family	Post-CoT prediction AUC	Sampled accuracy
index	1.000 / 1.000	1.000 / 0.989
strindex	1.000 / 1.000	0.998 / 0.998
regexgroup	1.000 / 1.000	1.000 / 1.000
runtime_prop	0.944 / 0.955	0.864 / 0.858
det3mod	0.984	0.928

E.4 Computational content of CoT traces

To check that opening the channel exposes computation-bearing traces rather than a longer answer-format policy, we parsed continuation records generated by the trained models (n=2560 completions, index and strindex, two seeds each; the parsing rule was specified before counting). Three measures agree. (i) Removing the reasoning channel collapses the readout: for index at seed 42, the same trained model evaluated in the answer-only setting reaches direct margin AUC 0.44 versus post-CoT 1.00 (Table 14), showing that the trained model's answer performance depends on which channel is available. (ii) Every parsed completion states the decisive computed index before answering (2560/2560; e.g. "$i = 8 + 2 - 13 = -3 \ldots$ a valid negative index for a list of length 5"), so the parsed traces consistently contain the computed value. (iii) The trace's own last in-range/out-of-range verdict matches the emitted Yes/No in 86.2% of completions, with answer accuracy 99.7% on the set that states both a computed value and a verdict (only 3/2560 lack a stated verdict, and every completion states the computed value, so a with-versus-without contrast is degenerate—near-total presence is itself the finding). The residual 14% verdict–answer gap mixes verdict phrasing the pre-specified parser misses with genuine restatement; we report the residual gap as measured. These measurements show that the parsed traces contain explicit computed values and that the trained model performs better with the reasoning channel available.

E.5 A pre-registered splice intervention on the stated value

We splice a corrupted value into the trace at the span where it states the decisive index and let the same trained model continue greedily (maximum continuation length of 512 tokens, using the pre-specified final-answer parser; the preregistration, deviation report for the retired first attempt, and continuation records are not included in this anonymous supplement). The implementation check—the same-value control must reproduce the original answer on ≥ 0.85 of prompts per family–seed condition—passed on all four family–seed conditions (control reproduction 1.000 on each; no exclusions, 320/320 prompts included).

The pre-registered pooled readout comes first: across 320 included continuations, corrupting the stated value flips the answer in 7.5% and the continuation reproduces the original verdict in 92.2% (0.3% other)—the pre-registered re-derivation outcome (an observable outcome classification—return to the original verdict after a contradictory splice—not by itself an identification of the model's internal algorithm). Of the 295 recoveries, 293 (99.3%) contain prespecified recalculation or range-check markers, and all 8 manually audited examples visibly recalculate, often rejecting the splice outright ("the calculation for i is $7+4-7 = 4$, not 999"). The per-condition follow

Table 16: Splice-and-continue intervention (pre-registered). *follow* = continuation adopts the implanted verdict; *recover* = reproduces the original verdict; *other* = neither (parse failures retained). Direction keyed to the validity of the stated value: valid→OOR corrupts to out-of-range 999; invalid→IR corrupts to in-range 0. Same-value control rates 1.00 in every condition; D = pooled mean of $e_{\text{flip}} - e_{\text{pres}}$ (final-answer logit-margin effects vs. the same-value control), prompt-clustered bootstrap CI. The pooled logit-margin contrast uses n=319 complete final-margin records (one index (s42) margin is missing), whereas follow/recover counts use all 320 continuations.

