FROM SELECTION TO SWITCHING IN RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Data selection can improve training efficiency and performance, but its computational costs and effectiveness must be analyzed before it is applied. We compare uniform sampling, on-policy gradient alignment, and cached success-rate (SR) selection in reinforcement learning with verifiable rewards (RLVR). Matched-update MATH experiments under GRPO show selection gains over uniform sampling, with on-policy alignment stronger initially on average and cached SR stronger at later checkpoints. We then examine whether the selection strategy can be changed online during training, from alignment to SR, through executed switching runs and retrospective diagnostics. On the evaluated MATH/GRPO trajectories, switching achieves higher final reward at matched updates than continued alignment and SR started immediately after a shared prefix. Fixed-time switching experiments provide complementary evidence. RLOO controls show objective-dependent selector ordering, while MBPP extends reward and cost comparisons to code generation. With an existing reward cache, incremental selection-and-training costs are lower than continued alignment but higher than early SR. These results support changing selection strategies during RLVR while accounting for selection costs.

1 INTRODUCTION

On-policy gradient-based data selection can improve reinforcement learning with verifiable rewards (RLVR), but incurs substantial rollout and backward-pass costs (Li et al., 2026; Yang et al., 2026). Success-rate and reward-history methods prioritize prompts using observed outcomes without selection gradients (Baroian and Berger, 2026; Zheng et al., 2025). Motivated by these approaches, our cached success-rate (SR) selector favors balanced successes and failures using rewards collected once under an initial policy. It holds these estimates fixed, permitting subset selection without new scoring rollouts; the cited methods instead update priorities or reward histories during training.

Existing RLVR studies primarily develop effective criteria for selecting useful training examples (Li et al., 2026; Yang et al., 2026; Baroian and Berger, 2026), sometimes refreshing their scores as the policy evolves (Yang et al., 2026). Changing the selection strategy itself during training has received less attention. We investigate this question through comparisons with uniform sampling, comparisons among selectors across training states, and experiments that change strategies during training.

From selection gains to strategy transitions. Matched-update MATH experiments under GRPO (Shao et al., 2024) show selection gains over uniform sampling. On-policy alignment leads cached SR on average at the initial checkpoint, while SR leads at later checkpoints, with seed-level variation. This changing ordering motivates testing transitions. The same analysis yields the *SR-referenced gradient contrast* (SR-GC), an auxiliary current-gradient indicator. We distinguish retrospective signal diagnostics from online switching experiments, which use temporal confirmation to guide the transition. Across two evaluated OLMo-3 7B MATH/GRPO trajectories, switching from alignment to SR improves final reward over continued alignment and SR started immediately after a shared prefix by 2.48 and 0.96 percentage points on average, respectively (Figure 2).

Objective dependence and selection costs. RLOO controls change the preferred strategy, including the initial ordering, and MBPP extends reward and cost comparisons to code generation. With an existing reward cache, Switch has lower incremental selection-and-training cost than continued alignment but higher cost than early SR (Figure 3). We also compare rewards under a fixed total

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.

budget and costs to a common reward target, distinguishing learning gains from their computational requirements.

Our contributions are: (i) evidence for selection gains over uniform sampling and changes in selector ordering across training states and objectives; (ii) empirical evidence for the benefit of changing strategies from executed transitions and fixed-time switching; and (iii) selection-and-training cost measurements characterizing the performance–cost trade-off in MATH and MBPP.

Related approaches. Gradient-based selection spans instruction tuning and RLVR (Xia et al., 2024; Li et al., 2026; Yang et al., 2026); learned loss-change estimates guide pretraining selection (Yu et al., 2024). Other inexpensive signals include success-rate curricula and rollout filtering (Gao et al., 2026; Qu et al., 2026; Yu et al., 2025; Wu et al., 2026). Stored-response influence reuses responses for scoring (Zhu et al., 2026), whereas response replay reuses them for training (Wan et al., 2026; Mao et al., 2026). All our learners generate fresh training responses. Curriculum gains can change with training duration (Weinshall et al., 2018; Wu et al., 2021); decaying prioritization and warmup motivate later use of cheaper sampling (Loshchilov and Hutter, 2016; Zhang et al., 2026). Adaptive importance sampling, compute-constrained finetuning and budget-aware optimization account for scoring overhead (Katharopoulos and Fleuret, 2018; Yin and Rush, 2025; Xia et al., 2025; Wan et al., 2025). Our experiments connect state-dependent selection gains to executed strategy transitions and their reward and cost outcomes in RLVR.

2 Setting and Comparison Objectives

Selectors and current-policy rollouts. We use On-policy for current-policy gradient-alignment selection, SR for cached success-rate selection, and Switch for training that begins with On-policy and later transitions to SR. In the executed switching experiment, SR-GC supplies the auxiliary signal for this transition. Let π be the current policy at a selection step and β the initial policy used to collect stored responses and rewards. From the candidate pool $X = \{x_1, \ldots, x_n\}$, SR, Off-policy and uniform sampling each fix a subset of $k < n$ prompts before training, from which every update takes its prompts; uniform sampling draws this subset without replacement. On-policy instead scores a random candidate sample drawn from X at each step, using fresh current-policy rollouts for selection and gradient computation at every training step, and trains on its highest-scoring prompts. Training generates its own fresh responses and separate optimization gradients.

Gradient-alignment selection. For candidate and validation gradient estimates $\widehat{g}_i$ and $\widehat{v}$, and a fixed projection P, the score is

$$\widehat{s}_i = \frac{\langle P\widehat{g}_i, P\widehat{v}\rangle}{\|P\widehat{g}_i\|_2 \, \|P\widehat{v}\|_2}, \tag{1}$$

with score zero when either norm is zero. Gradients use leave-one-out reward advantages and a fixed CountSketch projection (Charikar et al., 2002). Off-policy alignment prepares selection gradients in advance from responses cached under β, using clipped full-trajectory importance weights and evaluating both candidate gradients and the validation direction under the scoring checkpoint's policy. The switching experiments compare On-policy and cached SR. Appendix B motivates full-trajectory correction by showing that omitting prefix or suffix correction can reverse rankings at arbitrarily small policy KL.

Cached success-rate selection. For K binary rewards R_{ij} stored under β, define

$$\widehat{p}_i = \frac{1}{K}\sum_{j=1}^{K} R_{ij}, \qquad \widehat{s}_{\mathrm{SR},i} = -|\widehat{p}_i - 1/2|. \tag{2}$$

The top-k subset S_{SR} favors balanced successes and failures under β. The initial-policy success estimates remain fixed at later checkpoints. Fixing the cache removes repeated scoring rollouts. Alternatives refresh difficulty using a learned value model (Gao et al., 2026) or recent reward dynamics (Zheng et al., 2025).

Three comparison objectives. Matched-update rewards compare learning under different selection strategies and transitions between them. A fixed total budget additionally charges scoring against time available for learning. Writing $J_m(T)$ for expected reward after training compute T, total allocation B, and diagnosis-plus-selection charge $C_{\mathrm{tot},m}$, the net advantage over uniform training is

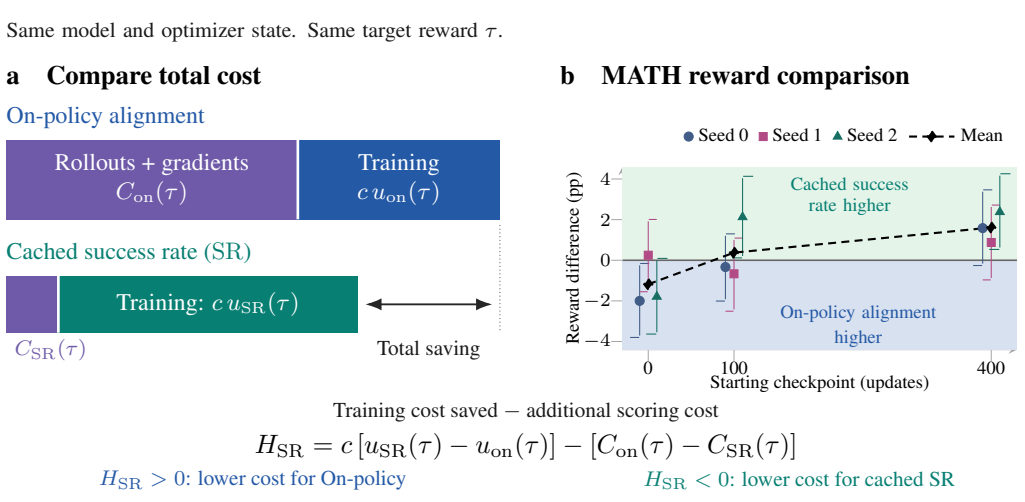
$$\Delta_{\mathrm{net}}(B) = J_m(B - C_{\mathrm{tot},m}) - J_{\mathrm{rand}}(B). \tag{3}$$

Appendix A.2 separates this matched-training benefit from the opportunity cost of scoring. A third comparison fixes reward and measures cost to reach it.

Common-target cost criterion. From the same model and optimizer state z, let $C_a(\tau)$ be the additional selection cost of action a through target attainment, including setup and repeated scoring, and $T_a(\tau)$ its cumulative training cost through the first evaluated checkpoint reaching target reward τ. For an alternative $a \in \{\mathrm{SR}, \mathrm{rand}\}$ with cheaper selection, define its total cost from z minus that of on-policy alignment:

$$\begin{aligned} H_a(z,\tau) &= [C_a(\tau) + T_a(\tau)] - [C_{\mathrm{on}}(\tau) + T_{\mathrm{on}}(\tau)] \\ &= \underbrace{T_a(\tau) - T_{\mathrm{on}}(\tau)}_{\text{training cost saved by alignment}} \underbrace{- [C_{\mathrm{on}}(\tau) - C_a(\tau)]}_{\text{extra selection cost}}. \end{aligned} \tag{4}$$

A positive value means retaining on-policy alignment costs less at that target; a negative value favors the alternative. Under equal per-update cost c, $T_a = c\,u_a$, where u_a counts updates after the common branch state. With unequal update costs we use cumulative measured costs instead. Costs are conditional on the same z and expressed in GPU-hours. Incremental costs exclude the shared prefix and pre-existing cache construction, but charge new selection, setup and metered retries; evaluation is separate (Appendix C.1). H_a is an observed cost difference: SR-GC neither predicts nor optimizes H_a. SR-GC guides only the current selection decision; target attainment and cost availability are reported separately.



(a) Schematic costs with equal update cost c. (b) Measured rewards after matched updates.

Figure 1: Learning and cost comparisons answer different questions. (a) Illustrative selection-plus-training costs to a common target, with equal per-update cost c. (b) Measured cached-SR minus on-policy reward after 100 additional GRPO updates. Pale markers and bars show individual seeds and paired 95% question-bootstrap intervals. The dashed line joins seed means at distinct starting checkpoints.

3 Switching Data-Selection Strategies

Analysis of completed On-policy/SR comparisons motivates SR-GC, a simple indicator computable during training from current gradients. Online switching uses only checks available through the current checkpoint; completed-run rewards are used separately for diagnostic analysis.

An auxiliary gradient contrast. Let t denote the cumulative training-update count at a checkpoint. Let $S_{\mathrm{on},t}$ denote the On-policy candidate set before training selection, and S_{SR} the cached-SR comparison set. Fresh ranking rollouts provide current-policy gradients for both sets at each selection step. At a check, SR-GC uses these scoring gradients $\widehat{g}_{i,t}$ and the validation direction $\widehat{v}_t$ already computed for selection at that same training step. The equally sized subsets $S_{\mathrm{on},t}$ and S_{SR} come from the same candidate pool and remain fixed during this measurement at the current policy. For a fixed check interval Δ, define the *SR-referenced gradient contrast* and its three-check mean:

$$D_t = \left\langle P\widehat{v}_t,\ \frac{1}{k}\sum_{i\in S_{\mathrm{on},t}} P\widehat{g}_{i,t} - \frac{1}{k}\sum_{i\in S_{\mathrm{SR}}} P\widehat{g}_{i,t} \right\rangle, \qquad \overline{D}_t = \frac{D_{t-2\Delta} + D_{t-\Delta} + D_t}{3}. \tag{5}$$

Both means use the full scoring sets, before selecting the smaller training batch. Overlapping prompts cancel. Unlike cosine ranking, D_t retains gradient magnitudes; its negative sign favors SR and its nonnegative sign favors on-policy alignment in the common validation direction. Because D_t reuses the selection gradients, it is not an independent-sample estimate.

Temporal confirmation. A single negative can reflect sampling noise. Requiring a second negative avoids switching on a single negative observation. Two consecutive negative checks trigger a switch to SR. For a negative–nonnegative–negative sequence with endpoints $-a, -c$ and intervening value $b \geq 0$, we switch only when the three-check mean is negative, that is, when $b < a + c$: an intervening value large enough to offset both negatives postpones switching. Otherwise, the latest negative check starts a new confirmation window. A nonnegative third check clears the window. Confirmation requires complete consecutive measurements. The zero threshold follows the local contrast direction; subsequent rewards evaluate the decision.

Online switching procedure. Before training, fix the cached SR subset and check schedule. At each check during On-policy training, compute D_t from that step's selection gradients on the same training path. On confirmation, retain the model and optimizer state, switch to S_{SR}, and stop on-policy scoring and checks. Otherwise continue on-policy selection. Training always generates separate fresh responses.

