BEYOND PLAN SELECTION: ATTRIBUTING AND IMPROVING EXECUTION IN RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Does reinforcement learning with verifiable rewards (RLVR) improve reasoning by selecting better plans or by executing them more reliably? We study this question through a matched plan-conditioned interface that allows Base and RLVR-trained selectors and executors to be interchanged while keeping the execution context fixed for each task–plan pair. Using a 1.5B model in two controlled two-plan environments, we separate realized performance contributions from available routing opportunity and the effects of targeted training. In arithmetic, the prespecified component-replacement analysis assigns 2.27 percentage points more improvement to execution than selection (95% CI: 1.14–3.41), although a common-repeat sensitivity analysis does not establish this ordering. Motivated by this diagnosis, we evaluate Execution-Focused RLVR (EF-RLVR), which removes the direct selector-policy loss while retaining shared parameters and KL regularization. EF-RLVR improves accuracy over Standard Joint RLVR by 1.28 points across eight fresh paired training seeds (95% CI: 0.77–1.79). A factorial follow-up and a supplementary direct comparison support a larger benefit from selector-term removal than executor up-weighting. In Search, input-dependent routing remains beneficial after training under both structured and final-answer scoring, but neither the original selector objective nor a uniform-action-weighting variant learns the conditional routing rule. These results show how component attribution can guide a useful training intervention, while demonstrating that available selection opportunity need not translate into learned routing.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) improves language-model performance on tasks with programmatically checkable outcomes (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025). Yet a higher success rate leaves an important question unanswered: did the model become better at choosing a solution plan, or at carrying out the plan it chose? These explanations suggest different interpretations of what RLVR has changed, but end-task accuracy alone cannot distinguish them.

Recent work studies RLVR through changes in reasoning-pattern frequencies, access to solution families, high-entropy decisions, and reasoning correctness (Chen et al., 2026; Wang et al., 2025; Zhou & Li, 2026; Wen et al., 2026). A shift toward a reasoning pattern does not by itself quantify that pattern's contribution to accuracy. Separating selection from execution requires comparing their effects while holding the other component fixed.

We study this distinction through an explicit plan interface. For each problem, a selector chooses one of two executable plans, and an executor receives a canonical context determined only by the task and the chosen plan. The selector transcript is discarded before execution, and the same plan-conditioned context is used whether the plan is selected naturally or assigned externally. This allows Base and RLVR-trained selectors and executors to be interchanged while holding the counterpart fixed. The resulting four-cell component-replacement design measures how end-task performance changes when replacing the selector, the executor, or both. The interface defines selection as choosing a plan identity and execution as generating the plan-conditioned response. In controlled arithmetic, component replacement attributes more of the realized RLVR gain to execution than to selection. This diagnosis motivates Execution-Focused RLVR (EF-RLVR), which improves Standard

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.

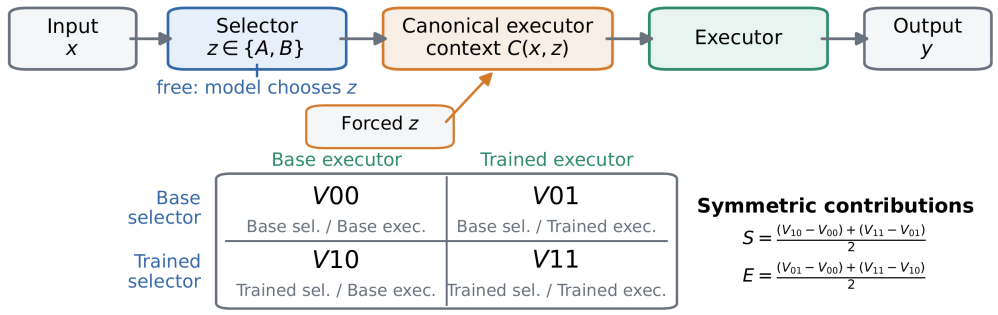


Figure 1: Component replacement under a matched plan interface. The selector chooses a plan identity; the executor receives a canonical context constructed from the task and that identity, without the selector transcript. Natural and forced assignment use the same executor context. Crossing Base and trained selectors and executors yields four systems, from which we compute symmetric replacement contributions and interaction.

Joint RLVR in an independent eight-pair confirmation. A subsequent factorial ablation attributes most of the improvement to removing the direct selector-policy term rather than increasing the executor coefficient. A separate prospective comparison also favors the specified execution-directed training procedure.

Search provides a complementary test: input-dependent routing remains valuable after training, but neither the original selector objective nor a prospectively frozen uniform-weighting variant learns the conditional rule. Together, our experiments connect measurable component attribution to a confirmed training intervention while distinguishing available routing opportunity from what a particular objective can exploit.

2 Related Work

RLVR and reasoning behavior. Outcome-based reinforcement learning improves verifiable reasoning performance in systems such as DeepSeekMath, DeepSeek-R1, and DAPO (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025). Recent work studies what changes under such training, including accessible reasoning capacity, reasoning-pattern selection, access to solution families, and reasoning correctness (Yue et al., 2025; Chen et al., 2026; Krishnamurthy et al., 2026; Zhou & Li, 2026; Wen et al., 2026). Most closely related, Zhou & Li (2026) distinguish changes at the entrance to a solution family from behavior after that family has been entered. Our intervention is operationally different: we assign explicit executable plans and replace the selector or executor while holding its counterpart fixed behind a matched execution interface. This measures the end-task effect of component replacement at a specified plan interface. Generated reasoning families, partial prefixes, and externally assigned executable plans define different interventions and partitions of behavior, so their attribution results need not coincide. Related routing work also studies learned skill selection and counterfactual expert routing in other settings (Li et al., 2026; Yoon et al., 2026). These systems differ from our setting in that our selector is an explicit two-plan functional role used for intervention and attribution rather than a learned architectural router.

Exploration and selective optimization. A related line of work studies how reinforcement learning redistributes probability and learning signal across reasoning trajectories. Work on entropy and token-level learning dynamics emphasizes concentration at high-entropy or low-probability decisions (Cui et al., 2025b; Wang et al., 2025; Xu et al., 2026; Huang et al., 2026a), while process supervision and implicit process rewards provide alternative ways to place learning signal within a trajectory (Cobbe et al., 2021; Lightman et al., 2023; Cui et al., 2025a). Other work studies exploration and probability-dependent reweighting of RLVR updates (Cheng et al., 2025; Huang et al., 2026b).

Our interventions operate at explicit plan-selection and plan-conditioned execution roles rather than at arbitrary token positions. EF-RLVR removes the direct selector-policy term while leaving the shared parameterization and reference-KL regularization intact. Uniform-S instead replaces the original old-policy action weighting with equal weighting over the two counterfactual selector actions. The Search experiments then test whether these selector objectives exploit routing opportunity measured independently through forced execution. Accordingly, the negative Uniform-S result should not be interpreted as a general result about entropy regularization, exploration methods, or low-probability-token reweighting.

Component attribution. Execution can remain difficult even when relevant knowledge or a plan is provided explicitly (Sinha et al., 2026), motivating separate measurement of plan choice and plan-conditioned success. Our four-cell allocation is related to the two-component Shapley value, which averages each component's marginal replacement effect over the two possible replacement orders (Shapley, 1953). Its relation to causal mediation is conceptual (Pearl, 2001; Imai et al., 2010): our estimand is defined by externally replacing observable selector and executor functions behind a canonical interface, rather than by identifying natural direct and indirect effects through an endogenous mediator. The methodological contribution is therefore not a new attribution algebra, but the matched intervention that makes selector–executor replacement measurable and connects the resulting attribution to prospective training interventions.

3 METHOD

3.1 A MATCHED PLAN-CONDITIONED INTERFACE

For a task $x \sim \mathcal{D}$, a selector chooses a plan $z \in \mathcal{Z} = \{A, B\}$ and an executor generates a response y from a canonical context $C(x, z)$. A checkpoint θ induces a selector $p_\theta(z \mid x)$ and an executor distribution $q_\theta(y \mid C(x, z))$. Let $R(x, z, y) \in \{0, 1\}$ denote the specified verifier. We define

$$e_\theta(z, x) = \mathbb{E}_{y \sim q_\theta(\cdot \mid C(x,z))}[R(x, z, y)], \tag{1}$$

$$V(p, e) = \mathbb{E}_{x \sim \mathcal{D}} \left[\sum_{z \in \mathcal{Z}} p(z \mid x) e(z, x) \right]. \tag{2}$$

Selection and execution are interface-defined roles implemented by the same model parameters.

Under free evaluation, the model selects z naturally. Under forced evaluation, the environment assigns z before execution. Both conditions use the same context constructor:

$$C_{\text{free}}(x, z) = C_{\text{forced}}(x, z) = C(x, z). \tag{3}$$

After selection, the selector transcript is discarded and the executor context is constructed only from (x, z). Thus forcing a plan changes its assignment rather than the plan-conditioned executor input.

In the production implementation, executor generation is invoked from the canonical plan-specific prompt, and selector-transcript tokens are not concatenated into the executor input. This provides an implementation-level interface guarantee; we do not claim that the archived audit exhaustively rules out every possible form of runtime hidden-state or cache dependence.

The execution quantity is therefore explicitly interventional and can be measured even for a plan that is rarely selected naturally.

This interface intentionally excludes information conveyed by arbitrary selector-generated rationale. Retaining such rationale would allow selection-produced text to mediate execution and would define a different estimand. Our decomposition therefore concerns plan-identity selection and plan-conditioned execution under the canonical interface, rather than an unrestricted decomposition of natural-language reasoning.

3.2 COMPONENT REPLACEMENT AND SYMMETRIC ATTRIBUTION

Let (p_0, e_0) denote the Base components and (p_1, e_1) the trained components. We evaluate

$$V_{ij} = V(p_i, e_j), \qquad i, j \in \{0, 1\}. \tag{4}$$

The diagonal cells V_{00} and V_{11} are the matched Base and trained systems, whereas V_{10} and V_{01} replace only the selector or executor. We symmetrically attribute the Base-to-trained performance difference as

$$S = \tfrac{1}{2}\left[(V_{10} - V_{00}) + (V_{11} - V_{01})\right], \tag{5}$$

$$E = \tfrac{1}{2}\left[(V_{01} - V_{00}) + (V_{11} - V_{10})\right]. \tag{6}$$

For two replaceable components, this symmetric allocation is equivalent to averaging each component's marginal replacement effect over the two possible replacement orders, i.e., the two-component Shapley allocation. These are the two-component Shapley contributions, obtained by averaging marginal effects over both replacement orders. The matched interface makes the required selector–executor combinations directly evaluable.