Family–seed condition	Direction	n	Follow	Recover	Other
index (s42)	valid→OOR	43	0	42	1
	invalid→IR	37	0	37	0
index (s7)	valid→OOR	43	0	43	0
	invalid→IR	37	0	37	0
strindex (s42)	valid→OOR	45	0	45	0
	invalid→IR	35	24	11	0
strindex (s7)	valid→OOR	45	0	45	0
	invalid→IR	35	0	35	0
pooled	both	320	24	295	1

counts are $0/80$, $0/80$, $24/80$, and $0/80$; all follow outcomes occur in the strindex (s42) invalid→in-range cell (Table 16). The semantic-specificity contrast is as follows: verdict-flipping edits shift the final-answer logit margin more than verdict-preserving edits against the same-value control pooled ($D = +3.98$ margin units, prompt-clustered bootstrap 95% CI $[2.29, 5.69]$, 10,000 resamples), but the per-condition sign agrees in only $2/4$ family–seed conditions (strindex $+18.70/+1.75$; index $-3.55/-1.07$), below the pre-registered $3/4$ gate, so we make no semantic-mediation claim; the interval quantifies prompt-level uncertainty conditional on these four trained models and does not establish generalization across trained models, and the positive pooled contrast is dominated by strindex (s42). Two structural notes: the corruption locus is set by trace grammar (bracket access vs. concluded-equality fallback) and differs systematically between trained models—the two trained strindex models write 98.75% bracket vs. 98.75% fallback traces, and follows occur only at fallback (verdict-nearer) loci—so family and locus are confounded by design; and within the one family–seed condition with follow variance, proximity to the verdict tracks following (exploratory, rank $\rho = 0.37$). A retired first run of this intervention used 160-token continuations and first-answer parsing, contrary to the preregistration; its numbers are unused, and its deviation report is outside this anonymous supplement.

F Cross-Model Evidence and Task-Scope Tests

F.1 Paired training endpoints

The following comparisons distinguish paired training endpoints, descriptive dissociation in unadapted models, and the Llama coverage repeat run; their measurements and success criteria differ.

The supervised-installs / RLVR-does-not gap on *index* holds across two independent RL seeds: RLVR margin AUC $0.60/0.61$ for Qwen3-4B and $0.71/0.48$ for Llama-3.1-8B, in every case far below the SFT values reported for those models in Table 17 (0.96–0.98). In a separate Qwen2.5 repeat run, SFT gives 0.97 on *index* and 0.51 on *zerodiv*, versus 0.963 and 0.485 in the original seed-42 run (Table 9). These are independent measurements; across the retained SFT loaders, *zerodiv*'s training split has no Yes examples, so the low endpoints do not establish SFT resistance. At the largest scales we report the dissociation only (Table 18): it holds at 27B (Gemma-2) and 70B (Llama-3.1), with computed-family probe values 67–82% versus answer behavior at chance. We omit the supervised/RLVR contrast for Gemma-2-27B because both its supervised and RL arms collapsed to a near-constant predictor on the held-out set (margin AUC ≈ 0.50, even with the eager-attention setup Gemma-2 requires for its logit soft-capping). We omit it for Llama-3.1-70B because training at that scale is out of budget.

Table 17: Supervised fine-tuning versus RLVR across model families (answer-margin AUC on computed families). On *index*, SFT reaches near ceiling while computed-only GRPO stays far lower; *zerodiv* moves little under SFT across model families, but the retained SFT loaders use a split with no Yes examples for this family, so it is not treated as an SFT-resistance control. Bold marks the SFT arm for *index*. RLVR values are seed 42; the running text reports a second seed (123) for Qwen3-4B and Llama-3.1-8B. The SFT entries in the `Qwen2.5-3B-Instruct` row come from an independent robustness repeat run, rather than the original SFT measurements in Table 9; these are two independent measurements, not a confirmation of the original values.

	index margin AUC			*zerodiv* margin AUC		
Model	Base	SFT	RLVR	Base	SFT	RLVR
Qwen2.5-3B-Instruct	0.44	**0.97**	0.49	0.52	0.51	0.50
Qwen3-4B-Base	0.44	**0.98**	0.60	0.49	0.59	0.48
Llama-3.1-8B-Instruct	0.65	**0.96**	0.71	0.51	0.56	0.57
Qwen3-14B-Base	0.66	**0.97**	0.72	0.55	0.59	0.58

F.2 Descriptive scale and surface-family measurements

Table 18 compares probes and answer behavior across 3B–70B Qwen, Llama, and Gemma models; Base and Instruct variants are identified per row. Across these models, *index* probe accuracy is 66–78% while realized answers remain at or near chance. Paired training endpoints are reported only at 3B, 4B, 8B, and 14B. The SFT/RLVR endpoints in Tables 10, 17, and 19 are specific to the model, setting, and seed; they do not establish uniformly near-ceiling SFT or a seedwise SFT-over-RLVR ordering across all settings.