Computation and evaluation. Computing D_t requires averages and inner products over the already computed gradients of the two scoring sets (at most $2k$ distinct prompts). Temporal confirmation adds $O(1)$ scalar operations and memory per check. SR-GC adds no rollout generation or backward pass; generation and gradient computation are already charged to On-policy selection. These scoring gradients differ from normalized training gradients and AdamW updates (Appendix C). Online switching is evaluated by subsequent training rewards (Section 5.2); retrospective sign agreement uses existing continuation outcomes (Section 5.6).

4 Experimental Design

Model and selection. We use pretrained OLMo-3 7B base weights and GRPO with verifiable rewards. MATH has a pool of 400 prompts. Each online On-policy step draws 40 candidate prompts uniformly at random from this pool, scores them, and selects the four with the highest gradient scores for training; the other selectors fix a 40-prompt subset before training. During On-policy selection, the scoring stage also evaluates the 40 SR comparison prompts at every step; SR-GC compares the two 40-prompt mean gradients. Gradient scoring uses a 4096-dimensional CountSketch projection. Off-policy full-trajectory importance weights are clipped to $[0.1, 10]$. SR-GC checks every 25 updates are separate from every-step selection. Detailed protocols and comparison units appear in Appendix C and Table 3.

Learning comparisons. The matched-update experiment compares uniform sampling, cached SR, on-policy alignment and off-policy alignment at MATH starting checkpoints $d \in \{0, 100, 400\}$, where d counts updates before continuation, for the development seeds 0–2: 36 continuations. Each uses four selected prompts per update, eight fresh responses per prompt (32 responses per update),

one optimization epoch and 100 updates. Noninitial branches resume both adapter and optimizer. For example, s3/t25 denotes seed 3 and starting step 25.

Retrospective diagnostic analysis. We analyze the nine completed On-policy/SR pairs to relate current gradient contrasts to observed reward ordering. The stage summary and SR-GC agreement statistics use these existing outcomes (Section 5.6).

Single-check selection experiments. We use six held-out starts (seeds 3/4, steps 25/50/100); the step-50/100 states are separately prepared. One SR-GC check before continuation selects On-policy or SR. Training then uses that strategy without further SR-GC decisions; subsequent rewards evaluate the choice.

Online switching experiments. Figure 2 compares continued On-policy, cached SR, uniform sampling, and a later switch to SR. All four paths share the first 25 On-policy updates. During its own continuation, Switch computes D_t every 25 updates by reusing the gradients already computed for selection. The temporal rule switches to cached SR at step 125 for seed 3 and step 100 for seed 4, retaining the model and optimizer state. All four paths are evaluated at total step 275, including the shared prefix.

Fixed-time switching experiments branch after 25, 50 or 100 shared on-policy updates, comparing continued alignment with a switch to SR. Each starting step has its own common prefix.

A/B replication for diagnostics. For diagnostic analysis, we measure the contrast in Equation (5) using independent reference batches A and B, with separate candidate responses and disjoint validation prompts, both separate from ranking split R. In the retrospective analysis and single-check selection experiments, we use $D_t^{AB} = (D_t^A + D_t^B)/2$, abbreviated as D in their tables. Each term is a complete inner product with its own validation direction. This mean combines replicates at one checkpoint; $\overline{D}_t$ in Equation (5) averages successive checks. A and B need not both be negative. Replication tests sampling sensitivity.

Response budgets. The separate full-pool diagnostic covers MATH and MBPP, five seeds and $d \in \{0, 25, 100, 400\}$ updates: 40 conditions, with one adapter and optimizer history per dataset–seed family and eight cached initial-policy responses per candidate. For MATH, this diagnostic uses $n = 400$ candidates and $k = 40$ selected prompts. Each full-pool measurement generates 32 current-policy responses per candidate: 16 for ranking diagnostics and eight each for A and B. The primary diagnostic ranking uses eight responses. A/B diagnostic comparisons use the On-policy and SR subsets of 40 prompts each: at most 80 distinct prompts, with eight responses per reference and 25 disjoint validation prompts per reference. Appendix C specifies subgroup averaging and all splits.

Evaluation and uncertainty. MATH evaluation uses 300 training-split questions (MATH-300) excluded from selection and learning; MATH-500 supplies candidates. Public evaluation uses AIME 2024/2025 (30 each), AMC 2023 (40), GSM8K (200), and MATH outside MATH-500 (200). Reward averages binary correctness over eight responses per question and then equally over questions; public macro reward weights the five sets equally. MBPP uses executable-test rewards on 300 held-out questions, with eight responses for endpoints and four for intermediate evaluations. Intervals use 10,000 paired question-bootstrap resamples conditional on trained policies; checkpoints within a seed share a training history. Public per-set intervals are unadjusted for multiple comparisons.

Objective and cost controls. RLOO changes the training objective while retaining the source states, selection procedures, update counts and evaluation settings (Appendix C.1). Fixed-budget MATH selectors have separately executed random controls. MBPP matches training allocations. GPU-time multiplies elapsed time by allocated GPUs and sums over selection operations. A correlation-gate control uses 40 uniformly sampled pilot prompts, choosing cached SR when a lower Fisher correlation bound is at least 0.25 and random otherwise (Appendix E).

Table 1: Reward gains over uniform random sampling after 100 additional updates (percentage points; mean of three seeds). d is the starting checkpoint in updates. The public macro gives equal weight to five benchmarks. Per-seed results appear in Appendix D.

	MATH-300			Public five-set macro	
Selector	$d = 0$	$d = 100$	$d = 400$	$d = 0$	$d = 400$
Cached success rate	-0.60	+1.15	+1.74	+0.35	+0.86
On-policy alignment	+0.58	+0.78	+0.12	+1.02	+0.06
Off-policy alignment	-0.58	+1.51	+0.82	+0.50	+0.22

5 RESULTS

5.1 STATE-DEPENDENT LEARNING GAINS

Selection gains depend on strategy and training state. Matched-update MATH experiments show that selection can improve reward over uniform sampling, with gains depending on the strategy and starting state. MATH-300 and the public macro both favor on-policy alignment over cached SR initially and cached SR at the late checkpoint (Table 1; Figure 1b shows the direct MATH contrasts). At the intermediate MATH checkpoint, off-policy alignment has the largest mean gain over random.

A mean reversal does not determine every state. At step 100, two of three seeds favor on-policy alignment even though the mean favors SR; at step 400 all three favor SR. Three of the nine direct paired intervals exclude zero. The complete seed-level gains over random are listed in Table 5. These comparisons motivate reconsidering the selection strategy as training progresses.

The evaluation target also affects selection gains. SR's initial gain over random is positive on the public macro but negative on MATH-300. On-policy's initial and SR's late public macro gains are positive on all three seeds. Seed-level public macro results are in Appendix D.

5.2 LEARNING BENEFITS AND COSTS OF ONLINE SWITCHING

Switch exceeds the On-policy and SR controls at matched updates. Switch changes to SR at step 125 for seed 3 and step 100 for seed 4, then continues learning on the SR subset. At total step 275, Switch reaches 33.04% and 37.00% for seeds 3 and 4. At the same step 275, SR rewards are 32.33% and 35.79%; On-policy control rewards are 32.12% and 32.96%. Switch exceeds SR by 0.71 and 1.21 percentage points, and On-policy by 0.92 and 4.04 points, respectively.

Fixed-time switching. Separate branches compare continued alignment with SR from $t \in \{25, 50, 100\}$. Among the eight development states whose two branches end after 301–319 additional updates, SR finishes higher in six (mean difference $+1.11$ points). It finishes higher in four of six held-out states ($+1.18$ points). Appendix D.4 reports all trajectories and the unequal-length s1/t50 endpoints excluded from the development mean.

Selection and training costs are metered separately. Scoring incurs costs for rollout generation and gradient computation. Purple regions in Figure 3 sum these costs, score calculation and selection-stage overhead. On-policy accumulates this cost throughout learning; Switch stops at its SR transition. At total step 275, Switch's mean post-prefix training cost is 19.12 GPU-hours. Shared SR preparation costs 1.231 GPU-seconds on average, charged once to both SR and Switch. This measures reading and ranking cached rewards. Figure 3 reports component means for all four methods; Appendix Table 11 gives individual seeds.

Switch changes the reward–cost trade-off. At step 275, mean rewards are 35.02% for Switch, 32.54% for On-policy, 34.06% for SR and 30.21% for Random. The corresponding reported selection-and-training costs are 21.74, 26.22, 19.20 and 20.18 GPU-hours. Switch gains 2.48 percentage points over On-policy; against SR it gains 0.96 points at approximately 13.2% higher cost. These are means over the same two seeds at matched updates (Figures 2–3), not common-target costs.

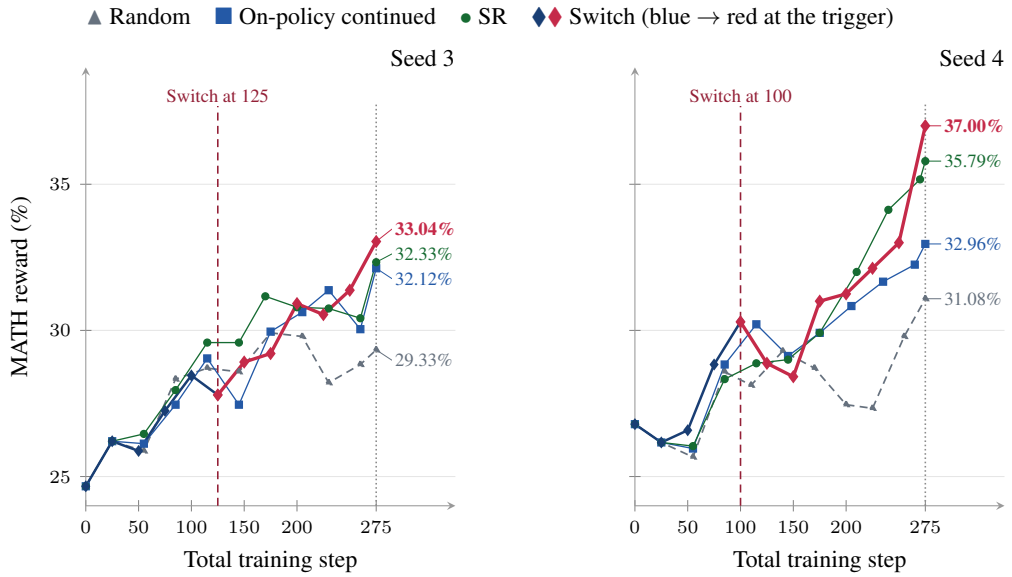


Figure 2: Online switching for seed 3 (left) and seed 4 (right). All paths share 25 On-policy updates and finish at step 275 (dotted lines). Controls are Random (gray dashed), continued On-policy (blue squares), and cached SR (green circles). Switch (thick blue, then red) changes from On-policy to SR at steps 125 and 100, respectively, after two consecutive negative D_t checks using the available selection gradients. Each point averages eight responses on each of 300 common evaluation questions; right-hand labels give endpoint rewards. Switch is evaluated every 25 steps, controls roughly every 30 and at step 275.

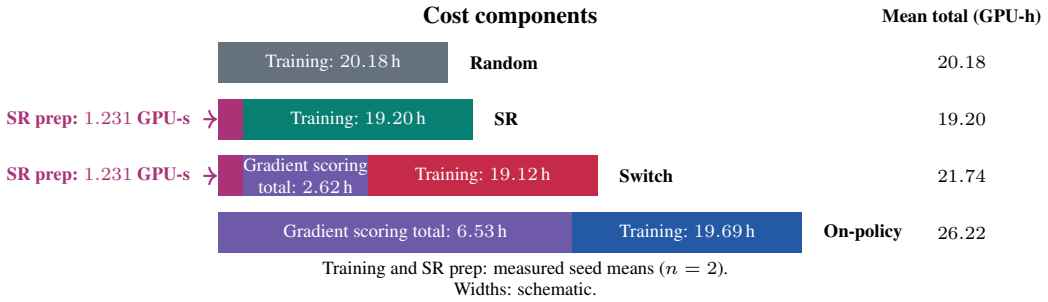


Figure 3: Cost components for Random, SR, Switch and continued On-policy, averaged over seeds 3 and 4. Purple totals sum selection rollouts, gradients, scoring and overhead. Training includes fresh rollouts and gradient updates at every step (four GPUs). Arrows mark one-time SR subset preparation from an existing cache. Widths are schematic; labels give GPU-hours, except SR preparation in GPU-seconds. Seed-level values appear in Table 11.

5.3 Selection-Inclusive Costs

Scoring consumes most of the MATH fixed budget. Under a common 29,040 GPU-second allocation, on-policy selection uses 82.6–87.5% of the budget. Its mean selection and training charges are 6.84 and 1.17 GPU-hours, leaving 14.3 updates versus 99.3 for its own random control. The mean reward difference is -1.74 points, with on-policy below random in five of six states (Table 7). The timing combines generation, gradient computation and selection-stage overhead.