These satisfy

$$S + E = V_{11} - V_{00}, \qquad E - S = V_{01} - V_{10}. \tag{7}$$

Our primary relative attribution contrast is therefore $E - S$. We additionally report

$$I_{\text{swap}} = V_{11} - V_{10} - V_{01} + V_{00}, \tag{8}$$

where I_{swap} measures how the effect of replacing one component depends on the other. A nonzero component-replacement interaction does not invalidate the symmetric attribution; it means that the simple replacement effects are context-dependent. Accordingly, S and E are symmetric replacement contributions rather than independent additive mechanisms.

When free-selection and forced-execution measurements are available separately, we additionally reconstruct performance from separately measured selection probabilities and forced-execution outcomes as an auxiliary consistency check; the estimator and historical reconstruction results are given in Appendix C.2. Direct four-cell evaluation remains the primary attribution measurement.

3.3 Training Interventions

Attribution describes where a completed performance difference is expressed. Whether it motivates a useful optimization target is a separate empirical question.

Training-signal-placement procedures. We compare two procedures that place the direct policy-learning signal on different parts of the same shared-parameter model. We refer to these conditions as S-directed and E-directed training.

The formal arithmetic comparison uses eight semantic groups per outer step and eight executor rollouts per group, balanced as four Plan-A and four Plan-B rollouts. Each arm runs for 100 outer steps, yielding 6,400 sampled executor trajectories and 100 optimizer updates.

For S-directed training, the balanced rollouts estimate both plan-conditioned values, and both selector actions are enumerated and optimized with an exact two-action PPO objective under the frozen old-policy weighting. For E-directed training, the same balanced rollout design supplies executor trajectories optimized with group-local GRPO advantages. Full objective and baseline definitions are given in Appendix B.2.

Both procedures update the same shared model parameters. They match sampled-trajectory count, plan exposure, and optimizer-update count, but not gradient-bearing token count or optimization geometry. Accordingly, they compare two specified training procedures rather than token-equalized measurements of intrinsic selector and executor trainability.

Execution-Focused RLVR. We next define Execution-Focused RLVR (EF-RLVR). For Standard Joint RLVR and EF-RLVR, write the direct policy objective as

$$\mathcal{L}_{\lambda_S,\lambda_E} = \lambda_S \mathcal{L}_S^{\text{PG}} + \lambda_E \mathcal{L}_E^{\text{PG}} + \mathcal{K}, \tag{9}$$

where $\mathcal{K}$ denotes the unchanged reference-KL regularization.

Standard Joint RLVR uses $(\lambda_S, \lambda_E) = (0.5, 0.5)$, whereas EF-RLVR uses $(0, 1)$, with sampling, grouping, and reference-KL regularization unchanged. Shared-parameter updates can still change selector behavior. The initial comparison evaluates the complete configuration; the factorial follow-up in Section 4.3 separates its two coefficient changes.

3.4 Routing Opportunity and Statistical Scope

For a frozen input-only router $r(x)$ and a specified executor e, define

$$H_{\text{route}}(e) = \mathbb{E}_x[e(r(x), x)] - \max_{z \in \{A,B\}} \mathbb{E}_x[e(z, x)]. \tag{10}$$

A positive value shows that input-dependent routing improves over either fixed plan for the specified executor, input distribution, and outcome. It does not imply that an earlier RLVR gain was caused by selection or that a particular selector-training objective will realize that opportunity.

Hybrid attribution uses whole-task bootstrap intervals conditional on the evaluated checkpoints. Base Search routing audits resample matched task pairs and recompute the stronger fixed plan within each draw. Paired training comparisons use the training block as the inferential unit; the arithmetic post-outcome two-stage block-and-task bootstrap additionally resamples evaluation tasks within selected blocks. Prospectively frozen Search follow-up analyses and post-training endpoint summaries use the units specified in their respective analyses, while descriptive archive-recovery intervals are labeled separately.

4 Experiments

We ask three questions: which component accounts for the realized RLVR gain; whether that attribution motivates a useful training intervention; and whether measurable Base routing opportunity is accompanied by learned conditional routing under the tested selector objective.

4.1 Experimental Setting

All experiments use DeepSeek-R1-Distill-Qwen-1.5B. Prospective training arms initialize from the same pristine Base checkpoint. We use two controlled environments in which plans can be externally assigned and outcomes verified programmatically.

Arithmetic. Given integers (X, Y, Z, K), the model computes $(X + Y)Z - K$. Plan A returns

$$(X + Y,\ (X + Y)Z,\ (X + Y)Z - K),$$

whereas Plan B returns

$$(XZ,\ YZ,\ XZ + YZ,\ XZ + YZ - K).$$

The two plans produce the same final answer through different execution paths. The primary outcome requires a valid plan-specific JSON integer array whose entries exactly match all required intermediate values. Invalid or wrong-length outputs fail.

The primary attribution evaluation uses 1,024 tasks. The formal S/E training comparison uses two frozen, disjoint 400-task training splits, assigned to seven and six paired blocks, respectively, and the fixed 1,024-task evaluation set.

Search. The two executable algorithms are interpolation search (Plan A) and binary search (Plan B). Tasks pair affine- and quartic-spacing arrays with matched structural properties. The frozen input-only router chooses A for affine spacing and B for quartic spacing without accessing the target answer. Primary structured correctness requires both the answer and the prescribed execution certificate; a separate answer-only criterion is reported as a sensitivity outcome. Qualification, Base production-interface auditing, and formal training evaluation use explicitly distinguished task sets and evidence roles.

4.2 Arithmetic Attribution under the Prespecified Hybrid Evaluation

Direct component substitution yields

$$V_{00} = 57.25\%, \qquad V_{10} = 55.49\%, \qquad V_{01} = 57.76\%, \qquad V_{11} = 58.94\%.$$

The symmetric replacement contributions are

$$S = -0.293 \text{ pp}, \qquad E = +1.978 \text{ pp}.$$

The primary relative contrast is

$$E - S = +2.271 \text{ pp}, \qquad 95\% \text{ CI} = [+1.139, +3.410]. \tag{11}$$

The component-replacement interaction is also positive:

$$I_{\text{swap}} = +2.946 \text{ pp}, \qquad 95\% \text{ CI} = [+0.505, +5.347].$$

A post-outcome sensitivity analysis restricting all four cells to the common repetition indices 0–3 retains a positive execution-minus-selection point estimate, but its interval includes zero: $E - S = +1.611$ pp (95% CI $[-0.318, +3.564]$). The component-replacement interaction likewise attenuates to $+0.122$ pp ($[-2.612, +2.856]$), indicating sensitivity to repetition scope (Appendix D). The prespecified full-repeat analysis remains primary; the restricted analysis does not establish the relative contribution ordering.

Sensitivity to intermediate-node scoring. A post-outcome rescoring of the same frozen Hybrid generations retains the original plan-specific validity requirements but checks only final-answer correctness. The execution-minus-selection contribution remains positive: $E - S = +1.994$ pp, with a 95% CI of $[+0.863, +3.125]$.

Historical reconstruction. Two earlier RLVR runs show the same execution-tilted reconstruction pattern, but the reconstruction residual intervals do not establish equivalence; full estimates and confidence intervals are reported in Appendix C.2. We therefore treat reconstruction as auxiliary consistency evidence and direct Hybrid substitution as the primary attribution result.

4.3 Testing an Attribution-Motivated Training Intervention

Prospective training-signal-placement comparison. Across 13 paired arithmetic blocks, E-directed training reaches 59.685% free accuracy, compared with 56.536% for S-directed training, giving

$$\text{Acc}(E) - \text{Acc}(S) = +3.148 \text{ pp},$$

with 95% CI $[+2.854, +3.443]$. All 13 paired differences favor E.

A post-outcome two-stage block-and-task bootstrap gives a consistent 95% percentile interval of $[+2.695, +3.594]$ pp. Each bootstrap replicate first resamples the 13 paired training blocks and then independently resamples the 1,024 evaluation tasks within each selected block, using matched task draws for the S- and E-directed endpoints within that block. This sensitivity analysis remains conditional on the frozen training manifests and does not capture variation over newly generated training manifests.

Forced Plan-A and Plan-B contrasts show the same ordering, at $+3.573$ and $+3.154$ pp, respectively. Conversely, S-directed training has a 2.109 pp higher Plan-A selection rate, so the free-accuracy difference is not explained by a larger shift toward the dominant Plan A.

This procedure-level comparison follows the budget-matching scheme in Section 3.3. Gradient and KL diagnostics document additional differences in the realized optimization dynamics (Appendix E). These results therefore compare specified training procedures rather than intrinsic component trainability.

Independent EF-RLVR confirmation. After development, EF-RLVR was frozen for a separate eight-pair comparison against Standard Joint RLVR using fresh training seeds; the complete development history, including the earlier EA configuration, is reported in Appendix F.1.

EF-RLVR improves mean free accuracy by

$$+1.279 \text{ pp}, \qquad 95\% \text{ CI} = [+0.766, +1.791],$$

with all eight paired differences positive (exact two-sided sign-flip $p = 0.0078125$). Forced Plan-A accuracy improves by $+1.874$ pp ($[+1.290, +2.458]$), while the forced Plan-B contrast is $+0.592$ pp ($[-0.012, +1.196]$).

An auxiliary offline reconstruction is consistent with an execution-tilted change but does not exactly reconstruct the directly measured gain; full component estimates and residuals are reported in Appendix F.3. The direct paired evaluation remains the primary EF-RLVR result.