Table 18: Cross-check across selected unadapted models: the probe–behavior dissociation on the computed families spans 3B–70B and Qwen, Llama, and Gemma. Rows distinguish Base and Instruct variants; "unadapted" means before task-specific training. In Panel A, probe accuracy uses stratified 5-fold CV over the deduplicated exact-program evaluation set, so identical programs cannot cross folds; *index* has $n = 200$ ($100/100$ labels) and *zerodiv* has $n = 170$ ($100/70$). Answer accuracy is realized Yes/No accuracy: at or near chance throughout, except one below-chance cell (*zerodiv* on Qwen3-14B, 41%). Panel B retains the distinct surface-family within-family probe and realized-answer measurements; all entries are percentages.

Panel A: computed-family scale comparison

	index		*zerodiv*	
Model	Probe (within-family)	Answer	Probe (within-family)	Answer
Qwen2.5-3B-Instruct	70%	50%	66%	59%
Qwen3-4B-Base	78%	50%	73%	59%
Llama-3.1-8B-Instruct	66%	50%	61%	55%
Qwen3-14B-Base	78%	50%	72%	41%
Gemma-2-27B-Base	75%	50%	82%	58%
Llama-3.1-70B-Base	68%	52%	67%	59%

Panel B: additional surface-family accuracy (percent)

Model	Metric	key	type	attr	value
Qwen3-4B-Base	Probe	98	100	99	100
	Answer	50	50	50	50
Llama-3.1-8B-Instruct	Probe	72	100	66	99
	Answer	50	50	50	50

F.3 Llama coverage comparison

On Llama-3.1-8B, forced coverage restores the answer-logit signal: its mean absolute derivative rises from approximately 0.003 to 0.33, and the zero-reward-variance skip rate falls from 0.997 to zero. On index and key, however, the supervised-versus-RLVR contrast is seed-dependent. SFT

Table 19: Supervised and forced-coverage training on Llama-3.1-8B-Instruct (native chat format, two seeds). Margin AUC per seed; forced-coverage RLVR is the answer-logit-signal-restored arm (mean absolute gold-logit derivative 0.33, no groups skipped for zero reward variance). "coverage below SFT in both seeds" is the criterion used here. Only *regexgroup* and *runtime_prop* pass; *det3mod* is the negative control (SFT also fails).

Family	SFT (seed 1, 2)	Forced-coverage RLVR (seed 1, 2)	Coverage below SFT in both seeds
regexgroup	0.82, 0.92	0.59, 0.59	yes
runtime_prop	0.82, 0.75	0.56, 0.46	yes
key	1.00, 0.58	0.33, 0.65	no (seed 2)
index	0.94, 0.53	0.47, 0.63	no (seed 2)
det3mod (negative control)	0.51, 0.56	0.50, 0.50	not applicable (SFT did not install)

reaches 0.94 and 1.00 in one seed but 0.53 and 0.58 in the other, so the per-seed criterion fails. Regexgroup and runtime_prop have forced-coverage RLVR below SFT in both seeds. Because key is not a consistently successful coverage control in this protocol, these results test scope rather than replicate the Qwen positive-control design.