Cached SR has a positive mean gain at similar continuation cost. Cached SR exceeds its own random control in five of six held-out states, averaging $+0.83$ points. One positive interval excludes zero. Each selector is compared with its own random control. Table 7 retains all state-level charges and intervals.

Costs to a common reward target. At a 35% reward target, H_{SR} is -8.70 GPU-hours at s2/t100 and -9.23 at s0/t50, the latter using reconstructed SR accounting (Table 13).

5.4 Domain and Objective Controls

MBPP learning gains can coexist with higher total cost. At four held-out states with paired rewards and costs, On-policy gains $+0.76$ points over random, but scoring plus training costs 16.69 versus 7.78 GPU-hours (2.14 times as much). Scoring is additional to the common 28,380 GPU-second training allocation. The full six-state reward comparison gives $+0.79$ points and is a different cohort from the four-state cost ratio (Appendix D.2).

The objective changes the preferred selection strategy. Cached SR has higher mean reward than On-policy under RLOO at all three checkpoints, including the initial state where GRPO favors On-policy. Our RLOO implementation uses leave-one-out reward centering and sums token gradients; GRPO divides centered rewards by within-prompt standard deviation and averages token terms. Removing these normalizers reweights prompts and response lengths (Appendix C). At step 100, all three seeds favor SR. At step 400, seed 0, all three selectors gain 13.71–13.88 points over their GRPO counterparts (Appendix D.3). This control tests selection under different training-gradient weights.

5.5 Score Gains over Uniform Selection

Figure 4 examines how noisy scores affect the advantage of selection over uniform sampling. Score gain is the selected prompts' mean score minus the candidate-pool mean, which is the expected mean under uniform selection. In the real-model panels, one response sample selects and an independent sample evaluates; we average both directions and standardize by score standard deviation. This measures selection-score advantage; Table 1 separately measures learning reward. At the evaluated budgets, on-policy scores carry little repeatable signal. Off-policy gains exceed the Gaussian prediction in 22/24 MATH and 13/24 MBPP estimator rows. Mean cached-SR gain is 0.80 times its mean prediction, so the Gaussian reference overstates this score's gain on average. Appendix A gives the model, proof and measurement details.

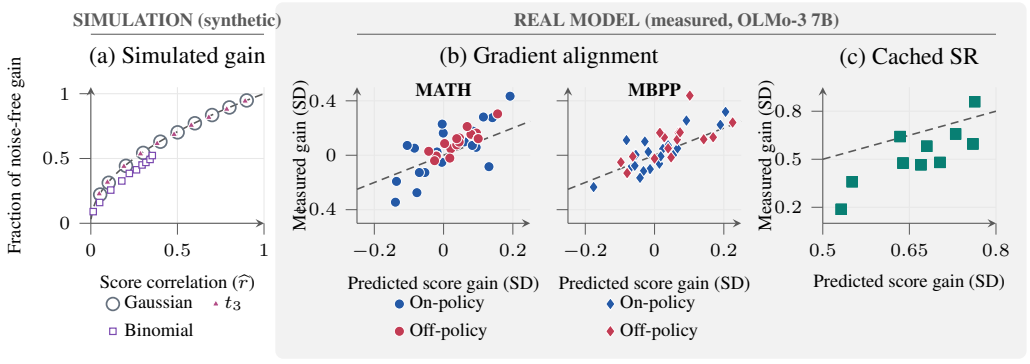


Figure 4: Selection-score gains over uniform sampling. (a) Mean simulated latent gain divided by mean noise-free gain; dashed curve: $\sqrt{\rho}$, where ρ is the correlation between repeated scores. (b,c) Independently measured gain versus a Gaussian prediction, in score standard deviations; dashed lines: equality. The prediction is $\hat{r}c_{k,n}$: sample score correlation times the expected top-k mean of n standard normal draws. In (b), circles/diamonds identify MATH/MBPP and blue/red identify on-/off-policy alignment. Panel (c) shows cached SR from both domains. Panel (a) is synthetic; shaded panels (b,c) use OLMo-3 7B measurements. Axis ranges differ across (a–c).

5.6 Retrospective SR-GC Diagnostics

Across the nine completed comparisons, the SR-GC sign agrees with the higher-reward strategy in seven states (Table 2).

Table 2: Retrospective comparison on nine completed 100-update On-policy/SR pairs. Agreement is with the higher-reward strategy; mean reward uses the corresponding observed outcomes.

Selection rule	Ordering agreement	Mean indicated reward (%)
Always on-policy	4/9	29.47
Always cached SR	5/9	29.74
SR-GC local sign	7/9	29.93

Separate summary using observed outcomes. The outcome-informed stage summary chooses On-policy at start 0 and SR at starts 100/400 (6/9, 30.13% mean reward) (Tables 5 and 17). It summarizes separate starting states using observed rewards.

Reference sensitivity in the result analysis. A, B and their mean agree with reward ordering in 5/9, 7/9 and 7/9 states; their signs differ at four states (Appendix F.1). Descriptive Pearson and Spearman correlations between D and the On-policy-minus-SR reward difference are 0.058 and 0.033.

5.7 Separate Single-Check Selection Experiments

In a separate six-state experiment, a strategy chosen by a single SR-GC check beats cached SR in 0/6 states and On-policy in 3/6, with mean differences of -2.17 and -0.99 percentage points, respectively (Appendix Table 12). Figure 2 separately evaluates online switching; Appendix F.2 analyzes A/B diagnostic histories along the continued On-policy controls.

6 Discussion and Limitations

These results motivate reconsidering data selection as training progresses. The evaluated switch improves final reward over continued On-policy and early SR, while reducing selection cost relative to continued On-policy.

Limitations. Online switching is evaluated on one backbone and two MATH/GRPO seed histories. Comparisons from a common prefix across intermediate switch points could clarify the contributions of timing and temporal confirmation. The projected local signal is reference-sensitive and does not predict long-horizon reward gains or optimizer steps. A/B signs differ at 4/9 states; approximate jackknife upper bounds for the 22 repeated A/B diagnostics do not fall below zero (Appendix F). The rule uses point estimates; its check interval and confirmation conditions are not optimized. SR-cache refresh remains untested. Repeated transitions would increase selection costs, although they may be needed as training progresses. Question-bootstrap intervals condition on trained policies and do not measure training-seed variation.

7 Conclusion

Matched-update MATH/GRPO comparisons show selection gains over uniform sampling, with On-policy alignment stronger initially on average and cached SR stronger at later checkpoints. Online and fixed-time switching support changing strategies during training. RLOO shows objective dependence; MBPP extends reward and cost comparisons to code generation. Given an existing cache, switching yields higher reward than continued alignment and early SR on the evaluated trajectories, with costs between them.

AI USE STATEMENT

AI tools were used to assist with manuscript drafting, language editing, and translation; literature search and discovery of related work; and research ideation and execution, including code development, debugging, data analysis, and figure preparation. The authors retain responsibility for all research decisions and for the accuracy and integrity of the manuscript, references, code, and reported results.

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. ACL 2024, pp. 12248–12267. https://aclanthology.org/2024.acl-long.662/

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program Synthesis with Large Language Models. arXiv preprint arXiv:2108.07732, 2021. https://arxiv.org/abs/2108.07732

Andrei Baroian and Rutger Berger. Prompt Replay: Speeding Up GRPO with On-Policy Reuse of High-Signal Prompts. arXiv preprint arXiv:2603.21177, 2026. https://arxiv.org/abs/2603.21177

Hubert E. Brogden. When testing pays off. Personnel Psychology, 2(2):171–183, 1949. https://doi.org/10.1111/j.1744-6570.1949.tb01397.x

William Brown. Some Experimental Results in the Correlation of Mental Abilities. British Journal of Psychology, 3(3):296–322, 1910. https://doi.org/10.1111/j.2044-8295.1910.tb00207.x

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding Frequent Items in Data Streams. ICALP 2002, pp. 693–703. https://doi.org/10.1007/3-540-45465-9_59

Lee J. Cronbach and Goldine C. Gleser. Psychological Tests and Personnel Decisions. University of Illinois Press, 1957.

Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. Prompt Curriculum Learning for Efficient LLM Post-Training. ICLR 2026. https://openreview.net/forum?id=zqOCacBD3P

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. NeurIPS 2021. https://arxiv.org/abs/2103.03874

Angelos Katharopoulos and François Fleuret. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. ICML 2018, PMLR 80:2525–2534. https://proceedings.mlr.press/v80/katharopoulos18a.html

Shipeng Li, Zhiqin Yang, Shikun Li, Xiaobo Xia, Hengyu Liu, Xinghua Zhang, Gaode Chen, Dong Fang, Ying Tai, and Zhe Peng. LearnAlign: Data Selection for LLM Reinforcement Learning with Improved Gradient Alignment. Findings of ACL 2026. https://aclanthology.org/2026.findings-acl.2009/

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. ICLR 2024. https://arxiv.org/abs/2305.20050

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. https://arxiv.org/abs/2503.20783

Ilya Loshchilov and Frank Hutter. Online Batch Selection for Faster Training of Neural Networks. ICLR 2016 Workshop. https://arxiv.org/abs/1511.06343

Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. RLVR without Ineffective Samples: Group Prioritized Off-Policy Optimization for LLM Reasoning. arXiv preprint arXiv:2606.01281, 2026. https://arxiv.org/abs/2606.01281

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can Prompt Difficulty be Online Predicted for Accelerating RL Finetuning of Reasoning Models? KDD 2026, Vol. 1, pp. 1240–1250. https://doi.org/10.1145/3770854.3780263

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv preprint arXiv:2402.03300, 2024. https://arxiv.org/abs/2402.03300

Charles Spearman. Correlation Calculated from Faulty Data. British Journal of Psychology, 3(3):271–295, 1910. https://doi.org/10.1111/j.2044-8295.1910.tb00206.x

Team Olmo. Olmo 3. arXiv preprint arXiv:2512.13961, 2025. https://arxiv.org/abs/2512.13961

Weilin Wan, Weizhong Zhang, and Cheng Jin. Computational Budget Should Be Considered in Data Selection. NeurIPS 2025. https://proceedings.neurips.cc/paper_files/paper/2025/hash/22c799f287fd05e7174fd65a3ce134af-Abstract-Conference.html

Xu Wan, Yansheng Wang, Wenqi Huang, and Mingyang Sun. Buffer Matters: Unleashing the Power of Off-Policy Reinforcement Learning in Large Language Model Reasoning. ICLR 2026. https://proceedings.iclr.cc/paper_files/paper/2026/hash/4ddea64304e8b069eed08c10c6feaa55-Abstract-Conference.html

Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks. ICML 2018, PMLR 80:5238–5246. https://proceedings.mlr.press/v80/weinshall18a.html

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When Do Curricula Work? ICLR 2021. https://arxiv.org/abs/2012.03107

Jiahao Wu, Ning Lu, Shengcai Liu, Kun Wang, Yanting Yang, Baijiong Lin, Chen Jason Zhang, Li Qing, and Ke Tang. Train at Moving Edge: Online-Verified Prompt Selection for Efficient RL Training of Large Reasoning Model. arXiv:2603.25184, 2026. https://arxiv.org/abs/2603.25184

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting Influential Data for Targeted Instruction Tuning. ICML 2024. https://proceedings.mlr.press/v235/xia24c.html

Tingyu Xia, Bowen Yu, Kai Dang, An Yang, Yuan Wu, Yuan Tian, Yi Chang, and Junyang Lin. Rethinking Data Selection at Scale: Random Selection is Almost All You Need. Findings of EMNLP 2025. https://aclanthology.org/2025.findings-emnlp.146/

Ningyuan Yang, Weihua Du, Weiwei Sun, Sean Welleck, and Yiming Yang. GradAlign: Gradient-Aligned Data Selection for LLM Reinforcement Learning. COLM 2026. https://arxiv.org/abs/2602.21492

Junjie Oscar Yin and Alexander M. Rush. Compute-Constrained Data Selection. ICLR 2025. https://arxiv.org/abs/2410.16208

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-Aware Data Selection for Efficient Pretraining with Data Influence Models. NeurIPS 2024. https://doi.org/10.52202/079017-3453

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. NeurIPS 2025. https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html

Yang Zhang, Amr Mohamed, Hadi Abdine, Guokan Shang, and Michalis Vazirgiannis. Beyond Random Sampling: Efficient Language Model Pretraining via Curriculum Learning. EACL 2026, pp. 5776–5794. https://aclanthology.org/2026.eacl-long.271/

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act Only When It Pays: Efficient Reinforcement Learning for LLM Reasoning via Selective Rollouts. arXiv:2506.02177, 2025. https://arxiv.org/abs/2506.02177

Erle Zhu, Dazhi Jiang, Yuan Wang, Xujun Li, Jiale Cheng, Yuxian Gu, Yilin Niu, Aohan Zeng, Jie Tang, Minlie Huang, and Hongning Wang. Data Efficient RLVR via Off-Policy Influence Guidance. ACL 2026. https://aclanthology.org/2026.acl-long.2141/

A Additional Theory and Score Diagnostics

We analyze how scoring noise affects ranking stability and independently evaluated score gain. Here gain refers to a selection-score differential; training reward is a separate outcome. These results characterize score measurements, not the performance of the SR-GC switching rule.