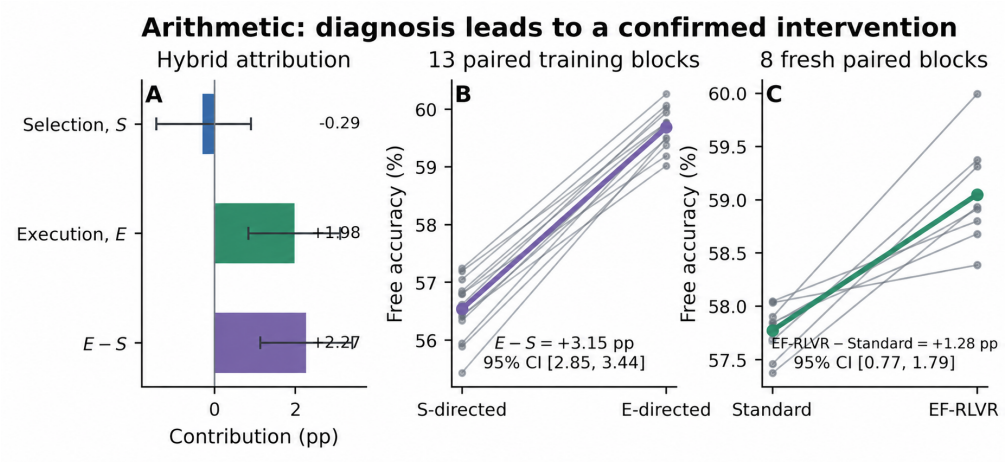


Figure 2: Arithmetic attribution and prospective training comparisons. (a) Prespecified full-repeat Hybrid replacement: execution-minus-selection contribution is $+2.27$ pp (95% task-cluster CI $[1.14, 3.41]$). (b) The specified E-directed procedure exceeds S-directed training in all 13 paired blocks ($+3.15$ pp; 95% paired-seed CI $[2.85, 3.44]$). (c) EF-RLVR exceeds Standard Joint RLVR in all eight fresh confirmation pairs ($+1.28$ pp; 95% paired-seed CI $[0.77, 1.79]$). These studies test distinct contrasts; their uncertainty units differ. Panels (b) and (c) use different vertical ranges.

Table 1: Post-outcome 2×2 factorial ablation of the EF-RLVR objective. Effects are paired free-accuracy differences in percentage points. I_λ denotes the interaction between selector-loss removal and executor up-weighting and is distinct from the component-replacement interaction I_{swap}.

Contrast	Effect (pp)	95% CI
Remove selector loss, $\lambda_E = 0.5$	$+1.170$	$[+0.797, +1.544]$
Remove selector loss, $\lambda_E = 1$	$+1.228$	$[+0.621, +1.836]$
Increase λ_E: $0.5 \to 1$, $\lambda_S = 0.5$	$+0.050$	$[-0.460, +0.561]$
Increase λ_E: $0.5 \to 1$, $\lambda_S = 0$	$+0.108$	$[-0.468, +0.685]$
Mean effect: selector-loss removal	$+1.199$	$[+0.764, +1.635]$
Mean effect: executor up-weighting	$+0.079$	$[-0.402, +0.561]$
Selector-off only − executor-up only[†]	$+1.120$	$[+0.359, +1.881]$
Objective-factor interaction I_λ	$+0.058$	$[-0.450, +0.566]$

[†] Additional post-outcome paired contrast. The reported interval is a nominal, unadjusted 95% paired-block t interval; exact two-sided sign-flip $p = 0.0234$.

Factorial objective ablation. After the EF-RLVR confirmation, we completed the 2×2 objective design by adding the two missing coefficient settings. These runs and their analysis were frozen before observing the new outcomes. Selector-term removal improves free accuracy at both executor coefficients (Table 1), with a mean factorial effect of $+1.199$ pp (95% CI $[+0.764, +1.635]$). Executor up-weighting has a substantially smaller and imprecise mean effect of $+0.079$ pp ($[-0.402, +0.561]$).

A post-outcome supplementary contrast directly compares selector removal alone, $(0, 0.5)$, with executor up-weighting alone, $(0.5, 1)$. Selector removal yields higher free accuracy by 1.120 pp (95% paired-block t interval [0.359, 1.881]; exact two-sided sign-flip $p = 0.0234$). This directly supports a larger benefit from selector-term removal within the tested configuration. The independent EF-RLVR–Standard comparison remains the primary confirmation.

4.4 Routing Opportunity and Selector Training in Search

Initial opportunity and training outcome. Under the production execution interface, a post-outcome, Base-only audit on a pair-disjoint task set shows substantial structured routing value: the frozen affine→A, quartic→B router exceeds the stronger fixed plan by $+10.94$ pp (95% CI

Table 2: Summary of Search results. Base routing estimates use the pair-disjoint production-interface audit, whereas the S/E and Hybrid contrasts use the fresh formal evaluation set. Trained-endpoint routing estimates are post-outcome archive-recovery diagnostics on the formal endpoints. Effects are in percentage points.

Study	Estimand	Effect	95% CI
Base routing audit	Router − best fixed, structured	$+10.94$	$[+6.25, +15.23]$
Base routing audit	Router − best fixed, answer-only	$+0.78$	$[-3.32, +4.69]$
Trained-endpoint audit	Router − best fixed, structured	$+8.960$	$[+7.567, +10.353]^{\dagger}$
Trained-endpoint audit	Router − best fixed, answer-only	$+3.333$	$[+1.848, +4.817]^{\dagger}$
Search S/E	$\mathrm{Acc}(S) - \mathrm{Acc}(E)$	-0.586	$[-2.417, +1.245]$
Search Hybrid	$D^{S}_{\mathrm{attr}} - D^{E}_{\mathrm{attr}}$	$+1.782$	$[-0.831, +4.395]$

†Post-outcome endpoint diagnostics. Headroom is computed separately for each endpoint, averaged within each S/E training block, and summarized across the eight blocks using a 95% block-level t interval. These intervals condition on the evaluated task set.

$[+6.25, +15.23]$). The original-interface qualification result and protocol-alignment details are reported in Appendix G.

Across eight fresh paired Search training blocks, S-directed training does not establish an advantage over E-directed training:

$$\mathrm{Acc}(S) - \mathrm{Acc}(E) = -0.586 \text{ pp}, \qquad 95\% \text{ CI} = [-2.417, +1.245].$$

The S-directed selector also remains near-deterministic on Plan A: mean $p(A)$ increases from 98.706% at Base to 99.536% after training, while routing accuracy remains 49.902%. A post-outcome Base-only label/order audit indicates that the initial concentration is primarily label-anchored; details are reported in Appendix H.3.

A block-aware analysis of archived forced evaluations from the original S/E endpoints also finds positive post-training routing headroom: $+8.960$ pp under structured scoring (95% block-level t CI $[+7.567, +10.353]$) and $+3.333$ pp under answer-only scoring ($[+1.848, +4.817]$; Table 2; Appendix G.5). This post-outcome diagnostic complements the fresh follow-up below.

Uniform action weights do not yield a detectable routing gain. We tested whether replacing old-policy action weights with equal weights for A and B would improve routing. The prospectively frozen comparison used eight paired training blocks, disjoint training sets across blocks, and a fresh 128-task test set.

Uniform-S does not improve the prospectively frozen primary routing endpoint:

$$\mathrm{RoutingAcc}(\mathrm{Uniform}) - \mathrm{RoutingAcc}(\mathrm{Original}) = -0.073 \text{ pp},$$

with 95% CI $[-0.399, +0.253]$ and exact sign-flip $p = 0.7188$. Both trained selectors remain near-deterministic on Plan A ($p(A) = 98.73\%$ for Original-S and 99.00% for Uniform-S).

Neither downstream free-performance contrast establishes an improvement: the Uniform-minus-Original differences are -1.538 pp ($[-3.289, +0.213]$) for structured correctness and -1.172 pp ($[-2.969, +0.625]$) for final-answer correctness. These intervals do not establish that Uniform-S is harmful, but the frozen success criterion is not met.

Opportunity persists after training. Forced evaluations on the fresh test set show that routing opportunity persists under both objectives (Figure 3; Appendix H.4). The frozen router exceeds the stronger fixed plan by 13.23 and 12.35 pp under structured scoring, and by 3.56 and 2.17 pp under answer-only scoring, for Original-S and Uniform-S respectively.

Thus, failure to learn conditional routing is not explained by an absence of input-dependent routing opportunity in these evaluations. The tested uniform-weighting intervention is insufficient to exploit the available opportunity.

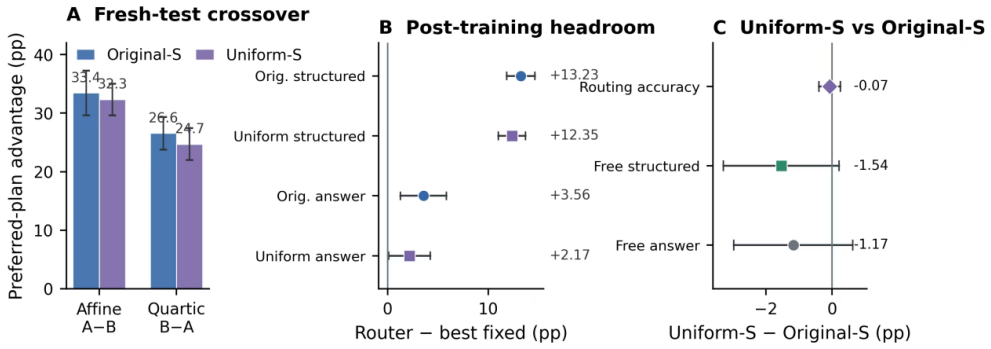


Figure 3: **Search: routing opportunity without learned routing. (A)** Forced-plan crossover after Original-S and Uniform-S training. **(B)** Frozen-router headroom under structured and answer-only scoring. **(C)** Uniform-S minus Original-S effects in the prospectively frozen eight-pair follow-up. Error bars show 95% paired-block intervals; both selectors remain near-Always-A.

5 Discussion and Limitations

The arithmetic contribution ordering is supported by the prespecified full-repeat Hybrid analysis, but is not statistically established when all four cells are restricted to four common repetitions. The point estimate remains execution-tilted, while both the magnitude and precision of the relative attribution depend on repetition scope. Interaction-based mechanistic interpretations are therefore tentative.

In our arithmetic configuration, removing the direct selector-policy term improves performance, but the ablation does not identify whether this benefit arises from gradient conflict, noise, or other shared-parameter optimization effects. EF-RLVR provides a separate prospective training result, while the factorial follow-up localizes its benefit primarily to removing the direct selector-policy term. Because parameters are shared, this intervention changes training-signal placement rather than disabling an independent selector module.

Search shows the complementary limitation: measurable routing opportunity can remain unexploited by both tested selector objectives. The absence of learned routing cannot be attributed solely to the use of a structured certificate metric: trained-endpoint evaluations also show positive answer-only routing headroom. The uniform-weighting result narrows one proposed explanation but does not identify the remaining optimization mechanism. Generalization is limited by one 1.5B model, two controlled two-plan environments, and the exclusion of selector-generated rationale.