F.4 Natural-task entry checks

Under pre-registered criteria (base probe ≥ 0.65 and base margin in the chance band), neither of two natural tasks is locked for Qwen3-8B. On GSM8K answer-verification pairs (n=2638, class-balanced, with negatives drawn from the model's wrong answers), base margin is 0.776 (probe 0.763; surface-only control 0.572) and remains 0.779 on problems the model cannot solve generatively. On execution-labeled MBPP runtime-error prediction (n=2612), the cross-validated probe over the full corpus is 0.914 and margin 0.748: a $+0.17$ probe–margin gap, but not a chance-margin lock (surface-cue control 0.614). Every natural task measured therefore lies in the high-probe, high-margin regime where outcome reward has signal to amplify. Only the controlled testbed isolated the locked high-probe, chance-margin regime; this supports no natural-task credit-assignment claim.

A separate pre-registered probe-only hunt across three natural paradigms was stopped after independent adversarial review: familiarity, popularity, and lexical difficulty confound the target bit without a surface-matched counterfactual. The testbed supplies that control by construction (identical surface, different computed label). This is why the natural-task result is a scope boundary rather than a failed repeat run.

F.5 Multiclass and arithmetic stress tests

F.5.1 Multiclass stress test (negative)

To probe whether the binary lock is an artifact of the two-way answer format, we built a three-way execution-verified task (exception type in {none, IndexError, ZeroDivisionError}) with static surface baselines held near chance by near-boundary computed operands. Two regimes appeared, and neither is a clean lock. (i) With tight operand margins the feature was not linearly decodable from the base residual under shape-grouped cross-validation (probe macro-AUC 0.55 at 4B, static surface maximum 0.53). (ii) Widening the operand margin made the feature decodable at 14B (probe macro-AUC 0.71–0.72, static surface baselines still ≤ 0.60), but the forced-choice answer channel then collapsed to a near-constant single-class prior (the majority class recalled at ≈ 0.94, the other two near 0)—a constant-prior collapse rather than a balanced at-chance readout. We therefore report the multiclass setting as a negative stress test and do not use it as evidence. The supervised-versus-RLVR dissociation does not rest on it. The defense against a constant-class reading is the supervised-versus-RL asymmetry itself: SFT installs the readout on the same task where the base and outcome-RLVR sit at constant-class chance, which a constant-class default cannot explain.

F.5.2 ARITHMETIC YES/NO SCREEN (NEGATIVE)

We screened a second controlled task construction—six arithmetic Yes/No template families (addition–comparison, divisibility, modular equality, digit-sum parity, product-plus-constant parity, chained comparison)—for another probe–behavior dissociation in Qwen3-14B-Base. Families the probe decodes are also expressed in the answer (addition–comparison probe AUC 0.998 / margin AUC 0.996; divisibility $0.999/1.000$; chained comparison $0.977/0.937$). The weakest-answer family is not linearly decodable under this probe (digit-sum probe 0.522), and the closest candidate (product-plus-constant parity, probe 0.861) is still expressed in the answer well above chance (0.634). In this scout, families with a linearly decodable label were also expressed in the answer, so we found no second clean probe–behavior dissociation. The observed dissociation therefore remains specific to the tested runtime-error construction.

F.6 A CANDIDATE THAT DID NOT MEET ITS ENTRY CRITERION

A pre-registered no-training base-model decodability check on Qwen2.5-3B did not meet the candidate's scientific entry criterion. Implementation checks and independent recomputation succeeded, but no training matrix was run. The retained preregistration and terminal record identify this entry decision; it gives no phase-diagram or causal conclusion and is not a failed training result.