A.1 Measurement Noise and Score Gain

Even on-policy scores vary across response samples. To isolate measurement error, suppose example i has a latent score θ_i and we observe two measurements $y_i = \theta_i + \varepsilon_i$ and $y'_i = \theta_i + \varepsilon'_i$, where the errors are independent of the latent scores and of each other, with mean zero and equal variance. Reliability $\rho = \mathrm{corr}(y_i, y'_i)$ is the signal's share of measurement variance and depends on the response budget. For a rule $S(y)$ selecting k examples, let $G(S)$ be the expected selected latent-score mean above the population mean μ; the oracle selects directly from latent scores. Appendix A.2 gives the full definitions and proof.

Proposition 1 (Gaussian score-gain attenuation)**.** *Let $\theta_i \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma_\theta^2)$ and $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2)$ be independent, with $\sigma_\theta > 0$, $\sigma_\varepsilon \geq 0$, and $0 < k < n$. For every size-k rule based only on y, including randomized rules,*

$$G(S) \leq \sqrt{\rho}\, \sigma_\theta c_{k,n} = \sqrt{\rho}\, G(S_{\text{oracle}}), \tag{6}$$

with equality for $S(y) = \mathrm{TopK}(y)$. Here $c_{k,n}$ is the expected mean of the k largest of n independent standard normal draws.

Reliability determines the fraction of available score gain retained. At $\rho = 0.25$, noisy top-k selection retains half of the expected oracle score gain; low reliability therefore does not imply zero gain (an average over sampled pools, Appendix A.2). Absolute gain also depends on σ_θ and score units, so comparisons require a common target and unit; classical selection theory provides the broader context (Brogden, 1949; Cronbach and Gleser, 1957). Appendix A.2 derives the split-sample prediction.

Measurement coverage in Figure 4. Each domain's off-policy series contains four estimators at six states (seeds 0–2; steps 0 and 400). Estimators share the same responses; at step 0 their points nearly coincide. Success-rate selection's 40 rows reuse ten caches, so repeated points coincide.

A.2 Proofs and Measurement Budgets

Reliability and latent gain. Under the independent additive-noise model of Section A, the signal fraction equals the correlation between repeated measurements:

$$\rho = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_\varepsilon^2} = \mathrm{corr}(y_i, y'_i). \tag{7}$$

For a size-k rule $S(y)$, define

$$G(S) = \mathbb{E}\left[\frac{1}{k} \sum_{i \in S(y)} \theta_i\right] - \mu, \qquad \mu = \mathbb{E}[\theta_i]. \tag{8}$$

The expectation includes the latent scores, measurement errors, and any independent randomization of the selection rule.

Conditioning and randomized selection. Under Proposition 1, let $\sigma_y^2 = \sigma_\theta^2 + \sigma_\varepsilon^2$. The score coordinates are independent. Gaussian conditioning gives $\mathbb{E}[\theta_i \mid y] = \mu + \rho(y_i - \mu)$. A randomized selection rule may also use auxiliary randomness independent of the latent scores and measurement errors. Conditioning on that randomness does not change this expectation. For any observed score vector, selecting its top-k entries maximizes the size-k sum. Taking expectations therefore gives

$$G(S) \leq \rho \sigma_y c_{k,n} = \sqrt{\rho} \sigma_\theta c_{k,n}.$$

The oracle obtains $\sigma_\theta c_{k,n}$ by selecting the latent scores. This also covers zero noise, for which $\rho = 1$.

Expected gain across candidate pools. Proposition 1 averages over the latent scores and the errors, $\mathbb{E}_{\theta,\varepsilon}$; "finite pool" means that n is finite. It does not give the same ratio for every fixed pool. For example, let $n = 2$, $k = 1$, fixed latent scores $\theta = (-1, +1)$, and independent $\mathcal{N}(0, 1)$ errors. For a uniformly chosen prompt, the correlation between repeated measurements is $\rho = 1/2$. The oracle gain is 1. Noisy top-1 selection chooses $+1$ with probability $\Phi(\sqrt{2})$, giving gain $2\Phi(\sqrt{2})-1 \approx 0.843$ rather than $\sqrt{\rho} \approx 0.707$. This example fixes θ, whereas the proposition samples it from a Gaussian distribution.

Independently evaluated score gain. We evaluate selected prompts on independent samples. For reference scores s_i^{ref}, define

$$U(S) = \frac{1}{k}\sum_{i\in S} s_i^{\mathrm{ref}}, \qquad G_{\mathrm{ref}}(S) = U(S) - \frac{1}{n}\sum_{i=1}^{n} s_i^{\mathrm{ref}}. \tag{9}$$

Score units. For $a > 0$, replacing every score by $as + b$ leaves TopK, ρ, and hence the selected training data and its reward unchanged, while multiplying G and G_{ref} by a. Absolute score gains can be compared across scorers only when the target and units match. Within one scorer, G_{ref} is the independently evaluated score gain defined in Equation (9).

The second measurement. For $y'_i = \theta_i + \varepsilon'_i$ with independent errors, $\mathbb{E}[y'_i \mid y] = \mu + \rho(y_i - \mu)$. Selecting on y and evaluating on y' therefore gives expected score gain $\rho\sigma_y c_{k,n}$:

$$\frac{\mathbb{E}[k^{-1}\sum_{i\in \mathrm{TopK}(y)} y'_i] - \mu}{\sigma_y} = \rho c_{k,n}. \tag{10}$$

The latent-gain ratio and standardized split-sample score gain divide by different quantities, so their factors differ. Replacing the population standard deviation with a within-pool estimate gives an approximation for a finite pool. The model experiments use this sample normalization and average both selection–evaluation directions. Figure 4 substitutes the sample correlation $\widehat{r}$, which can be negative, for ρ.

Top-k overlap and margins. As reliability approaches zero, the two rankings under the Gaussian model approach independent uniform top-k sets. Their expected overlap fraction is k/n: each of the k selected indices has probability k/n of appearing in the other set. This overlap depends on membership, whereas latent gain also depends on score differences. More directly, suppose exact scores have boundary gap $\Delta_k = \theta_{(k)} - \theta_{(k+1)} > 0$ and their estimates have maximum error e, with latent scores ordered from largest to smallest. Every selected score remains above every excluded score when $\Delta_k > 2e$, since the smallest remaining gap is $\Delta_k - 2e$. Near ties can therefore change membership under small errors.

Averaging and the deployed budget. For the average of K independent, equal-variance measurements of the same latent target, reliability is

$$\rho_K = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_\varepsilon^2/K} = \frac{K\rho_1}{1 + (K-1)\rho_1}.$$

This is the Spearman–Brown relation (Spearman, 1910; Brown, 1910). The relation assumes independent additive errors before averaging. The implemented success-rate score first averages binary rewards and then folds the result around $1/2$; the cosine score also normalizes vectors. We therefore report empirical half-budget correlations for these scores instead of applying the formula automatically.

Nonlinear success-rate targets. With $\widehat{p}_{i,K} \sim \mathrm{Binomial}(K, p_i)/K$, the K-response success-rate score has target $\theta_{i,K} = \mathbb{E}[-|\widehat{p}_{i,K} - 1/2|]$, which is not the population value $-|p_i - 1/2|$. At $p_i = 1/2$, $\theta_{i,4} = -0.1875$ and $\theta_{i,8} = -0.13671875$, whereas $-|p_i - 1/2| = 0$. Four-response and eight-response measurements thus estimate different targets, rather than the same target with different noise levels. A half-budget correlation measures repeatability at that half budget.

Net-gain decomposition. Adding and subtracting $J_{\mathrm{rand}}(B - C_{\mathrm{tot},m})$ in Equation (3) gives

$$\Delta_{\mathrm{net}}(B) = \underbrace{J_m(B - C_{\mathrm{tot},m}) - J_{\mathrm{rand}}(B - C_{\mathrm{tot},m})}_{\text{selection benefit at matched training compute}} - \underbrace{\big[J_{\mathrm{rand}}(B) - J_{\mathrm{rand}}(B - C_{\mathrm{tot},m})\big]}_{\text{reward from spending the scoring budget on random training}}. \tag{11}$$

This identity requires neither monotone learning curves nor verifiable rewards.

B Off-Policy Gradient Estimators and Counterexamples

Stored-response scoring removes the need for new candidate rollouts, but must account for the policy that generated the responses. This section shows what can remain unidentified when importance correction omits either earlier or later tokens. The result concerns the exact selected set, not the size of the training benefit or the performance of SR-GC switching.

Population estimators. For a stored trajectory $Y = (a_1, \ldots, a_T)$, let h_t be its history, $r_t = \pi(a_t \mid h_t)/\beta(a_t \mid h_t)$ the token-level ratio, and $z_t = \nabla \log \pi(a_t \mid h_t)$. Write $P_t = \prod_{j<t} r_j$ and $S_t = \prod_{j>t} r_j$. With terminal reward R, the four population estimators are

$$g_{ab} = \sum_{t=1}^{T} \mathbb{E}_\beta[P_t^a r_t S_t^b R z_t], \qquad a, b \in \{0, 1\}. \tag{12}$$

The first index includes correction for earlier tokens and the second for later tokens. Under common support, finite expectations, and exact unclipped ratios, $g_{11} = g_\pi$. Our implementation clips the completed weights to $[0.1, 10]$. The measurement matrix records all four corrections; the training comparison uses g_{11}.

For the population estimators in Equation (12), take a unit direction u and define $s_c(x) = u^\top g_c(x)$, $c \in \{10, 01\}$. Let $\mathcal{I}_c^{(N)}$ contain N independent behavior trajectories per prompt, the nonzero directional terms retained by s_c, their retained importance weights, and exact policy KL. A rule measurable with respect to this vector can use only that information. It cannot use the omitted ratios or newly sampled current-policy outcomes. Normalized effective sample size is $(\sum_j w_j)^2/(N \sum_j w_j^2)$. This observation set defines the partially corrected estimator being studied.

Theorem 1 (Impossibility after correcting only one component)**.** *Fix $c \in \{10, 01\}$, $N \geq 1$, $0 < k < n$, and $\zeta > 0$. There exist two pools W_0, W_1 with binary rewards whose scores $s_\pi(x) = u^\top g_\pi(x)$ induce different true top-k sets $S_0^\star, S_1^\star$, but*

$$\mathcal{L}_{W_0}(\mathcal{I}_c^{(N)}) = \mathcal{L}_{W_1}(\mathcal{I}_c^{(N)}), \qquad \max_{j \in \{0,1\}} \max_{x \in W_j} D_{\mathrm{KL}}(\pi_{j,x} \| \beta_x) < \zeta.$$

Hence the minimax risk for selection rules based only on estimate c obeys

$$\inf_{\substack{\widehat{S}:\, \mathcal{I}_c^{(N)}\text{-measurable} \\ |\widehat{S}| = k}} \max_{j \in \{0,1\}} \Pr_{W_j}(\widehat{S} \neq S_j^\star) \geq \frac{1}{2}.$$

The retained weights have unit normalized ESS. In the two-candidate top-1 construction, the prompt ranked first by s_c has a true directional gradient with the opposite sign. The counterexamples remain valid after the group normalization used in GRPO (Shao et al., 2024) for every group size $K \geq 2$.

Action-value construction (reverses g_{00}, g_{10}). This construction uses a two-token Markov decision process. The first action is $A_1 \sim \mathrm{Bernoulli}(q)$, with q=1/2 under both policies. The reward is $R = A_2$. The success probabilities under the behavior policy are $Q^\beta(A_1{=}0) = \frac{1}{2} - \epsilon$ and $Q^\beta(A_1{=}1) = \frac{1}{2} + \epsilon$. The current policy swaps the two values. The current-policy gradient with respect to the logit of q is $g_\pi = -\epsilon/2$, but $g_{00} = g_{10} = +\epsilon/2$. The history distributions already match, so correcting earlier actions has no effect. Without correcting later actions, the estimate retains the behavior policy's action value and has the wrong sign. The mean KL divergence over later-token distributions is $O(\epsilon^2)$. The one-dimensional cosine between the estimated and current-policy gradients is -1.

History distribution construction (reverses g_{00}, g_{01}). Under the behavior policy, $P(A_1{=}0) = \frac{1}{2} + \epsilon$; under the current policy, this probability is $\frac{1}{2} - \epsilon$. The second action is $\mathrm{Bernoulli}(1/2)$ under both policies; $R = A_2$ if $A_1{=}0$ and $R = 1 - A_2$ otherwise. For the logit of the second action, $g_\pi = -\epsilon/2$ while $g_{00} = g_{01} = +\epsilon/2$. No tokens follow A_2, so correcting later actions has no effect. Without correcting earlier actions, the estimate retains the behavior policy's history distribution and has the wrong sign. The trajectory KL divergence is again $O(\epsilon^2)$.

In both cases, the Bernoulli parameter that differs between the policies has

$$D_{\mathrm{KL}}(\pi \| \beta) = 2\epsilon \log \frac{1/2 + \epsilon}{1/2 - \epsilon},$$

which is finite for every $0 < \epsilon < 1/2$ and has the expansion $8\epsilon^2 + O(\epsilon^4)$.