6 Conclusion

We introduced a matched plan interface for attributing RLVR gains through selector–executor replacement. In controlled arithmetic, the prespecified full-repeat Hybrid favors execution under symmetric attribution, with sensitivity to evaluation repetition scope. Separately, EF-RLVR improves Standard Joint RLVR in an independent confirmation, and the factorial follow-up supports a larger benefit from selector-term removal than from executor up-weighting.

In Search, routing opportunity persists after training, but neither tested selector objective learns to exploit it. Together, these findings distinguish where realized gains are expressed, which training interventions help, and which available opportunities remain unrealized.

Reproducibility Statement

We provide the information needed to reproduce our main experiments and analyses across the main paper and appendix. The construction of the controlled plan-selection and plan-conditioned execution protocols, together with the component-swap analysis and attribution procedure, is described in the Methods section. The experimental setup, training and evaluation protocols, and statistical pro-

cedures are documented in the Experiments section and the corresponding appendices. Additional implementation details, prompts, hyperparameters, environment specifications, and extended results are provided in the appendix.

REFERENCES

Xingwu Chen, Tianle Li, and Difan Zou. Reshaping reasoning in llms: A theoretical analysis of rl training dynamics through pattern selection. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2OO399hRD6. Earlier arXiv version titled "On the Mechanism of Reasoning Pattern Selection in Reinforcement Learning for Language Models".

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective on reinforcement learning for llms. *arXiv preprint arXiv:2506.14758*, 2025. URL https://arxiv.org/abs/2506.14758.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025a. URL https://arxiv.org/abs/2502.01456.

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

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Guanhua Huang, Tingqiang Xu, Mingze Wang, Qi Yi, Xue Gong, Siheng Li, Ruibin Xiong, Kejiao Li, Yuhao Jiang, and Bo Zhou. Low-probability tokens sustain exploration in reinforcement learning with verifiable reward. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 24158–24188, San Diego, California, United States, 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1209. URL https://aclanthology.org/2026.findings-acl.1209/.

Kexin Huang, Haoming Meng, Junkang Wu, Jinda Lu, Chiyu Ma, Ziqian Chen, Xue Wang, Bolin Ding, Jiancan Wu, Xiang Wang, Xiangnan He, Guoyin Wang, and Jingren Zhou. On the direction of RLVR updates for LLM reasoning: Identification and exploitation. *arXiv preprint arXiv:2603.22117*, 2026b. URL https://arxiv.org/abs/2603.22117.

Kosuke Imai, Luke Keele, and Dustin Tingley. A general approach to causal mediation analysis. *Psychological Methods*, 15(4):309–334, 2010. doi: 10.1037/a0020761.

Akshay Krishnamurthy, Audrey Huang, and Nived Rajaraman. Select and improve: Understanding the mechanics of post-training for reasoning. *arXiv preprint arXiv:2606.13125*, 2026. URL https://arxiv.org/abs/2606.13125.

Qingyao Li, Wenxiang Jiao, Shuai Shao, Kangning Zhang, Yuan Lu, Yi Guo, Weiwen Liu, Weinan Zhang, and Yong Yu. SkillGate: Training in-policy skill selection in long-horizon agents. *arXiv preprint arXiv:2608.18852*, 2026. URL https://arxiv.org/abs/2608.18852.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Judea Pearl. Direct and indirect effects. In Jack S. Breese and Daphne Koller (eds.), *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pp. 411–420. Morgan Kaufmann, 2001.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

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. URL https://arxiv.org/abs/2402.03300.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton, 1953.

Akshit Sinha, Arvindh Arun, Shashwat Goel, Steffen Staab, and Jonas Geiping. The illusion of diminishing returns: Measuring long horizon execution in LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3lm8lWYxiq.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2506.01939.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2506.14245.

Huimin Xu, Shuai Zhao, Xiaobao Wu, and Anh Tuan Luu. Understanding and preventing entropy collapse in RLVR with on-policy entropy flow optimization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17759–17771, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.879. URL https://aclanthology.org/2026.findings-acl.879/.

Youngsik Yoon, Siwei Wang, Wei Chen, and Jungseul Ok. When are experts misrouted? counterfactual routing analysis in mixture-of-experts language models. *arXiv preprint arXiv:2605.07260*, 2026. URL https://arxiv.org/abs/2605.07260.

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

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2504.13837.

Qiancheng Zhou and Ruizhe Li. Locked at the entrance, open inside: Where rlvr narrows the solution space. *arXiv preprint arXiv:2608.29188*, 2026. URL https://arxiv.org/abs/2608.29188.

A AI USE STATEMENT

Large Language Models were used solely to improve the readability, grammar, and language of the paper. The authors reviewed and edited all AI-assisted text and take full responsibility for the final content of the paper.

B IMPLEMENTATION AND REPRODUCIBILITY DETAILS

This appendix provides implementation details for reproducing the formal experiments. All confirmatory runs were executed from frozen task manifests, seeds, model checkpoints, and analysis scripts. Unless explicitly labeled as exploratory or post-outcome, experimental configurations were fixed before the corresponding outcomes were inspected.

B.1 MODEL AND TRAINING CONFIGURATION

All formal arithmetic experiments use DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025; Qwen Team et al., 2025) as the Base model. Every prospective arm initializes its actor from the same pristine Base checkpoint. A separate fixed copy of the same Base checkpoint serves as the reference policy for KL regularization; the reference policy has no optimizer and is not updated during training.

Training data and task schedule. The formal arithmetic data contain two frozen 400-task training splits (800 distinct training tasks in total) and a separate 1,024-task held-out evaluation set, with zero overlap in either task identifiers or input tuples. Each formal arm uses the split specified by its frozen input slot. Within a slot, the S- and E-directed arms use the same ordered semantic task IDs.

The resolved training dataset therefore contains 400 tasks per arm. With a batch size of eight, each epoch contains 50 outer steps. Training runs for two epochs with `shuffle=false`, using sequential sampling, so every arm traverses task indices $0, \ldots, 399$ in stored order twice. Each task consequently appears exactly once per epoch and exactly twice over the complete training run. This yields 800 semantic-group presentations per arm.

The frozen slot assignment alternates across paired blocks: F01, F03, F05, F07, F09, F11, and F13 use T1, whereas F02, F04, F06, F08, F10, and F12 use T2.

Generation budgets. The maximum prompt length is 512 tokens and the outer response limit is 1,664 tokens. Within the production agent loop, reasoning generation is capped at 1,536 tokens and the final structured action at 64 tokens. The selector action consists of a single token constrained to $\{A, B\}$. Rollouts use temperature 0.7, top-$p = 1.0$, and top-$k = -1$. The rollout engine uses a maximum model length of 2,176 tokens.

The final structured action is constrained to a compact JSON integer array whose length is determined by the assigned plan: three integer values for Plan A and four for Plan B. If reasoning does not naturally emit the closing marker before the reasoning budget is exhausted, the environment inserts the closing token before the structured action; this injected token is excluded from the policy-loss mask.

Optimization configuration. Table 3 reports the resolved optimization configuration used by all F01–F13 formal arms. These values are taken from the frozen resolved Hydra configuration rather than reconstructed from software defaults.

With zero warmup steps, the learning rate remains fixed at 1×10^{-6} throughout all 100 optimization steps.

Matched sampled-experience budget. The formal selector- versus executor-directed intervention uses a separately qualified channel-specific production path. Each arm contains 100 outer optimization steps. Every step contains eight semantic groups, and every semantic group produces eight executor rollouts, with exactly four externally assigned to Plan A and four to Plan B. Thus, each arm

Table 3: Resolved optimization configuration for the formal F01–F13 arithmetic S/E intervention.

Parameter	Value
Optimizer	AdamW
Learning rate	1×10^{-6}
Adam betas	$(0.9, 0.999)$
Weight decay	0.01
LR scheduler	`get_constant_schedule_with_warmup`
Warmup steps	0
PPO epochs / outer step	1
PPO clip ratio	0.2
Negative-advantage clip c	3.0
Gradient-norm clip	1.0
Reference-KL coefficient	0.001
Executor KL estimator	`low_var_kl`
Entropy coefficient	0
GRPO std normalization	enabled
TIS importance correction	disabled
KL in reward	disabled

Table 4: Formal arithmetic selector- and executor-directed training configuration. "Matched" refers to sampled experience, plan exposure, and update count, not to the number of gradient-bearing tokens.

Quantity	Selector-directed S	Executor-directed E
Outer steps	100	100
Semantic groups / step	8	8
Rollouts / group	8	8
Plan exposure / group	$4A + 4B$	$4A + 4B$
Semantic-group presentations / arm	800	800
Generated rollouts / arm	6,400	6,400
Optimizer / scheduler updates	100 / 100	100 / 100
Direct-loss rows / step	16 selector rows	64 executor rows
Gradient-bearing tokens / arm	1,600 exactly	mean 6,399,322
Direct-loss unit	one-token plan decision	reasoning/action trajectory

uses

$$100 \times 8 \times 8 = 6{,}400$$

generated executor trajectories and exactly 100 optimizer and scheduler updates.

Table 4 summarizes the quantities matched across the two arithmetic intervention channels. The channels deliberately differ in the number and location of gradient-bearing tokens.

The large difference in gradient-bearing token count is intentional. A selector update acts on a one-token decision over two plans, whereas an executor update acts on a long plan-conditioned reasoning trajectory. Consequently, the formal S–E comparison should be interpreted as allocating the same sampled-experience and optimizer-update budget to two channel-specific objectives, rather than as a token-equalized comparison of intrinsic component learnability.

The production implementation performs exactly one optimizer update per outer step. Losses are first reduced within their intended semantic unit and are then combined using globally defined coefficients whose sum is one. Micro-batching is used only to accumulate additive gradient contributions and does not introduce additional optimizer steps. An earlier generic trainer path failed this batch-semantics requirement during pre-training qualification and was not used for the formal confirmatory intervention.

B.2 Training-Signal-Placement Objectives

Selection- and execution-directed training update the same shared model parameters. The intervention changes where the direct policy objective is applied; it does not assign selection and execution

to disjoint parameter subsets. Accordingly, updating one channel can change behavior measured in the other.