F.7 DEGENERATE MBPP AMPLIFICATION CHECK

The diagnostic map predicts that in the high-margin regime, answer-only outcome reward has something to amplify. We pre-registered a check on the real task already used in our natural-task scope check (Appendix F.4): MBPP runtime-error prediction (the question-ID-disjoint split specified for this check, train $2{,}207$ / eval 405, balanced; base margin measured by the evaluation protocol 0.763, probe 0.857), standard answer-only GRPO, two seeds, the same budget as every other arm ($2{,}400$ groups per seed). The criterion, interpretation regions, and a degeneracy rule (any seed with zero-variance skip rate ≥ 0.90 makes no validation claim in any region) were specified before training. The degeneracy condition was met: sampling was near-saturated (skip rate $0.975/0.982$; $61/44$ groups updated), so the amplify prediction is untested at this operating point and we claim no validation. On the sparse variance-bearing groups, the observed margin rose from 0.763 to $0.886/0.838$—consistent with the amplification regime, though per the pre-registered rule not counted as validating it. The saturation itself mirrors the testbed: the reward-variance collapse that motivates the coverage control dominates a real high-margin task as well; the contrast is that here the readout is already expressed, and the few mixed groups moved it upward, whereas on locked families even full coverage does not install.

G LOCAL GRADIENT ANALYSIS

G.1 IID ANSWER-LOGIT HEURISTIC AND ITS SCOPE

The bisection result poses a puzzle. Scaling-only and centering-only differ both in their correct-sample weights and in centering-only's negative weight on incorrect samples; a weighting-matched base-state gradient comparison isolates the latter, yet the empirical arms diverge sharply. The following single-logit calculation is an IID on-policy heuristic, not a derivation of the measured forced-coverage bisection arms: their groups contain forced answer slots, so the independence assumption below does not hold on that rollout stream. Within the heuristic, abstract the answer channel to one logit z with $\pi(\text{correct}) = p$, and let a GRPO group of size G carry rewards $r_i \in \{0, 1\}$ with group mean $\bar{r}$. For the centered advantage $A_i = r_i - \bar{r}$, the cross terms in $\mathbb{E}\big[\frac{1}{G}\sum_i (r_i - \bar{r})\,\nabla_z \log \pi(a_i)\big]$ vanish because $\mathbb{E}[\nabla_z \log \pi(a)] = 0$ and samples are independent, so the expected update equals $(1 - \frac{1}{G})\,\mathbb{E}[r\,\nabla_z \log \pi(a)] = (1 - \frac{1}{G})\,p(1-p)$—strictly positive toward the gold answer, exactly as for the uncentered and scaling-only objectives (an action-independent baseline preserves expectation; the sample group mean introduces the displayed $1 - 1/G$ factor, which preserves direction). Within this IID answer-logit heuristic, all three updates point the same way, and no logit-level algebra distinguishes the installing arm from the blocking arm. The observed divergence (scaling-only installs at $0.823/0.925$; centering-only blocks at $0.638/0.750$ on three of four family–seed cells) is not explained by a distinct per-prompt incorrect-sample direction: because the binary-softmax

score gradients for the two answers are collinear within each prompt, the aggregate rotation can arise through objective-dependent reweighting of prompt-level margin gradients across inputs and training time. The cosine test measures this aggregate reweighting relative to the supervised update: adding only the negative-advantage term rotates the update away from the supervised install direction (rotation 0.29–0.60 at 4B, 0.21–0.30 at 14B). We therefore report the localization as gradient-direction evidence in feature space rather than as an answer-logit story. This note is an observation about the IID abstraction, not a derivation of the forced-coverage arms or a training-dynamics proof.