Indistinguishable pools. We first construct two pools that produce the same observed information for $c = 10$ but different true rankings. The $c = 01$ construction follows below. At the start of each trajectory, an exogenous branch $B \in \{\mathrm{visible}, \mathrm{hidden}\}$ is sampled, with hidden-branch probability h. In both branches, $A_1 \sim \mathrm{Bernoulli}(1/2)$, and direction u controls only the logit of A_1. On the visible branch of prompt i, the behavior and current policies share

$$P(A_2 = 1 \mid A_1 = 1) = \tfrac{1}{2} + \nu_i, \qquad P(A_2 = 1 \mid A_1 = 0) = \tfrac{1}{2} - \nu_i.$$

On the hidden branch, the behavior policy uses $P_\beta(A_2 = 1 \mid A_1) = 1/2$, while the current policy uses a sign $\xi_i \in \{-1, +1\}$ and the visible-branch probabilities with ν_i replaced by $\xi_i\epsilon$. The reward is $R = A_2$. The retained importance weight for the earlier and current actions is one. Therefore,

$$u^\top g_{10}(x_i) = \frac{(1-h)\nu_i}{2}, \qquad u^\top g_\pi(x_i) = \frac{(1-h)\nu_i + h\xi_i\epsilon}{2}.$$

Choose $0 < \nu_2 < \nu_1$ with $(1-h)\nu_1 < h\epsilon$. World W_0 takes $(\xi_1, \xi_2) = (+1, -1)$ and W_1 takes $(-1, +1)$. The partially corrected estimate ranks $x_1 > x_2$ in both worlds. The true top-1 prompt changes between the worlds. In W_1, the selected prompt x_1 also has a negative true directional gradient.

Exact-set identification versus selection-gain loss. The minimax risk above is the probability of missing the exact top-k set. It is not a lower bound on score or reward loss. For example, with $h = 1/2$, $\nu_1 = \epsilon/2$ and $\nu_2 = \epsilon/4$, the two candidates' true directional scores in W_1 are $-\epsilon/8$ and $5\epsilon/16$, a margin of $7\epsilon/16$. This margin approaches zero with ϵ, although the identification error bound remains $1/2$. Making KL small also makes the score margins and gradient magnitudes small in this construction. Partial-correction statistics therefore cannot identify the exact ranking even with exact KL and unit ESS, but the loss from an incorrect ranking can be small. This distinction between exact-set identification and score gain also appears in Appendix A.2.

To verify indistinguishability, note that the behavior distribution, retained weight, and every retained directional term are identical across worlds. By Bernoulli symmetry, the current policy's distribution of A_2 on the hidden branch has the same KL divergence for signs $+1$ and -1. Exact policy KL therefore does not distinguish the worlds. The divergence is $O(h\epsilon^2)$ and can be made smaller than any ζ. Thus, $\mathcal{I}_{10}^{(N)}$ has the same distribution in both worlds for every N.

Suppose a randomized selection rule chooses x_1 with probability p. Its error probabilities in W_0 and W_1 are $1 - p$ and p, so at least one is $1/2$ or larger. The same ambiguity extends to any top-k problem. Add $k - 1$ prompts, identical in both worlds, with visible-branch parameter $\nu_1 < \nu < 1/2$ and hidden-branch sign $+1$. They rank above both original candidates under s_π and s_c in both worlds. Add $n - k - 1$ further prompts with $\nu = 0$ and hidden-branch sign -1; they rank below both original candidates under both scores. The added prompts have the same observation laws in both worlds. The unresolved choice is therefore at the selection boundary. The sign reversal also concerns that boundary candidate, not the highest-ranked added prompt, whose sign is positive.

For $c = 01$, the missing information instead concerns the earlier action. Let the direction control only the logit of the second action. Use $A_2 \sim \mathrm{Bernoulli}(1/2)$ under both policies, and set $R = A_2$ after $A_1 = 0$ and $R = 1 - A_2$ after $A_1 = 1$. On the visible branch, both policies have $P(A_1 = 0) = 1/2 + \nu_i$. On the hidden branch, the behavior probability is $1/2$ and the current-policy probability

is $1/2 + \xi_i\epsilon$, both for the event $A_1 = 0$. The retained importance weight for the current and later actions is one. Consequently,

$$u^\top g_{01}(x_i) = \frac{(1-h)\nu_i}{2}, \qquad u^\top g_\pi(x_i) = \frac{(1-h)\nu_i + h\xi_i\epsilon}{2}.$$

Using the same ν_i and signs gives the same indistinguishability and ranking reversal, proving the $c = 01$ case.

Lemma 1 (Binary group normalization)**.** *Let $R \in \{0,1\}$ satisfy $P(R=1) = 1/2$, let z be a scalar directional score with $\mathbb{E}z = 0$, and define $g = \mathbb{E}[Rz]$. For $K \geq 2$ i.i.d. rollouts, standardize rewards as $\hat{A}_j = (R_j - \bar{R})/(\sqrt{\bar{R}(1-\bar{R})} + \eta)$ with any fixed $\eta \geq 0$, assigning zero when all rewards agree, and let $g_{\text{group}} = K^{-1}\sum_{j=1}^{K} \hat{A}_j z_j$ be the group-averaged update. Then $\mathbb{E}[g_{\text{group}}] = C_{K,\eta} g$ for a constant $C_{K,\eta} > 0$ depending only on K and η.*

Proof. Let M be the number of successes and $\Delta_z = \mathbb{E}[z \mid R = 1] - \mathbb{E}[z \mid R = 0]$. Since $P(R=1) = 1/2$ and $\mathbb{E}z = 0$, $g = \Delta_z/4$. Set $m = M/K$. Conditional on M, the expected group update is

$$\frac{m(1-m)}{\sqrt{m(1-m)} + \eta}\,\Delta_z.$$

Consequently,

$$\mathbb{E}[g_{\text{group}}] = 4\,\mathbb{E}\left[\frac{m(1-m)}{\sqrt{m(1-m)} + \eta}\right] g = C_{K,\eta} g.$$

For $\eta = 0$, the fraction is defined as zero at $m = 0$ and $m = 1$, consistent with the homogeneous-group convention. The expectation is strictly positive for every $K \geq 2$ because $P(0 < M < K) > 0$. Using the sample standard deviation instead of the population standard deviation changes only this positive constant. Every counterexample branch has success probability $1/2$. Group normalization therefore preserves the gradient signs and strict ordering of prompts. □

C Experimental Protocols and Accounting

Table 3: Training experiments and cost treatment. Per-state outcomes and coverage appear in Appendix D.

Comparison	Common condition	Outcome and scoring treatment
MATH learning	9 states; 100 further updates	Reward; scoring costs not matched
Single-check SR-GC	6 held-out starts; one strategy choice	Reward after training with the chosen strategy
MATH fixed budget	6 held-out states; 29,040 GPU-s	Reward; selection, setup and training share the allocation
Fixed-time switching	Shared branch state; longer training	Reward curves and observed target costs; scoring charged
Online SR-GC switching	Seeds 3/4; common step-25 prefix and total step 275	Matched-update rewards and scoring/training costs
MBPP	28,380 GPU-s training allocation	Reward and separately charged on-policy scoring
RLOO control	9 states; 100 further updates	Reward under a changed continuation objective

Table 3 lists training experiments. Retrospective SR-GC diagnostics reuse nine completed MATH learning comparisons and are reported separately in Appendix F.1.

Model, data, and sampling. We use the pretrained `allenai/Olmo-3-1025-7B` checkpoint (Team Olmo, 2025) with MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024) and MBPP (Austin et al., 2021) candidate datasets. MATH uses 400 candidate and 100 validation prompts; MBPP uses 512 candidate and 100 validation prompts. Both use dataset split seed 0. Generation uses temperature 1, top-$p = 1$, no top-k truncation, and at most 2048 new tokens. Models

load in BF16 with eager attention. We use the released RL-Zero math and code prompts without a tokenizer chat template.

For MBPP, the full dataset configuration combines all published splits, removes exact duplicate prompt texts, and shuffles with split seed 0. The first 512 prompts form the candidate pool and the next 100 form ranking validation; this split seed is independent of the training seed. Evaluation excludes the union of candidate and validation prompts from all five source seeds after Unicode NFKC and whitespace normalization. The remaining normalized prompt keys are sorted, shuffled with seed 20260914, and sampled without replacement to obtain 300 questions. Thus MBPP reward is evaluated on a disjoint holdout from the combined pool, not on the official test split as a whole.

On-policy gradient estimation. Each online On-policy selection step draws 40 candidate prompts uniformly at random from the 400-prompt pool, generates eight fresh current-policy responses for each (320 rollouts) and ranks the prompts by gradient score. The four highest-scoring prompts are selected for training. The same scoring stage also computes current-policy gradients for the 40 SR comparison prompts. SR-GC uses the mean gradient of each 40-prompt set. The learner generates eight fresh responses per selected prompt (32 responses) for one training update.

The full-pool diagnostic generates 32 current-policy responses per candidate at a fixed checkpoint, using leave-one-out advantages in four-response subgroups. The ranking block contains 16 responses; its first eight define the selection score from split R. All 16 define the separate higher-budget ranking diagnostic. The A and B blocks each contain eight responses. The validation split assigns 50 prompts to R and 25 each to A and B, with eight responses in one leave-one-out group per validation prompt. For R, we average the two candidate subgroup gradients and the 50 validation-prompt gradients, then compute the cosine between these means. A and B use the same procedure with their own responses and validation prompts. The independent reference score is the arithmetic mean of the scalar A and B cosines. Each component score is therefore a cosine of mean gradients, not a mean of subgroup cosines. This cosine reference is used for the score diagnostics in Appendix A. The A/B SR-GC diagnostics use the inner-product form of Equation (5), rather than cosine scores. The fixed CountSketch map uses gradients from the final four dense model layers and final normalization. Identical-reward groups contribute zero.

Off-policy gradient estimation. Off-policy selection uses gradients prepared in advance from a behavior cache containing eight responses per candidate. To prepare them at a policy checkpoint, the scorer computes behavior log probabilities and then loads that checkpoint's policy to compute its log probabilities and gradients. Stored tokens and rewards are reused; the gradients are evaluated at the scoring checkpoint rather than copied from the behavior policy. The validation direction is prepared under the same checkpoint policy using responses from that policy. At $d = 0$, $\beta = \pi$; the stored-response arm becomes off-policy as training changes π. Each completed token, prefix, suffix, or full-trajectory weight is clipped to $[0.1, 10]$. The leave-one-out advantage is

$$\widehat{A}_j = R_j - \frac{1}{K-1} \sum_{\ell \neq j} R_\ell.$$

Prompt gradients average over responses and sum response-token terms. The reference for off-policy scoring comes from split R only. Off-policy scoring uses one eight-response advantage group; the primary on-policy score averages two four-response subgroups. Both use eight responses, but their advantage groups differ. For the off-policy score diagnostic, each half recomputes leave-one-out advantages within four responses and uses the same 50-prompt R validation direction. Success-rate selection uses four rewards per half and eight for selection, without a validation gradient.

Training. The learner uses four distributed ranks, with one prompt and eight new responses per rank. It uses one optimization epoch, LoRA rank 16 and scaling 32, AdamW learning rate 10^{-5}, and maximum gradient norm 1. Sampling is indexed by seed, step, and rank. GRPO standardizes rewards within each response group; score estimation instead uses leave-one-out advantages. The GRPO loss averages over each response's tokens, whereas scoring sums token contributions. A continuation from a trained checkpoint resumes both its adapter and optimizer. LoRA targets query and value projections with zero dropout. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and zero weight decay. For GRPO, the reward-standard-deviation stabilizer is 10^{-4}, the reference-KL coefficient is zero, and the implemented ratio clip range is $[0.8, 1.2]$. With one epoch, the loss is evaluated only

before the update. The configured clip range therefore does not bound the probability ratio after the update.

RLOO learning structure. RLOO (Ahmadian et al., 2024) uses sequence-level REINFORCE with the mean reward of other responses as a baseline. Under independent on-policy sampling, excluding the response itself preserves the expected policy gradient. Our verifier-based adaptation uses eight fresh responses and zero reference-KL coefficient. It shares GRPO's architecture, LoRA parameters, verifier and response count; neither learner trains a critic.

For one prompt x, let $\bar{R} = K^{-1}\sum_j R_j$ and let ℓ_j be the number of response tokens in y_j. Our unclipped RLOO loss is

$$\widehat{A}_j^{\mathrm{RLOO}} = R_j - \frac{1}{K-1}\sum_{h\neq j} R_h = \frac{K}{K-1}(R_j - \bar{R}),$$

$$\mathcal{L}_{\mathrm{RLOO}} = -\frac{1}{K}\sum_{j=1}^{K} \widehat{A}_j^{\mathrm{RLOO}} \sum_{u=1}^{\ell_j} \log \pi(y_{j,u} \mid x, y_{j,<u}),$$

where advantages are held constant when differentiating with respect to policy parameters. GRPO instead uses $\widehat{A}_j^{\mathrm{GRPO}} = (R_j - \bar{R})/(\sigma_R + \eta)$, where σ_R is the population standard deviation within the group and $\eta = 10^{-4}$, and averages each response's token terms. At the pre-update policy, its ratios equal one. Each token's RLOO gradient coefficient is then $\frac{K}{K-1}(\sigma_R + \eta)\ell_j$ times its GRPO counterpart. Thus changing only mean-centering to leave-one-out centering contributes the constant $8/7$, whereas removing the two normalizers changes weights across prompts and response lengths, as analyzed by Liu et al. (2025). This identity compares raw gradients at the same policy, before gradient clipping and AdamW; it is not a multiplier for the resulting optimizer step.