Selector-directed objective. For each semantic group g, the environment generates four independent executor rollouts under Plan A and four under Plan B. Let

$$Q_g(z) = \frac{1}{4}\sum_{m=1}^{4} R_{g,z,m}, \qquad z \in \{A, B\},$$

where $R_{g,z,m} \in \{0, 1\}$ is the programmatic verifier reward. Thus, both plan-conditioned values are estimated from the same balanced sampled-experience budget.

The selector objective then explicitly enumerates both actions rather than sampling only one plan for the policy update. Using the frozen old selector policy, we form

$$b_g = \sum_{z\in\{A,B\}} \pi_{\text{old}}(z \mid x_g) Q_g(z)$$

and

$$A_g(z) = Q_g(z) - b_g.$$

The policy ratio for action z is

$$r_g(z) = \frac{\pi_\theta(z \mid x_g)}{\pi_{\text{old}}(z \mid x_g)}.$$

The selector policy objective is therefore

$$\mathcal{L}_S^{\text{PG}} = \frac{1}{G}\sum_{g=1}^{G}\sum_{z\in\{A,B\}} \pi_{\text{old}}(z \mid x_g)\, \ell_{\text{PPO}}(r_g(z), A_g(z)), \qquad G = 8,$$

where ℓ_{PPO} denotes the frozen clipped PPO token objective (Schulman et al., 2017). Because the selector support contains exactly two allowed actions, A and B, the reference regularizer is evaluated exactly on this normalized support:

$$\mathcal{L}_S^{\text{KL}} = \frac{1}{G}\sum_{g=1}^{G} D_{\text{KL}}(\pi_\theta(\cdot \mid x_g) \,\|\, \pi_{\text{ref}}(\cdot \mid x_g)).$$

Thus, the selector KL is an exact two-action categorical KL rather than a single-action sampled approximation.

Executor-directed objective. The executor-directed arm uses the same eight groups and the same balanced $4A + 4B$ rollout schedule. In this arm, the selector receives no direct policy-loss row. Instead, all 64 forced-plan executor trajectories in the outer step enter the execution objective.

Within each semantic group, verifier rewards from the eight executor trajectories are converted into group-local GRPO advantages. The resulting trajectory-level advantage $A_{g,j}$ is broadcast over the valid generated tokens of trajectory j. For a trajectory with valid loss-token set $\mathcal{T}_{g,j}$, the policy loss is first reduced as a token mean,

$$\bar{\ell}_{g,j} = \frac{1}{|\mathcal{T}_{g,j}|}\sum_{t\in\mathcal{T}_{g,j}} \ell_{\text{PPO}}(r_{g,j,t}, A_{g,j}).$$

The 64 trajectory means then receive equal weight:

$$\mathcal{L}_E^{\text{PG}} = \frac{1}{64}\sum_{g=1}^{8}\sum_{j=1}^{8} \bar{\ell}_{g,j}.$$

The executor loss mask includes generated reasoning tokens and generated structured-action tokens. If the environment inserts a closing token after the reasoning budget is exhausted, that environment-injected token is excluded from the policy-loss mask. When reference-KL regularization is applied,

its tokenwise contribution is reduced using the same trajectory-wise token means and global trajectory coefficients as the policy objective.

This reduction prevents longer executor traces from receiving greater weight merely because they contain more tokens. The production implementation enforces one complete outer-step objective before the optimizer update; splitting the rows into micro-batches changes memory usage but not the mathematical reduction.

Budget interpretation. The two channel-directed interventions are exactly matched in outer steps, semantic groups, generated rollouts, Plan-A/Plan-B exposure, and optimizer updates. They are not matched in gradient-bearing token count or backward-call count. Across the formal runs, each S arm contains exactly 1,600 gradient-bearing selector tokens, whereas an E arm contains approximately 6.4 million gradient-bearing executor tokens on average.

The resulting contrast therefore answers the following intervention question: given the same sampled executor experience and the same number of parameter updates, is it more effective to direct the policy objective toward plan selection or toward plan-conditioned execution? It does not estimate a hypothetical comparison in which the two channels are forced to use equal numbers of loss tokens, equal backward calls, or equal effective optimization geometry.

Relation to Standard Joint RLVR and EF-RLVR. The channel-directed S/E experiment above is distinct from the Standard-versus-EF-RLVR comparison. For Standard Joint RLVR, the direct selector/executor policy weights are

$$(\lambda_S, \lambda_E) = (0.5, 0.5),$$

whereas Execution-Focused RLVR (EF-RLVR; denoted E0 in the frozen experimental manifests) uses

$$(\lambda_S, \lambda_E) = (0, 1).$$

Reference-KL weighting is unchanged between Standard and EF-RLVR. Because all objectives update shared model parameters, EF-RLVR should not be interpreted as freezing the selector, nor should a nonzero selector weight be interpreted as affecting selection alone. The Standard–EF-RLVR experiment evaluates the complete objective configuration rather than the isolated causal effect of removing the direct selector-policy loss.

C Arithmetic Interface Qualification and Reconstruction

The attribution framework requires free and forced evaluation to expose the same execution problem once a plan has been selected. We therefore validate the canonical executor interface before interpreting selection–execution effects.

C.1 Canonical Interface Qualification

Example. For $(X, Y, Z, K) = (555, 364, 47, 547)$, Plan A returns

$$[919, 43193, 42646],$$

whereas Plan B returns

$$[26085, 17108, 43193, 42646].$$

Both terminate at the same verified answer, 42646.

The final interface passed all prespecified Gate-0 qualification criteria, including executor parsing, environment plan enforcement, nondegenerate forced execution, and free-performance reconstruction (Table 5). The reconstruction-difference 95% whole-task bootstrap interval was $[-1.578, +1.941]$ pp. This interval includes zero but does not establish statistical equivalence; we use the qualification to establish a nondegenerate, mechanically matched interface rather than an equivalence claim.

The repair modified only the final-action grammar, replacing the failing whitespace-permitting schema with a compact grammar over the same integer-array action space. Because rerunning the stochastic model changed many reasoning trajectories, we use this qualification only to establish a viable and comparable execution interface; we do not claim behavioral equivalence to the failed parent run or independence between selector and executor effects.

Table 5: Arithmetic interface qualification.

Quantity	Result
Parser success before repair	94.68%
Parser success after repair	100.00%
Forced Plan-A success	67.11%
Forced Plan-B success	66.99%
Reconstruction gap	0.1785 pp

C.2 Historical Reconstruction: Correct Sources and Residuals

When free-selection and forced-execution measurements are available separately, we reconstruct the four component combinations as

$$\widehat{V}_{ij} = \frac{1}{N} \sum_{n=1}^{N} \sum_{z} \widehat{p}_i(z \mid x_n) \widehat{e}_j(z, x_n). \tag{12}$$

This reconstruction is used only as an auxiliary interface-consistency diagnostic; direct Hybrid evaluation of the four component combinations remains the primary attribution measurement.

The historical reconstruction uses the two frozen Joint Gate 1 triplets, evaluated on 1,024 common tasks. Their free accuracies are 57.800% and 57.483%, compared with 55.603% for Base. Observed gains are $+2.197$ and $+1.880$ pp; reconstructed gains are $+2.386$ and $+1.857$ pp. The associated selection contributions are $+0.103$ and $+0.060$ pp, whereas the execution contributions are $+2.283$ and $+1.797$ pp.

For each triplet, define the gain residual as observed gain minus reconstructed gain. The archived matched-task bootstrap results are:

Historical triplet	Residual (pp)	95% task-bootstrap CI (pp)
T1	-0.1892	$[-2.0356, +1.7228]$
T2	$+0.0229$	$[-1.8433, +1.8921]$

The residual point estimates are small, but the intervals permit non-negligible discrepancies and do not establish equivalence. We therefore treat reconstruction auxiliary reconstruction-consistency evidence and use direct Hybrid component substitution as the primary attribution analysis.

D Full Arithmetic Component-Swap Results

The Hybrid experiment evaluates all four combinations of Base and trained selectors and executors under the same canonical interface.

Table 6: Observed free-task accuracy under realized component substitutions.

	Base executor	Trained executor
Base selector	$V_{00} = 57.25\%$	$V_{01} = 57.76\%$
Trained selector	$V_{10} = 55.49\%$	$V_{11} = 58.94\%$

Historical Base and Hybrid baseline. The historical Base artifact contains 8,192 outcomes over 1,024 tasks, whereas the Hybrid V_{00} artifact contains 4,096 outcomes over the same task identifiers. Both reference the same verified Base checkpoint path. For all 3,963 V_{00} rows whose task–plan pair is represented in the historical artifact, canonical executor prompt hashes agree. The remaining 133 rows have no historical same-task/same-plan counterpart for this comparison; they are therefore not observed prompt mismatches.

Historical Base accuracy is 55.603% and V_{00} accuracy is 57.251%. Their descriptive difference is $+1.648$ pp, with a 20,000-resample whole-task bootstrap interval of $[-0.610, +3.870]$ pp. Repeat

counts differ, and the historical rows lack explicit request-seed fields. This audit supports alignment of the checked task identities, checkpoint path, and canonical inputs, but not exact request-level reproduction or statistical equivalence. All Hybrid contrasts use their own V_{00} baseline throughout.

Table 7: Full component-swap contrasts. Confidence intervals use 10,000 whole-task cluster bootstrap resamples.

Quantity	Estimate (pp)	95% CI (pp)
Selector effect, Base executor	-1.766	[-3.410,-0.065]
Selector effect, trained executor	+1.180	[-0.488,+2.824]
Executor effect, Base selector	+0.505	[-1.164,+2.205]
Executor effect, trained selector	+3.451	[+1.831,+5.094]
Selection contribution S	-0.293	[-1.441,+0.899]
Execution contribution E	+1.978	[+0.834,+3.109]
$E - S$	+2.271	[+1.139,+3.410]
Component-replacement interaction I_{swap}	+2.946	[+0.505,+5.347]

Common-repeat sensitivity and seed alignment. The prespecified Hybrid analysis uses four repetitions for V_{00} and V_{11} and twelve repetitions for V_{01} and V_{10}. As a post-outcome repetition-scope sensitivity analysis, we restrict all four cells to the common repetition indices 0–3.

For each of 10,000 bootstrap draws, we resample the 1,024 tasks with replacement while retaining all four component-replacement cells and all four retained repetitions within each sampled task. Thus, uncertainty is computed using a whole-task cluster bootstrap rather than by treating task–repetition rows as independent observations.