G.2 Endpoint weighting contrasts and a matched local-gradient test

The endpoint comparisons hold the prompt schedule, forced-sampling rule and random-number construction fixed, while each arm samples from its own current policy. They use direct log-probability losses without a PPO ratio or clipping term. For a given binary sample group, uncentered reward weighting and correct-completion cross-entropy select the same correct answers; this algebraic identity does not imply identical realized responses across diverged policies. We bisect the transform on two locked families. Scaling-only (adv $= r/\sigma$: positive weight on correct samples, none on incorrect) *installs* the readout (index 0.823, strindex 0.925, all four family–seed cells ≥ 0.78); centering-only (adv $= r - \bar{r}$: adds a negative-advantage term on incorrect samples) blocks it on three of four family–seed cells (index 0.638, strindex 0.750). These endpoints do not admit a single ordering across families: scaling-only installs both, centering-only blocks three of four cells, and RAFT improves mean AUC on the matched sampling protocol with mixed seed outcomes. Together with the weighting-matched gradient test below, they provide local evidence about the *negative-advantage term*. Because the endpoint arms also differ in per-sample positive weights, the endpoint contrast is not a one-term ablation; the matched test holds correct-sample weights fixed. The update is per-example correctly signed, so the block is not a naive sign error. Indeed, at the answer-logit level an IID single-logit heuristic predicts no block (Appendix G.1); this motivates testing whether the objectives differ in reweighting prompt-level margin gradients into shared features.

G.3 Weighting-matched 4B/14B base-state measurements

A weighting-matched gradient-geometry test probes exactly that: on locked *index* (Qwen3-4B, gradients at the base state), a correct-only update aligns with the SFT install direction at cosine ≈ 0.92, while adding *only* the group-centering negative-advantage term—with the same positive weight on correct samples held fixed—rotates it away (alignment 0.32–0.64 across three seeds; rotation 0.29–0.60). With the correct-sample weight held fixed, adding the incorrect-sample negative-advantage term changes the aggregate weighting of prompt-level margin gradients and rotates the update away from the supervised direction. The separate 8B scaling-only experiment reaches a higher AUC under the same sampling protocol as its coverage control (§3); it changes both correct- and incorrect-sample weights. The base-state geometry above is a separate diagnostic, not an identification of the component responsible for that endpoint difference. The block is also not specific to GRPO's normalization: on *index*, the leave-one-out advantage of RLOO (Kool et al., 2019; Ahmadian et al., 2024)—an exact $\frac{G}{G-1}$ rescaling of group centering—stays locked as well (0.61 mean; the advantage of Dr.GRPO (Liu et al., 2025) is algebraically our centering-only arm here), while on *strindex*, whose centering-only outcomes differed between seeds, the rescaling installs (0.83 mean): the class claim depends on the family (Appendix C.3). The same test on **Qwen3-14B** reproduces the direction (rotation 0.21–0.30 across three seeds; absolute alignment is lower, cosine ≈ 0.70), so the effect is not specific to the 4B model. This is evidence about the update direction at the locked state (4B and 14B, cross-scale to the 8B bisection), not a full training-dynamics proof. The erase branch of the dose-response is consistent with this reading.

H Probe-Gap Association and Its Scope

We examine whether the base probe–answer gap is associated with the supervised advantage over the tested on-policy RLVR recipe. The analysis is descriptive within the selected family set; the fresh-family comparison and shared baseline dependence limit its use as a prospective predictor. For each family we define a single scalar from the base model, the *probe gap*, $\mathrm{gap} = R - \mathrm{lock}$, where R is the hidden-state probe AUC (what is internally decodable) and lock is the base answer-margin AUC (what the answer channel already expresses). The outcome is the *supervised advantage*

SFTadv, the margin-AUC that supervised fine-tuning installs over what on-policy RLVR installs on the same data—i.e. how much of the readout RL leaves on the table. Each point is an independent task family on Qwen3-8B (seed-averaged); we report Spearman ρ with a $10{,}000$-permutation test (family as the unit) and a $5{,}000$-resample family bootstrap.