Selection tie handling. Scores sort in descending order, with a seeded random secondary key breaking exact ties. The continuation subset builder uses seed $s + 1000$ for each selector, whereas its random arm assigns independent uniform keys using seed $s + 424243$; s is the run seed. These training subsets differ from the 20 randomized tie resolutions in the score-measurement matrix (ten per half, with independently seeded streams). The split-sample calibration in Figure 4 instead uses a stable descending score sort, preserving numeric prompt-ID order for exact ties, and standardizes evaluation scores with sample standard deviation (denominator $n - 1$). These tie-handling choices matter for discrete success-rate scores. The MBPP off-policy extension uses all 512 candidates and selects 51 per half, at steps 0 and 400 for seeds 0–2. It reuses the eight stored responses, split into two halves of four, for each of the four correction estimators.

Evaluation units. We normalize question text and whitespace before excluding candidate and ranking-validation questions from the 300-question MATH-train evaluation. The public GSM8K and MATH subsamples are fixed within each seed; they may differ across seeds. Bootstrap resamples whole questions, preserving paired comparisons; the public macro weights each set by $1/5$. Reward averages response-level correctness rather than whether any response succeeds. Endpoint rewards use eight responses per question; MBPP intermediate evaluations use four. For endpoints on a test set Q with binary rewards R_{qj},

$$\widehat{J} = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{8}\sum_{j=1}^{8} R_{qj}. \tag{13}$$

Reward verification. MATH continuation training and evaluation use `math-verify` 0.9.0; MBPP uses execution-based verification. Cached MATH responses for seeds 0–3 were rescored at all four checkpoints. Seed 4 uses the available initial-policy reward cache for SR selection. The policies that generated the stored responses were trained with a different verifier configuration; rescoring updates the stored rewards while retaining those policies.

Compute accounting. Table 3 specifies charged phases. Matched-update training multiplies total update wall time by four GPUs. Public evaluation sums its separately timed shards. In the score-measurement matrix, scoring jointly computes several estimators and reference splits, so that combined runtime does not give separate costs for individual selectors.

C.1 Accounting and Comparison Conditions

Incremental costs. For seed s, the selection charge in GPU-hours is

$$C_{\mathrm{sel},s} = \frac{1}{3600} \sum_{j \in \mathcal{S}_s} c_j, \qquad \overline{C}_{\mathrm{sel}} = \frac{1}{N} \sum_{s=1}^{N} C_{\mathrm{sel},s},$$

where c_j is the measured GPU-seconds for each nonoverlapping selection operation in $\mathcal{S}_s$. The reported value sums all such charges within the comparison interval. Shared prefixes and pre-existing caches are excluded; new scoring, subset preparation and metered retries are charged. Blank costs are unmeasured.

Comparison intervals. Table 7 stops at a fixed 29,040 GPU-second allocation; Table 11 ends after 250 post-prefix updates. Training in Table 7 includes startup and checkpointing; Table 11 excludes checkpointing.

Cross-objective inputs. Starting mean rewards agree at all nine evaluated states: 23.375, 23.750 and 22.583% at step 0, 25.417, 25.792 and 26.708% at step 100, and 29.917, 31.500 and 30.375% at step 400.

Table 4: Training and evaluation settings for the GRPO–RLOO comparison.

Item	Setting
Starting model	Pretrained base at $d = 0$; source GRPO-trained adapter at $d = 100, 400$.
Optimizer	Initialize at $d = 0$; resume the source optimizer at later checkpoints.
Selection	Same selection criteria and tie handling. On-policy recomputes selection gradients from fresh rollouts at every step; cached SR retains its reward-based subset.
Evaluation	300 questions, eight responses per question, with the common sampling settings and prompt format.
Verifier	MATH correctness verifier used for both training and evaluation.
Updates	100 additional updates per run.

Observed target attainment. For evaluated post-branch update indices $u_0 = 0, \ldots, u_M$, define $j_a^\star = \min\{j : \widehat{J}_a(u_j) \geq \tau\}$ and $u_a(\tau) = u_{j_a^\star}$. With shared prefix p, the cumulative step is $p + u_a(\tau)$. An empty set means no observed attainment. A target cost requires cumulative costs linked to that evaluated model and update index; reconstructed costs are marked separately. Evaluation is reported apart from the primary selection-and-training cost.

D Additional Training and Evaluation Results

Table 5 gives the individual MATH outcomes underlying the main-paper means; Table 6 shows variation in public macro gains across seeds.

Table 5: MATH-300 before and after training, and differences from random in percentage points.

d	Seed	Condition	Before (%)	After (%)	vs. random [95%]
0	0	Random	23.38	25.75	+0.00 [+0.00, +0.00]
0	0	Cached success rate	23.38	25.12	-0.62 [-2.29, +1.08]
0	0	On-policy alignment	23.38	27.12	+1.38 [-0.33, +3.12]
0	0	Off-policy alignment	23.38	26.04	+0.29 [-1.29, +1.88]
0	0	Correlation gate	23.38	25.04	-0.71 [-2.29, +0.92]
0	1	Random	23.75	25.75	+0.00 [+0.00, +0.00]
0	1	Cached success rate	23.75	25.75	+0.00 [-1.67, +1.71]
0	1	On-policy alignment	23.75	25.50	-0.25 [-1.92, +1.42]
0	1	Off-policy alignment	23.75	25.21	-0.54 [-2.12, +1.08]
0	1	Correlation gate	23.75	24.96	-0.79 [-2.33, +0.75]
0	2	Random	22.58	26.58	+0.00 [+0.00, +0.00]
0	2	Cached success rate	22.58	25.42	-1.17 [-2.88, +0.58]
0	2	On-policy alignment	22.58	27.21	+0.62 [-1.08, +2.33]
0	2	Off-policy alignment	22.58	25.08	-1.50 [-3.25, +0.21]
0	2	Correlation gate	22.58	25.62	-0.96 [-2.62, +0.71]
100	0	Random	25.42	27.79	+0.00 [+0.00, +0.00]
100	0	Cached success rate	25.42	28.46	+0.67 [-0.79, +2.12]
100	0	On-policy alignment	25.42	28.79	+1.00 [-0.62, +2.54]
100	0	Off-policy alignment	25.42	28.96	+1.17 [-0.67, +3.00]
100	1	Random	25.79	28.25	+0.00 [+0.00, +0.00]
100	1	Cached success rate	25.79	28.58	+0.33 [-1.46, +2.12]
100	1	On-policy alignment	25.79	29.25	+1.00 [-0.75, +2.83]
100	1	Off-policy alignment	25.79	29.25	+1.00 [-0.71, +2.75]
100	2	Random	26.71	26.50	+0.00 [+0.00, +0.00]
100	2	Cached success rate	26.71	28.96	+2.46 [+0.54, +4.42]
100	2	On-policy alignment	26.71	26.83	+0.33 [-1.42, +2.04]
100	2	Off-policy alignment	26.71	28.88	+2.38 [+0.50, +4.33]
400	0	Random	29.92	33.96	+0.00 [+0.00, +0.00]
400	0	Cached success rate	29.92	34.42	+0.46 [-1.42, +2.38]
400	0	On-policy alignment	29.92	32.83	-1.12 [-2.79, +0.58]
400	0	Off-policy alignment	29.92	34.17	+0.21 [-1.67, +2.08]
400	0	Correlation gate	29.92	32.75	-1.21 [-2.92, +0.54]
400	1	Random	31.50	33.42	+0.00 [+0.00, +0.00]
400	1	Cached success rate	31.50	35.21	+1.79 [-0.08, +3.62]
400	1	On-policy alignment	31.50	34.33	+0.92 [-0.71, +2.54]
400	1	Off-policy alignment	31.50	34.21	+0.79 [-0.88, +2.50]
400	1	Correlation gate	31.50	35.46	+2.04 [+0.33, +3.75]
400	2	Random	30.38	32.75	+0.00 [+0.00, +0.00]
400	2	Cached success rate	30.38	35.71	+2.96 [+1.25, +4.75]
400	2	On-policy alignment	30.38	33.33	+0.58 [-1.21, +2.33]
400	2	Off-policy alignment	30.38	34.21	+1.46 [-0.17, +3.12]
400	2	Correlation gate	30.38	32.75	+0.00 [-1.71, +1.71]

D.1 Selection-Inclusive MATH Results

The on-policy and cached-SR arms each have their own random controls. For each contrast, bootstrap resampling sorts question IDs numerically, resets NumPy PCG64 to seed 0, and takes sorted indices 250 and 9749 as interval endpoints.

On-policy alignment finishes below its random control in all nine development states, averaging -1.52 points. Cached SR improves on its random control in four of nine development states, averaging $+0.30$ points. Charged retries reduce the SR arm's remaining training time at s0/t25, s0/t50 and held-out s3/t25.

Table 6: Public five-benchmark macro gains over random by starting checkpoint and seed (percentage points). Each macro equally weights AIME 2024, AIME 2025, AMC 2023, GSM8K and MATH outside MATH-500. Seed means appear in Table 1.

d	Seed	On-policy	Cached SR	Off-policy	Correlation gate
0	0	+0.30	-1.02	-0.19	+0.20
0	1	+1.44	+0.66	+0.17	+0.82
0	2	+1.32	+1.40	+1.51	+1.10
400	0	+0.20	+1.67	+1.33	-0.29
400	1	-0.41	+0.28	-0.72	-0.05
400	2	+0.40	+0.62	+0.05	+0.53

Table 7: Costs, training updates, and MATH rewards at six held-out states. **Each selector is compared with its own separately executed random control.** Paired columns: selection S / its random R; $\Delta = S - R$ in percentage points, with paired 95% question-bootstrap intervals. Train updates counts completed optimizer updates after each state's starting checkpoint. Selection, Train, and Eval. refer to S; Total is selection plus training and excludes the shared prefix and cache creation. Selection is in GPU-seconds; other costs are GPU-hours. Each Mean averages the six states equally, including their training-update counts. Shading marks reward contrasts and means.

				Train updates			
State	Selection	Train	Total S/R	S/R	Reward S/R (%)	Δ vs. own R [95%]	Eval.
On-policy alignment: selection S versus its own random R							
s3/t25	24000.91	1.317	7.984 / 8.035	16 / 100	27.38 / 27.67	$\mathbf{-0.29}$ $[-2.25, +1.62]$	4.298
s3/t50	24363.37	1.255	8.023 / 8.040	16 / 100	25.58 / 28.83	$\mathbf{-3.25}$ $[-4.96, -1.54]$	4.113
s3/t100	24462.50	1.219	8.014 / 7.957	15 / 100	28.50 / 29.88	$\mathbf{-1.38}$ $[-3.00, +0.29]$	4.438
s4/t25	25418.71	0.932	7.992 / 7.982	11 / 98	25.50 / 28.21	$\mathbf{-2.71}$ $[-4.62, -0.75]$	4.394
s4/t50	24753.15	1.155	8.031 / 8.030	14 / 99	26.54 / 29.67	$\mathbf{-3.12}$ $[-4.83, -1.33]$	4.322
s4/t100	24714.76	1.164	8.030 / 7.974	14 / 99	29.38 / 29.08	$\mathbf{+0.29}$ $[-1.54, +2.12]$	4.518
Mean	**24618.90**	**1.174**	**8.012 / 8.003**	**14.3 / 99.3**	**27.15 / 28.89**	$\mathbf{-1.74}$	4.347
Cached success rate: selection S versus its own random R							
s3/t25	4.34	7.994	7.995 / 8.014	76 / 100	28.33 / 28.25	$\mathbf{+0.08}$ $[-1.62, +1.79]$	4.256
s3/t50	4.15	7.988	7.989 / 8.002	103 / 98	28.42 / 28.58	$\mathbf{-0.17}$ $[-1.83, +1.54]$	4.217
s3/t100	17.45	7.974	7.979 / 8.012	98 / 103	30.33 / 28.88	$\mathbf{+1.46}$ $[-0.29, +3.17]$	4.345
s4/t25	10.32	8.014	8.017 / 8.011	101 / 102	29.21 / 28.33	$\mathbf{+0.88}$ $[-0.96, +2.71]$	4.388
s4/t50	3.88	8.027	8.028 / 8.032	101 / 100	29.71 / 29.29	$\mathbf{+0.42}$ $[-1.46, +2.33]$	4.386
s4/t100	20.96	8.010	8.016 / 8.024	102 / 98	31.71 / 29.38	$\mathbf{+2.33}$ $[+0.50, +4.21]$	4.166
Mean	**10.18**	**8.001**	**8.004 / 8.016**	**96.8 / 100.2**	**29.62 / 28.78**	$\mathbf{+0.83}$	4.293

D.2 MBPP Learning and Compute Measurements

Reward comparisons. Table 8 gives the complete development and held-out endpoints. Across nine diagnosis-adjusted development pairs, on-policy selection improves by 1.26 percentage points on average. At six held-out states, full-allocation selection averages 33.58% versus 32.78% for random (a 0.79-point difference before rounding), with four positive differences, one zero and one negative; mean update counts are 89.83 and 87.17. The separate diagnosis-adjusted executions average 34.22% versus 32.65% (a 1.57-point difference). Reported paired intervals are available for eight development, four full-allocation held-out and four diagnosis-adjusted held-out pairs. These allocation conditions are not pooled.