Under the restricted repetition scope, the symmetric contributions are

$$S^{0:3} = +0.037 \text{ pp}, \qquad 95\% \text{ CI} = [-1.306, +1.392],$$

and

$$E^{0:3} = +1.648 \text{ pp}, \qquad 95\% \text{ CI} = [+0.208, +3.052].$$

The direct relative attribution contrast is

$$(E - S)^{0:3} = +1.611 \text{ pp}, \qquad 95\% \text{ CI} = [-0.318, +3.564],$$

compared with the prespecified full-repeat estimate of $+2.271$ pp ($[+1.139, +3.410]$).

The point estimate therefore remains execution-tilted, but the restricted analysis does not establish that the execution contribution exceeds the selection contribution. The component-replacement interaction also decreases from $+2.946$ pp to

$$I_{\text{swap}}^{0:3} = +0.122 \text{ pp}, \qquad 95\% \text{ CI} = [-2.612, +2.856].$$

These differences arise from the set of included evaluation outcomes rather than from changing the algebraic estimand or bootstrap pairing. Because the repetitions use the same evaluation protocol, we treat the full-repeat and common-repeat calculations as estimates of the same population quantities using different subsets of stochastic evaluation outcomes. The prespecified full-repeat analysis remains the primary analysis.

Seed alignment. For all 4,096 retained task–repetition pairs, reasoning and action seed values match between V_{00} and V_{01} and between V_{10} and V_{11}. Selector seeds also match across executor endpoints within each selector, but differ between the Base and trained selectors. We therefore do not describe the four-cell evaluation as using identical random numbers across all cells. Matching recorded seed values also does not by itself establish identical random-number consumption inside every model call.

D.1 Final-Answer Sensitivity Analysis

The post-outcome sensitivity analysis rescores the same frozen Hybrid outputs. It retains the plan-specific parsing contract: a length-three integer array for Plan A and a length-four array for Plan

B. Only the final element is checked against the answer; invalid or wrong-length arrays still fail. Thus, this analysis relaxes intermediate-node correctness rather than output validity and is not a new generation experiment.

Selection, execution, their difference, and interaction are -0.338, $+1.656$, $+1.994$, and $+2.677$ pp, respectively. The execution interval is $[+0.545, +2.791]$ pp and the $E - S$ interval is $[+0.863, +3.125]$ pp. The point estimate of $E - S$ is smaller than under full-structure scoring, while its interval still excludes zero.

Table 8: Hybrid attribution under the preregistered full-structure outcome and the post-outcome final-answer sensitivity criterion.

Quantity	Full structure	Final answer
Selection S	-0.293	-0.338
Execution E	+1.978	+1.656
$E - S$	+2.271	+1.994
Interaction I	+2.946	+2.677

Plan A must still produce a valid length-3 integer array and Plan B a valid length-4 integer array; invalid or wrong-length outputs remain failures. Thus, this sensitivity changes the correctness criterion for intermediate nodes, not output-format validity.

E Arithmetic Training-Signal-Placement Details

We conduct 13 prospectively specified paired training blocks. Within each block, S- and E-directed arms start from the same pristine Base checkpoint and share the matched macro training budget. Feasibility runs are excluded from confirmatory inference.

Averaging forced execution across the two plans gives an E–S difference of $+3.3635$ pp (95% CI $[+3.1192, +3.6078]$), with all 13 paired differences positive. For free accuracy,

$$E - S = +3.1485 \text{ pp}, \qquad 95\% \text{ CI} = [+2.8538, +3.4432].$$

All 13 paired differences are positive. The exact two-sided sign-flip test gives $p = 2.44 \times 10^{-4}$.

The Plan-A selection-rate contrast has the opposite sign:

$$E - S = -2.1090 \text{ pp}, \qquad 95\% \text{ CI} = [-2.4679, -1.7501].$$

Thus, the execution-directed advantage in end-task accuracy is not explained by a larger shift toward the dominant plan.

Optimization diagnostics and inferential scope. The recorded mean pre-clipping gradient norm is 1.140 for S and 0.517 for E. The fraction of steps exceeding the gradient-norm threshold of 1.0 is 43.15% and 1.62%, respectively. Recorded weighted KL terms are 0.01197 and 0.00116.

These quantities document different realized optimization dynamics; they do not identify effective parameter-update magnitude or isolate the cause of the performance difference. Gradient-norm clipping is distinct from likelihood-ratio clipping inside the policy objective. Cumulative parameter-update norms were not logged, and exact final checkpoint displacement cannot be reconstructed from the currently preserved archive.

Update-event and loss-diagnostic coverage spans all 26 arms, whereas full trainer-scalar coverage spans 22 of the 26 arms. Each reported scalar should therefore be interpreted with its corresponding coverage and loss reduction.

Two-stage block-and-task bootstrap. We additionally conduct a post-outcome uncertainty sensitivity analysis that resamples both paired training blocks and evaluation tasks. Each of the 13 paired blocks contains 1,024 evaluation tasks, with the eight fixed evaluation repeats retained in the task-level mean.

Table 9: Summary of the primary arithmetic results reported in the main text. Effects are in percentage points.

Study	Estimand	Effect [95% CI]
Hybrid	$E - S$	$+2.271$ $[+1.139, +3.410]$
S/E training	$\mathrm{Acc}(E) - \mathrm{Acc}(S)$	$+3.148$ $[+2.854, +3.443]$
EF-RLVR	$\mathrm{Acc}(\mathrm{EF}) - \mathrm{Acc}(\mathrm{Standard})$	$+1.279$ $[+0.766, +1.791]$

Table 10: Aggregate results across 13 paired arithmetic training blocks.

Metric	S	E	E–S
Free accuracy (%)	56.536	59.685	+3.148
Forced Plan A	–	–	+3.573
Forced Plan B	–	–	+3.154
Plan-A selection rate (%)	98.052	95.943	-2.109

For each of 10,000 bootstrap replicates, we first resample the 13 paired training blocks with replacement. Within each selected block occurrence, we then independently resample the 1,024 evaluation-task IDs with replacement. The S- and E-directed endpoints use identical task draws within the same selected block, preserving the paired comparison, whereas different selected blocks do not share task draws.

The original paired-block estimate is

$$\mathrm{Acc}(E) - \mathrm{Acc}(S) = +3.148 \text{ pp}.$$

The two-stage bootstrap distribution has mean $+3.146$ pp and standard deviation 0.230 pp, with a 95% percentile interval of

$$[+2.695, +3.594] \text{ pp}.$$

Across the 13 observed block-level contrasts, the standard deviation is 0.488 pp. This sensitivity analysis incorporates evaluation-task sampling variation together with resampling across the 13 paired training blocks. It remains conditional on the two frozen 400-task training splits and the corresponding frozen training manifests, and therefore does not quantify variation over newly generated training manifests.

The higher S gradient norms and larger weighted KL show that the raw gradient-bearing-token ratio should not be interpreted as a proportional difference in realized update magnitude. At the same time, these diagnostics do not eliminate differences in information content, correlation structure, loss geometry, or optimizer dynamics between the two procedures.

Common-Base comparison. The historical Base evaluation and all 26 F01–F13 endpoints use the same 1,024-task file, evaluator, request-seed table, condition grid, and canonical executor prompts. The common Base free accuracy is 55.603%. Descriptively, S improves over this fixed Base by $+0.933$ pp and E by $+4.082$ pp. Because Base is a single fixed checkpoint evaluation rather than a randomized training arm, these contrasts are diagnostic only.

F DEVELOPMENT AND INDEPENDENT CONFIRMATION OF EF-RLVR

F.1 DEVELOPMENT AND INDEPENDENT CONFIRMATION

Table 13 records the experiments preceding EF-RLVR, which is named E0 in the frozen manifests. The earlier EA comparison did not establish superiority over Standard Joint RLVR. EA also changed group composition, so its result cannot isolate the effect of the loss coefficients.

In the subsequent two-seed development study, EA-W improved accuracy by 0.024 and 0.586 pp. Its mean gain of 0.305 pp fell below the prespecified 1.0 pp threshold, and no confirmation was

Table 11: Descriptive optimization diagnostics for the formal arithmetic S/E interventions. Update/loss diagnostics cover all 26 arms; full trainer scalars are available for 22 arms. Parameter-update norms were not logged.

Diagnostic	S-directed	E-directed
Mean pre-clip gradient norm	1.140	0.517
Steps above grad-clip threshold	43.15%	1.62%
Weighted KL contribution / step	0.01197	0.00116
Optimizer steps / arm	100	100
Cumulative parameter-update norm	not logged	not logged
Final checkpoint displacement	not computable from preserved archive	

Table 12: Per-block free-policy accuracy for the 13 paired arithmetic training-signal-placement blocks.

Block	S (%)	E (%)	E–S (pp)
F01	55.9448	59.7778	+3.8330
F02	56.7993	59.7656	+2.9663
F03	57.0435	60.0708	+3.0273
F04	55.4321	59.5093	+4.0771
F05	56.6162	60.0098	+3.3936
F06	56.4331	59.0210	+2.5879
F07	55.8838	59.3750	+3.4912
F08	56.4087	59.4849	+3.0762
F09	56.8481	59.9365	+3.0884
F10	56.3354	59.7290	+3.3936
F11	57.2388	60.2661	+3.0273
F12	56.7993	59.1919	+2.3926
F13	57.1899	59.7656	+2.5757
Mean	56.5364	59.6849	+3.1485

launched. E0, included as a diagnostic arm, improved accuracy by approximately 1.48 and 2.39 pp. We then froze a separate eight-pair comparison of E0 against Standard.

Only these eight fresh pairs contribute to the EF-RLVR confirmation. All eight differences are positive, with an exact two-sided sign-flip $p = 0.0078125$. Table 14 reports the direct evaluation and the auxiliary reconstruction separately.

F.2 Factorial objective ablation

After observing the EF-RLVR confirmation, we added the two missing coefficient settings to complete the factorial design. Their training and evaluation protocols and the original factorial contrasts were frozen before the new outcomes were inspected. The Standard and EF-RLVR runs were reused from the confirmation.