Association within the analyzed family set

Across the 15 analyzed families the probe gap correlates with the supervised advantage at $\rho = 0.84$ ($p < 10^{-4}$), and the relationship survives leave-one-family-out cross-validation ($\rho_{\mathrm{LOO}} = 0.77$, $p = 0.001$, RMSE 0.13), so the association is stable under leave-one-family-out within the analyzed family set; it is not a prospective prediction for newly selected families. The bootstrap 95% CI on ρ is $[0.57, 0.94]$. The result is not an artifact of the three families we added to enrich the locked regime: on the 12 original, unselected families alone the association remains ($\rho = 0.80$, $p = 0.002$), with the new families only tightening the interval. The 15-family set is *crtless*, *det3mod*, *falsecmp*, *guardedraise*, *index*, *key*, *mf_bytes_fromhex_fail*, *mf_deque_pop_empty*, *mf_heap_pop_empty*, *output_equals*, *regexgroup*, *return_type*, *runtime_prop*, *squarecmp*, and *type*; the three enrichment families are the three *mf_** entries, and the original set is the other 12. Families not already defined in Appendix A or glossed in §2 are templated comparison or stdlib-failure variants used only in this predictor set (*crtless*, *falsecmp*, *guardedraise*, *output_equals*, *squarecmp*, *mf_bytes_fromhex_fail*, *mf_deque_pop_empty*). Two correlational checks probe natural confounds: a difficulty baseline (the label-entropy proxy $p(1-p)$ of the base answer rate) has no significant association ($\rho = -0.44$, n.s.), while a calibration-robust gap $R - \max(\mathrm{lock}, 1 - \mathrm{lock})$ remains correlated ($\rho = 0.70$, $p = 0.004$). One caveat on interpretation remains: the gap subtracts the base answer margin, whereas the supervised advantage subtracts the RLVR answer margin. When RLVR remains near the base answer margin, the association may partly reflect this baseline dependence. The calibration-robust control addresses a reversed head but not this potential baseline dependence. We therefore read the association as a descriptive *regime-conditional indicator* in the locked regime, not as proof of an independent causal pathway.

Scope

On 28 fresh templated families, the association is weaker ($\rho = 0.34$, $p = 0.077$). Most of these families have near-ceiling SFT and RLVR endpoints, leaving little supervised advantage to distinguish. This difference in outcome range may contribute to the attenuation, but the fresh-family result does not validate a prospective predictor, including within a newly selected high-gap subset. The 15-family association remains a descriptive relationship in the analyzed set; its strength, the shared-baseline concern and the weaker fresh-family result should be interpreted together.

I Additional Related Work

This paper locates a boundary of answer-only RLVR rather than proposing a new algorithm or probing method: when a feature is already recoverable from hidden states, under what conditions does outcome reward train it into the emitted answer?

Sampling-support accounts

The closest prior account is that tested vanilla RLVR recipes mainly sharpen what the base model can already sample—pass@k coverage can narrow rather than expand under verifiable rewards (Yue et al., 2025)—and can remain support-constrained while contracting empirical answer support (Wu et al., 2025). These accounts operate at the policy/trajectory level. Our binary forced-coverage intervention saturates only support over the *final-answer action*; full trajectory support remains outside this control. The *index* family still does not install, showing that final-answer action support alone is insufficient on this constructed rollout stream, without refuting trajectory-support accounts. Our matched testbed therefore operates one level down, separately checking linear decodability, final-answer action support, the answer-logit learning signal, and a positive-control family. Boundary-aware curriculum RL can first introduce patterns with teacher guidance and then consol-

idate them with RL (Cai et al., 2026); accordingly, any "amplification" statement here is restricted to the tested direct answer-only outcome-reward arms, not RLVR in all forms.[1]

Representation-deficit accounts

A parallel line locates unlearnability in the representations themselves: some examples resist RLVR even when correct rollouts are present, because the needed features are absent or ill-formed (Chen et al., 2026). Our *index* family is complementary at the measurement level: the label is linearly decodable in the original Qwen2.5-3B study (probe accuracy 70%) and SFT installs, whereas the *det3mod* SFT-resistant tier does not install under the tested answer-only SFT recipe and is treated as an unresolved control. Failure of that recipe does not by itself identify a missing representation. The results delimit our diagnostic claim: a credit-assignment limitation where SFT installs and coverage-restored RL does not, and an unresolved tier where SFT also fails.