Scoring inside the allocation. In a separate protocol, on-policy scoring is charged to the 28,380 GPU-second allocation. All 21 selected arms exhaust their allocation without a measured final reward, while their random controls have final evaluations. Recorded selection-time totals average 28,359.80 GPU-seconds (range 28,318.58–28,380.19).

Table 8: MBPP selection versus its paired random control with scoring charged separately. Cells give reward (%) / additional updates. Full uses a 28,380 GPU-second training allocation; Diag.-adj. subtracts the shared diagnostic charge. Differences are selection minus random (percentage points), with reported paired 95% question-bootstrap intervals. All endpoints are measured; – indicates that a paired interval is unavailable.

State	Allocation	On-policy	Random	Difference [95%]
s3/t25	Full	32.50 / 98	32.29 / 95	–
s3/t50	Full	32.92 / 97	32.92 / 71	$+0.00$ $[-1.42, +1.46]$
s3/t100	Full	35.12 / 98	34.08 / 95	$+1.04$ $[-0.46, +2.50]$
s4/t25	Full	32.75 / 73	31.25 / 95	–
s4/t50	Full	32.00 / 72	32.04 / 71	-0.04 $[-1.38, +1.29]$
s4/t100	Full	36.17 / 101	34.12 / 96	$+2.04$ $[+0.33, +3.75]$
s0/t25	Diag.-adj.	31.96 / 95	30.12 / 91	$+1.83$ $[+0.38, +3.29]$
s0/t50	Diag.-adj.	33.88 / 99	32.29 / 89	$+1.58$ $[+0.00, +3.12]$
s0/t100	Diag.-adj.	33.88 / 96	32.79 / 96	$+1.08$ $[-0.25, +2.42]$
s1/t25	Diag.-adj.	32.46 / 97	31.25 / 93	$+1.21$ $[-0.25, +2.67]$
s1/t50	Diag.-adj.	33.12 / 98	32.58 / 96	$+0.54$ $[-0.83, +1.92]$
s1/t100	Diag.-adj.	36.29 / 98	34.12 / 94	$+2.17$ $[+0.62, +3.67]$
s2/t25	Diag.-adj.	35.17 / 100	34.08 / 99	$+1.08$ $[-0.38, +2.54]$
s2/t50	Diag.-adj.	34.79 / 95	33.54 / 96	–
s2/t100	Diag.-adj.	38.04 / 97	37.42 / 101	$+0.62$ $[-0.67, +1.92]$
s3/t25	Diag.-adj.	32.75 / 100	30.71 / 71	$+2.04$ $[+0.42, +3.67]$
s3/t50	Diag.-adj.	32.21 / 97	32.54 / 72	-0.33 $[-1.92, +1.25]$
s3/t100	Diag.-adj.	36.75 / 97	34.04 / 94	$+2.71$ $[+1.33, +4.08]$
s4/t25	Diag.-adj.	34.21 / 96	31.88 / 95	$+2.33$ $[+0.96, +3.71]$
s4/t50	Diag.-adj.	33.46 / 96	32.83 / 94	–
s4/t100	Diag.-adj.	35.96 / 102	33.92 / 93	–

Table 9: MBPP scoring and training costs at the four held-out full-allocation states with paired final-reward and cost measurements. S uses on-policy alignment and R uniform sampling. All costs are GPU-hours. Phase sums exclude evaluation, other setup, diagnosis and shared initial training, and include failed attempts within the metered phases. Means weight these four states equally; each seed contributes two states.

State	Scoring S	Training S	Training R	Score+train S	Score+train R
s3/t50	9.059	7.761	7.763	16.820	7.763
s3/t100	8.930	7.807	7.803	16.737	7.803
s4/t50	8.994	7.784	7.793	16.778	7.793
s4/t100	8.618	7.806	7.779	16.425	7.779
Mean	8.900	7.790	7.785	16.690	7.785

Cached success-rate control. With cached-SR scoring included in the allocation, s0/t25 selection and its paired random control score 32.08% and 32.00% after 96 and 98 updates. The difference is $+0.08$ points, with paired interval $[-1.46, +1.58]$; selected-arm scoring costs 1.693 GPU-seconds. At s0/t50, selection scores 33.46% after 100 updates, without a corresponding random endpoint. These controls are separate from the on-policy protocol's random arms.

D.3 Objective Control and Direct Selector Measurements

The control comprises 27 training runs and nine starting-policy evaluations; Table 4 gives the settings.

Cross-objective outcomes. At $d = 400$, seed 0, random selection starts at 29.92% reward and ends at 33.96% under GRPO versus 47.79% under RLOO, a 13.83-point difference after 100 further updates. The corresponding cross-objective differences are 13.71 points for cached selection and 13.88 for on-policy alignment. All three selection strategies show similar reward increases under RLOO at this state. At $d = 100$, seed 0, the same comparison is not uniform: random and cached

selection end 7.00 and 7.96 points above their GRPO counterparts, whereas on-policy alignment ends only 2.29 points above. Appendix C derives how the objectives reweight the same response gradients.

Table 10: Within-seed RLOO reward differences (pp), with paired 95% question-bootstrap intervals conditional on the trained policies. Here On-policy, SR, and rand denote on-policy alignment, cached success-rate selection, and random selection. Each comparison uses complete evaluations from the same starting state.

d	Seed	On-policy − rand [95%]	SR − rand [95%]	SR − On-policy [95%]
0	0	-0.62 $[-2.21, +0.96]$	-2.58 $[-4.25, -0.92]$	-1.96 $[-3.58, -0.33]$
0	1	$+0.50$ $[-1.25, +2.29]$	$+1.46$ $[-0.29, +3.12]$	$+0.96$ $[-0.79, +2.71]$
0	2	$+2.54$ $[+0.83, +4.29]$	$+4.08$ $[+2.29, +5.88]$	$+1.54$ $[-0.12, +3.17]$
100	0	-3.71 $[-5.75, -1.67]$	$+1.62$ $[-0.29, +3.54]$	$+5.33$ $[+3.25, +7.46]$
100	1	$+3.46$ $[+1.42, +5.50]$	$+5.42$ $[+3.50, +7.42]$	$+1.96$ $[+0.08, +3.83]$
100	2	$+5.17$ $[+3.12, +7.21]$	$+6.54$ $[+4.42, +8.75]$	$+1.38$ $[-0.46, +3.21]$
400	0	-1.08 $[-2.88, +0.71]$	$+0.33$ $[-1.58, +2.25]$	$+1.42$ $[-0.50, +3.33]$
400	1	$+4.83$ $[+2.67, +7.00]$	$+2.50$ $[+0.46, +4.54]$	-2.33 $[-4.29, -0.38]$
400	2	$+0.46$ $[-1.67, +2.42]$	$+4.96$ $[+2.79, +7.08]$	$+4.50$ $[+2.38, +6.67]$

D.4 Fixed-Time Switching and Target Costs

Table 11 gives the seed-level cost components behind Figure 3. The fixed-time switching experiments below examine additional starting checkpoints and the costs to a common reward target.

Seed	Arm	Reward (%)	Train (GPU-h)	SR prep (GPU-s)	Scoring (GPU-h)
3	Random	29.33	19.55	0	0
3	On-policy	32.12	19.61	0	6.37
3	SR	32.33	19.03	1.179	0
3	Switch	33.04	19.29	1.179	2.84
4	Random	31.08	20.81	0	0
4	On-policy	32.96	19.77	0	6.70
4	SR	35.79	19.37	1.283	0
4	Switch	37.00	18.95	1.283	2.40

Table 11: Individual-seed costs through total step 275, excluding the shared prefix and cache creation. Every arm completes the same 250 post-prefix updates. Training time varies with generated response lengths. Zero denotes a stage not performed by that method. Scoring includes selection-gradient computation, selection scores, rollout generation and stage overhead, summed whenever selection is performed. On-policy sums selection time throughout its continuation; Switch sums it only before the SR transition. Per selection step this averages 92–96 GPU-seconds along the On-policy continuations and 102–115 GPU-seconds over the pre-transition steps of Switch. Each On-policy selection step scores 40 On-policy candidates and 40 SR comparison prompts; the top four On-policy candidates are used for training. SR-GC reuses both 40-prompt gradient means from this scoring step; its arithmetic (means, difference and inner product) takes 18.32 ms per check, averaged over every D computation in these experiments. Identical SR preparation is charged once to SR and Switch.

Figure 5 retains all 18 development trajectories. Both branches end after 301–319 additional updates except at s1/t50: On-policy reaches 38.625% after 407 updates, while SR reaches 34.333% after 311. This unequal-length pair is excluded from the mean in Section 5.2. Its On-policy evaluation costs 29.13 GPU-hours, reported separately from training.

The 24 held-out runs each have 11 evaluations. Endpoints occur after 288–315 additional updates, except Random s4/t100, which reaches 31.583% after 156 updates; its evaluation costs 29.56 GPU-hours. The sole SR choice in the single-check experiment is s3/t50 (Table 12).

D.4.1 Observed Costs to a Common Target

At s0/t50, on-policy alignment first attains the 35% target after 285 additional updates and cached selection after 250; their recorded costs differ by 9.23 GPU-hours. The cached accounting includes reconstructed events. At s2/t100, the corresponding update counts are 310 and 285, with an 8.70

Table 12: Single-check SR-GC choices and final rewards (%) at six held-out starts (seeds 3/4, steps 25/50/100); the step-50/100 states are separately prepared. Initial D is the A/B-mean contrast measured once before each continuation (Section 4): $D < 0$ selects cached SR; $D \geq 0$ selects On-policy. Rewards are each run's final evaluation after 288–315 additional updates (Random s4/t100: 156). The step-25 On-policy, cached SR and Random runs are the controls of Figure 2, whose step-275 values are earlier evaluations of the same runs.

Start state	Initial D	SR-GC choice	SR-GC	On-policy	Cached SR	Random
s3/t25	+2.03	On-policy	33.58	32.71	33.75	31.58
s3/t50	−1.21	SR	34.58	38.62	37.38	36.50
s3/t100	+16.15	On-policy	35.92	35.54	37.00	33.67
s4/t25	+1.62	On-policy	33.83	34.46	36.83	31.33
s4/t50	+1.43	On-policy	33.46	32.75	36.62	31.75
s4/t100	+5.27	On-policy	36.96	40.21	39.79	31.58

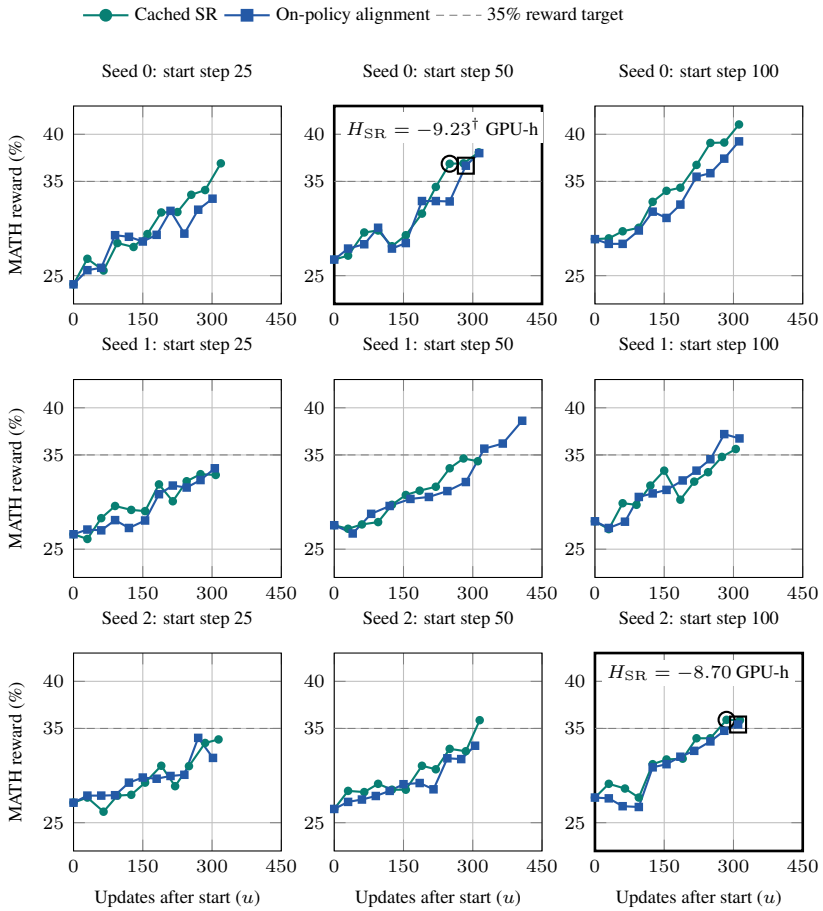


Figure 5: All 18 development learning trajectories after continued On-policy (blue squares) or a switch to cached SR (teal circles). Titles give the shared starting step t; the horizontal axis counts additional updates u. Each trajectory has 11 evaluations with eight responses per question. The dashed line is 35% reward. Bold borders identify paired target-cost observations, and outlined points mark first observed attainment. Labels give the SR-minus-On-policy cost H_{SR} in GPU-hours; negative values favor SR. †: reconstructed cost.