For each block, let

$$Y_{00} = Y(0.5, 0.5), \quad Y_{10} = Y(0, 0.5), \quad Y_{01} = Y(0.5, 1), \quad Y_{11} = Y(0, 1),$$

where the arguments are (λ_S, λ_E). The factorial main effects and interaction are

$$M_{\mathrm{S-off}} = \tfrac{1}{2}[(Y_{10} - Y_{00}) + (Y_{11} - Y_{01})],$$
$$M_{\mathrm{E-up}} = \tfrac{1}{2}[(Y_{01} - Y_{00}) + (Y_{11} - Y_{10})],$$
$$I_\lambda = Y_{11} - Y_{10} - Y_{01} + Y_{00}.$$

Table 15 gives the block-level accuracies; Table 1 reports the simple effects, main effects, and interaction.

After inspecting the factorial results, we additionally compared the two isolated modifications:

$$D_b = Y_{10,b} - Y_{01,b} = M_{\mathrm{S-off},b} - M_{\mathrm{E-up},b}.$$

Table 13: Experimental provenance for execution-focused training configurations. The "Direct S/E coefficients" column reports the nominal weights on the direct selector and executor policy-loss terms. EA and EF-RLVR are bundled configurations rather than isolated single-factor ablations; in particular, EA also changes group composition and within-group plan correlation. "E0" is the frozen manifest name of EF-RLVR.

Configuration	Direct S/E coefficients	Original status	Result and disposition
Standard	$(0.5, 0.5)$	Frozen comparator	Reference configuration.
EA	$(0.25, 0.75)$	Prospective eight-pair comparison	$+0.700$ pp vs. Standard; 95% CI $[-0.008, +1.409]$. Positive point estimate, but superiority was not established.
EA-W	See frozen development configuration	Two-seed development	Mean difference $+0.305$ pp; the frozen development gate failed, so no confirmation was launched.
EF-RLVR (E0)	$(0, 1)$	Diagnostic development arm, then separate fresh confirmation	Development mean $\approx +1.94$ pp; the separate eight-pair confirmation gives $+1.279$ pp, 95% CI $[+0.766, +1.791]$.

Table 14: Independent EF-RLVR confirmation. Direct evaluation outcomes are separated from offline component reconstruction. Differences are EF-RLVR–Standard.

Metric	Difference (pp)	95% CI (pp)
Direct evaluation		
Free accuracy	+1.279	[+0.766,+1.791]
Forced Plan A	+1.874	[+1.290,+2.458]
Forced Plan B	+0.592	[-0.012,+1.196]
Plan-A selection rate	-0.542	[-1.141,+0.057]
Offline component reconstruction		
Selection contribution S	-0.074	[-0.158,+0.010]
Execution contribution E	+1.787	[+1.192,+2.383]
Reconstructed gain $S + E$	+1.713	[+1.120,+2.306]
Direct–reconstructed residual	-0.434	[-1.221,+0.352]
$E - S$	+1.862	[+1.252,+2.471]
Interaction I	-0.010	[-0.042,+0.022]

The mean difference is 1.120 pp, with a nominal 95% paired-t interval of [0.359, 1.881] pp and an exact two-sided sign-flip $p = 0.023438$. This supplementary contrast favors selector-term removal over executor up-weighting within the tested configuration. It does not identify the optimization mechanism responsible.

F.3 Reconstruction Check

We apply the estimator in Appendix C.2 to each Standard and EF-RLVR endpoint. Writing $\rho_m = F_m - \widehat{F}_m$, the discrepancy between the directly measured and reconstructed treatment effects is

$$d_{\text{res}} = (F_{\text{EF}} - F_{\text{Standard}}) - (\widehat{F}_{\text{EF}} - \widehat{F}_{\text{Standard}}) = \rho_{\text{EF}} - \rho_{\text{Standard}}.$$

The direct gain is 1.279 pp, compared with a reconstructed gain of 1.713 pp (Table 14). Their difference is -0.434 pp (95% CI $[-1.221, 0.352]$). The model-level residuals are 0.130 pp for Standard ($[-0.350, 0.610]$) and -0.304 pp for EF-RLVR ($[-0.844, 0.236]$). These intervals allow non-negligible reconstruction error and do not establish equivalence.

Table 15: Per-block free accuracy for the post-outcome 2×2 objective ablation. The two factors are removal of the direct selector-policy term and increasing the executor coefficient from 0.5 to 1.

Block	Standard $(.5, .5)$	Selector-off $(0, .5)$	Executor-up $(.5, 1)$	EF-RLVR $(0, 1)$
E0C01	57.898	58.887	58.594	59.998
E0C02	57.678	59.180	57.947	59.375
E0C03	58.032	58.313	58.765	58.386
E0C04	57.849	59.607	57.776	58.679
E0C05	57.458	58.472	57.434	59.314
E0C06	58.044	59.277	56.873	58.801
E0C07	57.837	59.302	57.544	58.911
E0C08	57.373	58.496	57.642	58.936
Mean	57.771	58.942	57.822	59.050

G Search Environment and Routing Qualification

G.1 Task Construction

The Search environment provides two valid algorithms: interpolation search (Plan A) and binary search (Plan B). Tasks are constructed in matched affine- and quartic-spacing regimes. Array length and target-index structure are matched across regimes, while the observable spacing pattern determines which algorithm has the structured execution advantage.

The routing rule is frozen from the input structure:

$$r(x) = \begin{cases} A, & \text{affine spacing,} \\ B, & \text{quartic spacing.} \end{cases}$$

The router does not use the target answer or observed evaluation outcome.

G.2 Prospective Held-Out Routing Qualification

The prospective qualification contains 128 held-out tasks forming 64 matched template pairs, evaluated with four stochastic repeats. All 1,536 executor requests complete successfully.

Under joint structured correctness, the frozen router reaches 53.32% accuracy compared with 41.21% for the stronger fixed plan:

$$H_{\text{route}} = +12.11 \text{ pp}, \qquad 95\% \text{ CI} = [+7.81, +15.82].$$

This frozen symbolic router is defined directly from the observable spacing regime and is distinct from the earlier learned-router qualification, which failed its own qualification criterion. The plan crossover is substantial: interpolation search is favored on affine inputs, whereas binary search is favored on quartic inputs. In contrast, the Base selector chooses Plan A on approximately 97% of both regimes.

G.3 Production-Interface Alignment Audit

The original qualification predates the final Search production executor interface. After completion of the formal Search study, we therefore run a Base-only protocol-alignment audit using the frozen router under the production $1536 + 96$ execution interface.

This audit does not constitute a second independent confirmation: the routing rule and analysis question were already known. Its purpose is to determine whether the structured routing opportunity survives the interface change used in the formal Search experiments. The production-interface audit uses a separately frozen task sample from the same Search generator rather than the fresh task instances used for formal S/E evaluation; Appendix G.4 audits this relationship explicitly. All 1,536 expected rows are present with no missing or duplicate requests.

Table 16: Search routing opportunity under the production interface.

Outcome	Router–best fixed	95% CI
Joint structured	+10.94 pp	[+6.25,+15.23]
Final answer only	+0.78 pp	[-3.32,+4.69]

For the structured outcome, Plan A exceeds Plan B by $+34.38$ pp on affine inputs, whereas Plan B exceeds Plan A by $+21.88$ pp on quartic inputs.

The final-answer routing effect is substantially smaller, and its confidence interval includes zero; thus, the analysis does not establish a routing advantage under that outcome definition.

Executor information flow. A read-only audit of the frozen Search production loop verifies that forced execution is constructed from a canonical plan-specific executor prompt. The plan assignment selects the corresponding canonical Plan-A or Plan-B prompt, and executor reasoning is generated from that prompt rather than from selector-transcript tokens. The guided action is subsequently generated from the canonical executor prompt together with the executor's own generated reasoning and the environment-inserted closing marker when required.

The production loop records the semantic plan, assignment source, and canonical-prompt hash for each completion. This establishes the intended code-level separation between plan assignment and plan-conditioned execution. The archived materials do not provide a complete runtime dump of hidden states, caches, or process-level state, so we do not make a stronger claim of experimentally verified hidden-state independence.

G.4 Evaluation-set provenance

The Base production-interface audit reuses the 128 qualification tasks, arranged in 64 matched pairs. Formal S/E evaluation uses a fresh 128-task set with the same paired construction. The two sets have no overlap in task IDs, pair IDs, or complete semantic task records, including arrays, targets, answers, regimes, and probe traces.

The Search Hybrid evaluation uses exactly the formal S/E task set. Its 32,768 requests cover 16 endpoints, four replacement cells, 128 tasks, and four repeats, with no task or regime mismatches. Thus, the Base audit measures routing opportunity on the earlier qualification sample, whereas the S/E and Hybrid comparisons are task-matched on the fresh formal sample. Both samples use the same generator and frozen spacing-based router.

G.5 Routing Opportunity After Training

After the formal Search study, we recovered the archived ordinary evaluation rows for all 16 trained endpoints. Each endpoint contains 2,048 rows: 512 selector-only, 512 free, 512 forced-A, and 512 forced-B evaluations, covering 128 tasks arranged in 64 matched pairs. Request keys are unique, and both joint structured correctness and final-answer correctness are retained.

Using the previously frozen routing rule

$$r(x) = \begin{cases} A, & \text{affine spacing,} \\ B, & \text{quartic spacing,} \end{cases}$$

we recompute, for each endpoint, the routed value and the stronger fixed-plan value from its forced-A and forced-B outcomes.

For each trained endpoint $a \in \{S, E\}$ in block b, we first compute router-minus-best-fixed headroom as

$$H_{b,a} = V_{b,a}(r) - \max_{z \in \{A,B\}} V_{b,a}(z),$$

where the stronger fixed plan is selected separately for each endpoint. We then average the two endpoint-level headroom values within each paired training block:

$$\bar{H}_b = \frac{H_{b,S} + H_{b,E}}{2}.$$

Block-aware sensitivity. Across the eight training blocks, mean headroom is $+8.960$ pp under structured scoring, with a 95% block-level t interval of $[+7.567, +10.353]$ pp. Under answer-only scoring, mean headroom is $+3.333$ pp, with a 95% block-level t interval of $[+1.848, +4.817]$ pp. The block-averaged headroom values are positive in all eight blocks.

This post-outcome analysis measures average routing opportunity across the two endpoint types on the fixed evaluation set. It accounts for the paired training-block structure rather than treating the 16 trained endpoints as independent training replicates. It is not a prospectively frozen training comparison or an additional confirmation, and it does not identify why the learned selector fails to exploit the available opportunity.