Optimization-pathology accounts

A third line attributes binary-reward GRPO failure to degenerate zero-variance groups and studies fixed-reference credit as a cure, alongside coverage and group-size routes (Nie et al., 2026). Motivated by that diagnosis, our intervention guarantees final-answer coverage rather than using fixed-reference credit. It removes zero-variance groups and supplies a nonzero answer-logit learning signal, and the *key* family installs. Its failure on the *index* family shows that zero-variance groups alone are an insufficient explanation on this rollout stream and motivates the credit-assignment analysis; it does not exhaust other sampler–objective or optimization interactions. The later negative-advantage-term localization is family-specific after mixed-reward groups and a nonzero answer-logit signal are supplied, not a claim that centering or negative weights generally prevent learning.

Signed cancellation and positive-only credit. Recent work studies signed credit at finer resolution. The cancellation hypothesis argues that coupling positive and negative rollouts can cancel shared-token directions while concentrating updates on success-specific tokens; SALT diagnoses signed low-rank cancellation in group-normalized gradient geometry and reweights that geometry (Cheng et al., 2026; Chang et al., 2026). POPO trains from positive rollouts, whereas HAPO reallocates sign-preserving token credit toward high-entropy positions (Xu & Fang, 2026; He et al., 2026). These are general optimizer or reasoning-trajectory analyses. Positive-only credit is used here as a matched-sampling diagnostic, not proposed as an optimizer, and we study a particular final-answer readout after both answer actions, mixed-reward groups, a nonzero answer-logit signal, and a same-intervention positive-control family are supplied. Cancellation can be useful elsewhere; our evidence is an endpoint contrast plus a separate aggregate-gradient rotation when the negative-advantage term is added, not a general failure of negative credit.

Process-aware credit assignment

A complementary line improves RLVR by making credit assignment fine-grained rather than uniform—using attention-derived token salience to route credit to the tokens that drive a solution, with reasoning-benchmark gains (Li et al., 2025). Process-supervised reward models make the same move at the objective level, scoring intermediate steps directly (Uesato et al., 2022; Lightman et al., 2023). That work asks how to distribute a sparse outcome reward across a long trajectory to raise accuracy; we ask the upstream question of whether uniform outcome-level credit installs a specific decodable readout at all. Our controlled comparison documents a learning gap under the tested outcome-level updates. Process-aware credit assignment offers a possible follow-up intervention, which we do not test here.

Probes, latent knowledge, and reasoning channels

Linear probes recover task-relevant information (Alain & Bengio, 2016; Hewitt & Manning, 2019; Tenney et al., 2019; Belinkov, 2022), while latent-knowledge methods recover Yes/No information

[1] RL/RLVR foundations: (Schulman et al., 2017; Christiano et al., 2017; Ouyang et al., 2022; Bai et al., 2022; Shao et al., 2024; Lambert et al., 2024; Guo et al., 2025).

from activations (Burns et al., 2023). Mechanistic work also uses readouts to study features before they surface (Li et al., 2023; Zou et al., 2023; Elhage et al., 2021; Nanda et al., 2023). Decodability need not imply use (Belinkov, 2022; Hewitt & Manning, 2019; Sudheendra & Srivastava, 2026). We do not call our probe "knowledge" or a causal route. Format residualization passes, while single-direction amnesic erasure (Elazar et al., 2021) does not block SFT (Appendix B.3). The cross-model dissociation therefore counters an inference from decodability to short-channel RL elicitation, not prior probe or latent-knowledge results. The SFT-seeded RL threshold result is consistent with work on low-likelihood correct solutions and KL-minimal updates (Nguyen et al., 2025; Shenfeld et al., 2025). Chain-of-thought exposes sampled intermediate tokens (Wei et al., 2022), if imperfectly (Turpin et al., 2023); our intervention distinguishes opening a channel from creating its signal: the successful Qwen comparisons expose readouts already expressed before task-specific training, while the tested Llama follow-up does not meet its installation criterion.