GPU-hour difference favoring cached selection. Total On-policy selection costs are 6.68 and 6.49 GPU-hours at these two states, respectively. The cost advantage therefore combines reduced scoring

Table 13: Costs to 35% reward. H_{SR} is SR cost minus On-policy cost; negative values favor SR. NR: target not reached within observation; –: finite cost difference unmeasured. The dagger marks SR cost reconstructed from execution records.

State	On-policy (GPU-h)	Cached SR (GPU-h)	H_{SR} (GPU-h)
s0/t25	NR	24.17	–
s0/t50	28.57	$19.34^{\dagger}$	$-9.23^{\dagger}$
s2/t100	30.65	21.95	-8.70

with fewer updates to the first observed qualifying evaluation. At s2/t100, this cost difference accompanies a final reward difference of only 0.46 points.

At s0/t25, cached selection attains 35% after 319 additional updates at 24.17 GPU-hours. On-policy alignment ends at 33.17% after 301 updates, with no qualifying evaluation. Its cost to the target and the corresponding H_{SR} remain unmeasured.

All 18 recorded starting rewards are below 35%. Eleven trajectories attain the target and seven have no observed attainment. The s1/t50 On-policy curve first reaches the target at 325 additional updates (35.67%), without a measured target cost. Five observed attainments have checkpoint-linked costs.

E Reliability-Based Decision Controls

Two applications of the same rule. This control measures score repeatability rather than the relative validation-gradient direction of On-policy and cached SR measured by SR-GC. This diagnostic control is applied at fixed checkpoints and in an executed pilot: ten uniform updates supply 40 score pairs, then a success-rate-selected or random subset supplies 90 updates. Equations (14) and (15) give the interval and action rule.

Correlation gate. Draw n_p pilot prompts uniformly and score each twice, giving pairs $(\widehat{s}_i^A, \widehat{s}_i^B)$. Let $\widehat{r}$ be their sample correlation. The lower end of the approximate two-sided $(1-\alpha)$ Fisher interval is

$$L = \tanh\left(\operatorname{atanh}(\widehat{r}) - \frac{z_{1-\alpha/2}}{\sqrt{n_p - 3}}\right). \tag{14}$$

Here z_q is the standard-normal q-quantile and $n_p > 3$. When both halves are valid measurements with positive variance, the rule is

$$A_m = \begin{cases} m, & L \geq r_{\min}, \\ \mathrm{rand}, & \text{otherwise}, \end{cases} \tag{15}$$

where rand denotes uniform random selection.

Pilot settings. We fixed $n_p = 40$, $\alpha = 0.10$, and $r_{\min} = 0.25$ before examining any $d = 0$ continuation reward. Thus $L \geq 0.25$ requires $\widehat{r} \geq 0.482$, not merely 0.25. The threshold measures score repeatability. Before the Fisher transform, the implementation clips the sample correlation to $[-0.999999, 0.999999]$ to avoid infinite transformed values. The cached-SR and uniform-random subsets are prepared before the pilot.

Reliability diagnostic assumptions. In the six fixed-checkpoint cases, success-rate selection's split-half correlations are 0.55–0.59, repeated at $d = 0, 400$ because the cache is shared. On-policy correlations range from -0.19 to 0.31 and off-policy correlations from -0.16 to 0.10.

The Fisher interval assumes independent bivariate-Gaussian pairs measured at one policy checkpoint and target. On-policy A/B scores vary both the candidate responses and validation directions, while shared validation samples induce dependence across prompts. Their correlation therefore measures the entire scoring procedure. The training pilot also changes the policy. The bound serves as a score-repeatability diagnostic under these sampling conditions.

Table 14: Independent on-policy alignment ranking overlap (fraction of shared selected prompts): five-seed means at each checkpoint against the independent-uniform reference k/n.

Pool	$d = 0$	$d = 25$	$d = 100$	$d = 400$	k/n
MATH	0.135	0.170	0.135	0.095	0.1000
MBPP	0.102	0.133	0.125	0.122	0.0996

Table 15: Executed training pilot: split-half score correlation $\widehat{r}$, decision, and MATH reward differences of the complete adaptive path against uniform sampling and unchanged success-rate selection (percentage points, with paired 95% question-bootstrap intervals conditional on the trained paths).

d	Seed	$\widehat{r}$	Decision	vs. random [95%]	vs. cached [95%]
0	0	0.28	Random	-0.71 [-2.29, +0.92]	-0.08 [-1.79, +1.58]
0	1	0.11	Random	-0.79 [-2.33, +0.75]	-0.79 [-2.38, +0.83]
0	2	0.64	Cached success rate	-0.96 [-2.62, +0.71]	+0.21 [-1.46, +1.83]
400	0	0.45	Random	-1.21 [-2.92, +0.54]	-1.67 [-3.62, +0.29]
400	1	0.64	Cached success rate	+2.04 [+0.33, +3.75]	+0.25 [-1.62, +2.08]
400	2	0.68	Cached success rate	+0.00 [-1.71, +1.71]	-2.96 [-4.71, -1.17]

Interpreting repeat measurements. Two measurements repeat the same target only when they share the scorer, the policy, the response budget, and the validation condition. The executed pilot therefore measures the repeatability of current-policy success-rate selection before the cached subsets selected using initial-policy rewards are used for training. Appendix A.2 explains how the nonlinear success-rate target depends on the response budget.

Charging diagnostic work to either action. Let B be the total budget, C_{diag} the charge of a single-checkpoint diagnostic, and C_{rem} the remaining scoring cost on continued selection beyond work already charged to the diagnostic. For an action $a \in \{m, \mathrm{rand}\}$, where m denotes the selector and rand uniform sampling,

$$C(a) = C_{\mathrm{diag}} + \mathbf{1}\{a = m\}C_{\mathrm{rem}}, \qquad T(a) = B - C(a).$$

Choosing random still costs C_{diag}, leaving $B - C_{\mathrm{diag}}$ for training. A training pilot changes the policy and optimizer; both actions resume from that state, with pilot updates charged once. For this diagnostic followed by continued selection, $C_{\mathrm{tot},m} = C_{\mathrm{diag}} + C_{\mathrm{rem}}$.

Later application at $d = 100$. The same pilot is evaluated at $d = 100$ for seeds 0–2 (Table 16). The gate retains cached SR at seeds 0 and 1 ($\widehat{r} = 0.60$ and 0.55) and chooses random at seed 2 (0.28). All three gate paths end below the fixed cached-SR subset at $d = 100$ (-1.00, -0.50 and -1.83 points); only the seed-2 interval excludes zero. Against the uniform-sampling control the differences are -0.33, -0.17 and $+0.62$ points. As in Table 15, the pilot adds diagnostic work without improving mean MATH reward over the unchanged cached subset.

Table 16: GRPO correlation-based selection at $d = 100$ on MATH-300. r_{half} is the split-half score correlation; the decision follows Equation (15). Controls train on fixed uniform or cached-SR subsets from the same starting state. Differences are final rewards minus control rewards in percentage points; brackets are paired 95% question-bootstrap intervals.

Seed	r_{half}	Chosen subset	vs. random [95%]	vs. cached SR [95%]
0	0.60	Cached SR	-0.33 $[-1.92, +1.25]$	-1.00 $[-2.58, +0.54]$
1	0.55	Cached SR	-0.17 $[-1.83, +1.54]$	-0.50 $[-2.12, +1.12]$
2	0.28	Random	$+0.62$ $[-1.12, +2.38]$	-1.83 $[-3.67, -0.08]$

Correlation-gate public-benchmark gains. The public five-set macro gains over random average $+0.71$ and $+0.06$ percentage points at $d = 0$ and $d = 400$, respectively, across three seeds. Per-seed public macro results are in Appendix D.

F SR-GC DIAGNOSTICS

F.1 RETROSPECTIVE DIAGNOSTICS OF COMPLETED COMPARISONS

Table 17 reports both reference contrasts, their mean D, the indicated selector and the observed reward difference for the nine completed On-policy/SR comparisons.

Reference averaging and pooling. Write $v_h = P\widehat{v}_{t,h}$ and δ_h for the projected selected-subset gradient difference in reference h. At this checkpoint, $D_t^{AB} = (\langle v_A, \delta_A\rangle + \langle v_B, \delta_B\rangle)/2$. Pooling the gradient means would give $D_{t,\text{pool}} = \langle (v_A + v_B)/2, (\delta_A + \delta_B)/2\rangle$, so

$$D_t^{AB} - D_{t,\text{pool}} = \tfrac{1}{4}\langle v_A - v_B, \delta_A - \delta_B\rangle.$$

Equal total response budgets therefore do not make these measured statistics identical.

Table 17: Retrospective diagnostics of nine completed On-policy/SR comparisons; no switches are executed. D is the mean of the experimental A/B gradient contrasts; Choice follows its sign at the starting checkpoint. Observed reward differences are On-policy minus SR after 100 updates; positive values favor On-policy.

Step	Seed	A	B	D	Choice	Reward diff. (pp)	Agreement
0	0	+6.41	+3.05	+4.73	On-policy	+2.00	Yes
0	1	-5.63	-8.54	-7.08	SR	-0.25	Yes
0	2	+7.13	+4.59	+5.86	On-policy	+1.79	Yes
100	0	-4.92	+9.00	+2.04	On-policy	+0.33	Yes
100	1	-14.26	-11.71	-12.99	SR	+0.67	No
100	2	-8.48	-5.14	-6.81	SR	-2.13	Yes
400	0	+6.83	-17.61	-5.39	SR	-1.58	Yes
400	1	+3.75	-4.86	-0.56	SR	-0.88	Yes
400	2	-2.57	+27.19	+12.31	On-policy	-2.38	No

Disagreement and signal magnitude. The two incorrect mean-sign recommendations have different reference patterns (Table 17). At step 100, seed 1, both references favor SR but the observed 100-update reward favors On-policy by 0.67 points; the paired On-policy-minus-SR interval $[-1.00, +2.42]$ includes zero. At step 400, seed 2, the reference signs disagree and the observed On-policy-minus-SR reward difference is -2.38 points, with interval $[-4.17, -0.63]$. Replacing projected dot products with mean cosine leaves both incorrect mean signs unchanged. The largest absolute single-prompt contributions, 1.77 and 2.21 respectively, are smaller than the absolute mean contrasts; removing only that contribution at fixed normalization cannot reverse either sign. Local scoring gradients and 100-update GRPO outcomes differ in objective, horizon and sampling, providing several possible sources of disagreement.

F.2 REPEATED-CHECKPOINT DIAGNOSTICS

Table 18 reports 11 A/B diagnostic checks per seed along On-policy continuations from step 25. A and B assess sampling variation at the same policy. In the A/B-mean series, seed 3 first has $D < 0$ at step 50, reverses to $+12.14$ at step 75, then is negative at steps 100 and 125. The three-check mean at step 100 is $+2.15$, so the temporal rule first signals at step 125. Seed 4's first consecutive negatives occur at steps 75 and 100, giving a signal at step 100.

Retrospective sensitivity analysis of the check histories. We apply the temporal rule separately to A, B and their means to assess reference sensitivity. Jackknife bounds characterize uncertainty in these diagnostic measurements.

Retrospective uncertainty analysis of the measured checks. Approximate one-sided, stratified leave-one-prompt-out jackknife upper bounds use $\alpha_j = 0.05/[j(j+1)]$ at check $j = 1, 2, \ldots$ along each trajectory. The bound is $U_j = D_j + \Phi^{-1}(1-\alpha_j)\,\text{SE}_{\text{jack}}(D_j)$, where Φ is the standard normal CDF and the standard error combines prompt-deletion variances for the candidate union and each

Seed / total step	25	50	75	100	125	150	175
Seed 3	+2.03	−3.56	+12.14	−2.14	−0.98	−18.30	−10.17
Seed 4	+1.62	+4.65	−4.62	−0.48	−7.42	−1.34	−3.11

Seed / total step	200	225	250	275
Seed 3	−4.64	+1.07	+8.65	+9.17
Seed 4	−3.01	−3.05	−11.08	−8.84

Table 18: A/B-mean SR-GC diagnostics along the On-policy continuations from step 25 through step 275. Columns give total training steps; negative D favors SR, nonnegative On-policy. The step-25 measurements are shared with Table 12; later columns follow these continuations, whereas that table's step-50/100 rows are separately prepared starting states.

Table 19: Reference sensitivity of temporal confirmation. Entries give the first qualifying step when the rule is applied to the A, B or A/B-mean diagnostic histories (Table 18).

Signal	Seed 3 step	Seed 4 step
A only	100	225
B only	150	75
A/B mean	125	100

validation reference. Candidate deletions keep A/B measurements paired. None of the 22 checkpoint contrasts has an upper bound below zero. At the first consecutive negative pair in each A/B-mean history, the successive upper bounds are seed 3 at steps 100/125 (+27.09/+22.97) and seed 4 at steps 75/100 (+23.92/+23.61). The bounds describe the sampling uncertainty of the A/B diagnostics.