H Search Training-Signal Experiments and Secondary Analyses

H.1 Primary Paired S/E Comparison

The formal Search study contains eight fresh paired S/E training blocks. Each trained endpoint is evaluated on 128 tasks with four repeats under each evaluation condition.

All formal endpoints pass completeness validation with no missing or duplicate rows.

Table 17: Search channel-directed training. Differences are S–E.

Metric	Difference (pp)	95% CI (pp)
Free joint accuracy	-0.586	[-2.417,+1.245]
Free answer accuracy	-1.514	[-3.849,+0.821]
Forced A joint accuracy	-1.050	[-3.517,+1.417]
Forced B joint accuracy	+0.366	[-1.904,+2.636]
Routing-regime accuracy	+0.391	[-0.302,+1.083]

For the primary free-joint outcome,

$$S - E = -0.586 \text{ pp}, \qquad 95\% \text{ CI} = [-2.417, +1.245],$$

with four blocks favoring S and four favoring E. This result does not establish superiority or equivalence of either training direction.

H.2 Prespecified Search Component-Swap Secondary Analysis

Scoring identity. Endpoint-level Hybrid summaries are indexed by both endpoint and outcome. The structured score and answer-only score are two measurements of the same endpoint, not independent trained models. All contrasts and confidence intervals are computed within a specified outcome definition.

The Hybrid analysis uses exactly the same 128 frozen evaluation tasks and 64 matched pairs as the formal S/E endpoint evaluation. We additionally evaluate selector–executor substitutions for the trained Search endpoints. For each endpoint type, define the attribution balance

$$D_{\text{attr}} = S_{\text{contrib}} - E_{\text{contrib}}.$$

For S-directed endpoints,

$$D^{S}_{\text{attr}} = +1.880 \text{ pp}, \qquad 95\% \text{ CI} = [+0.150, +3.610],$$

whereas for E-directed endpoints,

$$D^{E}_{\text{attr}} = +0.098 \text{ pp}, \qquad 95\% \text{ CI} = [-2.357, +2.552].$$

The direct treatment contrast is

$$C = D^{S}_{\text{attr}} - D^{E}_{\text{attr}} = +1.782 \text{ pp},$$

with

$$95\% \text{ CI} = [-0.831, +4.395], \qquad p = 0.151.$$

Table 18: Post-outcome Search Base-selector label/order counterbalance. Values are reconstructed from frozen forced executor outcomes; only selector decisions are newly generated.

Cell	Routing acc.	$p(\text{interp.})$	Joint value	Answer value
C_{00}	50.78	98.44	45.90	72.66
C_{01}	50.39	98.44	46.09	72.66
C_{10}	49.41	1.37	39.84	61.52
C_{11}	49.80	2.15	40.43	61.91

Thus, the S-directed endpoints have a descriptively more selection-tilted attribution profile, but the prespecified between-treatment contrast is not established.

Repeating the analysis under final-answer-only scoring gives

$$C = +2.051 \text{ pp}, \qquad 95\% \text{ CI} = [-0.937, +5.039],$$

with the same qualitative conclusion.

H.3 Selector concentration and label sensitivity

Across the eight S-directed blocks, mean Plan-A probability increases from 98.706% to 99.536%, while routing accuracy remains 49.902%. All eight blocks become more concentrated on A.

To test whether the Base preference tracks algorithm identity or the label itself, we independently swap the A/B mapping and the presentation order on the 128-task Base-audit set. We generate new selector decisions and reuse the archived forced-execution outcomes.

Changing presentation order has little effect. Swapping the labels reduces interpolation selection from 98.44% to 1.37–2.15%, while routing accuracy stays near 50% (Table 18). The Base preference therefore follows the literal label A more closely than the algorithm. This Base-only, post-outcome test does not establish the cause of the trained selectors' continued concentration.

H.4 Uniform Action Weighting

We conduct a prospectively frozen paired follow-up to test whether the near-deterministic Search selector behavior can be repaired by replacing the original old-policy-weighted counterfactual action weighting with uniform A/B weighting.

The study contains eight paired training blocks. Within each block, Original-S and Uniform-S use the same initialization seed, task order, and training schedule. Across blocks, the training sets are mutually disjoint: each block uses a distinct 800-task training set, for 6,400 independent training tasks in total. Evaluation uses a fresh 128-task set arranged into 64 matched affine/quartic pairs, with zero task-ID overlap with the earlier Search experiments.

The prospectively frozen primary endpoint is selector routing accuracy on the selector-only evaluation. The primary result is

$$\text{RoutingAcc}(\text{Uniform}) - \text{RoutingAcc}(\text{Original}) = -0.073 \text{ pp},$$

with 95% CI $[-0.399, +0.253]$ and exact two-sided sign-flip $p = 0.7188$. The prespecified success criterion is therefore not met. Mean Plan-A probability is 98.73% for Original-S and 99.00% for Uniform-S, so the selector remains near-deterministic under both objectives.

Uniform-S therefore does not improve the frozen primary routing endpoint, and the downstream free-performance contrasts also do not establish an improvement. Because the corresponding confidence intervals include zero, these results do not establish that Uniform-S is harmful; rather, the intervention fails to meet its prespecified success criterion.

Routing opportunity after both training objectives. We next use the forced-A and forced-B evaluations to measure whether input-dependent selection opportunity remains after either selector-training objective. The routing rule remains frozen as affine→A and quartic→B.

Table 19: Prospectively frozen Original-S versus Uniform-S comparison. Effects are Uniform-S minus Original-S in percentage points.

Metric	Effect (pp)	95% CI	Sign-flip p
Routing accuracy	-0.073	$[-0.399, +0.253]$	0.7188
Free structured	-1.538	$[-3.289, +0.213]$	0.0625
Free answer-only	-1.172	$[-2.969, +0.625]$	0.0781
Forced-A structured	-0.146	$[-1.526, +1.233]$	0.8125
Forced-A answer-only	$+0.488$	$[-2.709, +3.686]$	0.7344
Forced-B structured	-0.513	$[-2.964, +1.939]$	0.6406
Forced-B answer-only	-0.488	$[-2.116, +1.140]$	0.5156

Table 20: Routing headroom after Original-S and Uniform-S training on the fresh Search evaluation set. Headroom is the frozen router value minus the stronger fixed plan, in percentage points.

Endpoint family	Structured headroom	Answer-only headroom
Original-S	$+13.232$ $[+11.831, +14.634]$	$+3.564$ $[+1.281, +5.848]$
Uniform-S	$+12.354$ $[+10.996, +13.711]$	$+2.173$ $[+0.118, +4.228]$

The plan crossover is substantial under both endpoint families. For Original-S, Plan A exceeds Plan B by $+33.447$ pp on affine inputs, whereas Plan B exceeds Plan A by $+26.562$ pp on quartic inputs under structured scoring. For Uniform-S, the corresponding differences are $+32.324$ pp and $+24.707$ pp. Under answer-only scoring, the quartic Plan-B advantage also remains positive: $+7.129$ pp for Original-S and $+4.346$ pp for Uniform-S.

Thus, the failure of both selector objectives cannot be explained by the absence of input-dependent selection opportunity. The experiment instead shows that simple uniform counterfactual action weighting is insufficient to make the tested selector objective exploit that opportunity.

Primary selector result versus full-evaluation replay. The prospectively frozen primary selector result is the selector-only comparison reported above, -0.073 pp. A selector replay performed as part of the complete four-condition evaluation gives a secondary difference of -0.122 pp. This replay is retained only as a batching/run diagnostic and does not replace the frozen primary selector endpoint.

I Exploratory arithmetic routing analyses

Arithmetic selectors are highly concentrated, and natural on-policy groups can lack plan variation. The archived routing-qualification records are insufficient to reproduce the earlier headroom estimate under a verified manifest and scoring definition; we therefore do not use it to infer the presence or absence of routing opportunity.

A separate cross-fitted analysis tested whether Base execution readiness and routing opportunity predict the response to S- versus E-directed training. Only 51.27% of records received the same predefined regime assignment across folds, and the low-execution, high-routing region was sparsely populated. Although aggregate effects favored E across the analyzed subsets, these measurements did not support a reliable task-level rule for when S-directed training should perform better.

J Evidence Provenance

Table 21 summarizes the evidential status of the analyses reported in the paper.

Table 21: Evidence level and role of the main and supplementary analyses.

Analysis	Status	Role
Historical arithmetic reconstruction	Frozen Joint Gate 1 / auxiliary	Reconstruction consistency
Arithmetic Hybrid	Prospectively frozen	Primary attribution
Arithmetic common-repeat Hybrid	Post-outcome; existing outputs, repetitions 0–3	Repetition-scope sensitivity of attribution and interaction
Arithmetic final-answer rescoring	Post-outcome	Robustness
13-pair arithmetic S/E	Prospectively specified	Training-procedure comparison
Arithmetic two-stage block-and-task bootstrap	Post-outcome sensitivity	Uncertainty robustness
EA eight-pair comparison	Prospective	Earlier formal execution-focused test
EA-W	Development; frozen gate failed	Candidate screening
EF-RLVR (E0) eight-pair confirmation	Fresh prospective confirmation	Configuration-level confirmation
EF-RLVR reconstruction audit	Post-outcome read-only	Mechanism consistency
Early learned-router qualification	Failed prospective qualification	Negative routing-method evidence
Search symbolic-router qualification	Prospective held-out	Base routing opportunity
Search production-interface audit	Post-outcome Base-only	Interface-alignment audit
Search trained-endpoint opportunity audit	Post-outcome archive recovery	Routing headroom after training
Search eight-pair S/E	Prospectively frozen	Training-procedure comparison
Search Hybrid	Prespecified secondary	Attribution diagnostic
Search selector concentration	Post-outcome	Behavioral diagnostic
Search label/order counterbalance	Post-outcome Base-only robustness	Label/order confound audit
Cross-setting comparison	Exploratory	Descriptive comparison
EF-RLVR objective 2×2 ablation	Post-outcome; frozen before ablation outcomes	Objective-component localization
Direct selector-off vs. executor-up contrast	Added after factorial outcomes	Supplementary objective-modification comparison
Search Uniform-S vs Original-S	Prospectively frozen paired follow-up	Selector-objective intervention test
Search Uniform/Original routing-headroom analysis	Prespecified secondary of the paired follow-up	Post-training selection opportunity