CAGE: BLOCKWISE POLICY GRADIENTS FOR EXPLICITLY ROUTED LANGUAGE-MODEL WORKFLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning from terminal rewards in routed language-model workflows requires two distinct comparisons: between routes to learn which workflow to select, and within routes to learn how to execute it. Small execution budgets make both comparisons difficult: independent sampling can omit routes, while global centering mixes route-level reward differences into execution credit. We introduce Credit Assignment with Graph-aware Estimation (CAGE), which allocates executions across routes before observing rewards. It sums the Router gradient over the finite route set and uses a route-local leave-one-out baseline for the Specialist. Live-policy weights correct for the allocated sample counts. Under independent conditional sampling and sampling-consistent scores, both raw estimators are conditionally unbiased for the original current-policy objective. We establish deterministic route coverage, invariance of Specialist credit to route-specific reward offsets, and removal of the additional categorical sampling variance of a same-batch sampled-Router estimator. An exact covariance decomposition quantifies the noise added by estimating the Specialist's local baselines from finite samples. A fixed-policy diagnostic confirms the coverage mechanism: IID sampling misses routes on 29 of 32 questions, whereas CAGE covers every route. A one-question full-LoRA-gradient check on a frozen 4B model also verifies samplewise route-shift invariance. These diagnostics support the coverage and local-credit mechanisms; an end-to-end reward advantage remains unestablished. CAGE estimates both route-selection and route-execution gradients from the same fixed complete-workflow budget while preserving the current-policy objective.

1 INTRODUCTION

Many language-model systems organize inference as a routed workflow: a Router selects how a request should be processed, one or more Specialists execute the selected route, and a verifier supplies an end-to-end outcome. Learned routing can trade response quality against inference cost (Ong et al., 2025), while layered multi-model systems demonstrate the value of composing heterogeneous language models (Wang et al., 2025b). These systems expose a known computational graph. A common reinforcement-learning estimator, however, treats its complete executions as exchangeable.

An IID/global-RLOO update samples complete workflows independently and centers them against a global leave-one-out baseline. Under sampling-consistent scores, it is unbiased for the routed objective. Our motivation is its finite-batch behavior, not a defect in its expected gradient. First, the Router controls which Specialist trajectories are observed. With a small fixed workflow budget, independent routing produces random route counts and can leave a route entirely unobserved during an update. Second, a global leave-one-out baseline centers a Specialist trajectory using rewards from other routes, even when those routes have different depths or conditional reward distributions. The residual then contains between-route variation unrelated to the Specialist action receiving credit. The two policy blocks therefore need different comparisons: the Router must retain differences between route values, whereas Specialist credit can remove route-level offsets. Applying route-local centering to both blocks indiscriminately would erase the route-selection signal.

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.

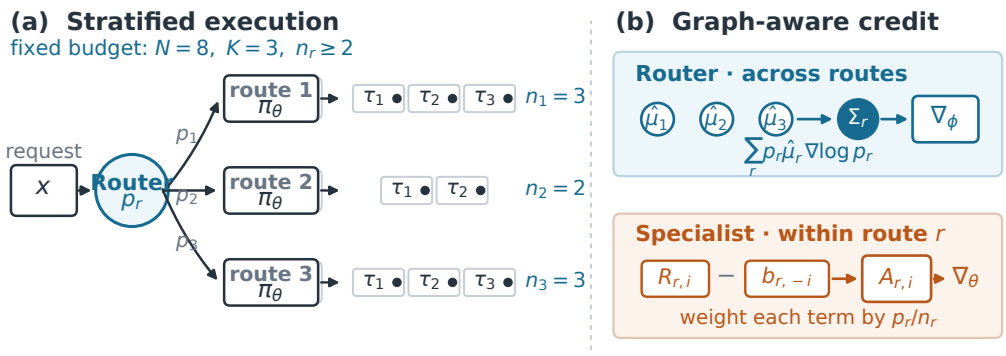

Figure 1: CAGE separates the finite Router action from stochastic Specialist execution. It allocates the fixed workflow budget across all routes, forms the Router gradient by exactly summing route-conditional reward means, and centers each Specialist trajectory only against executions of the same route. The p_r/n_r weight preserves the current-policy objective under forced coverage; the model, reward, and total number of executions are unchanged.

How can a fixed execution budget supply usable local comparisons without changing the routed objective? CAGE addresses this question by allocating at least two executions to each route before observing rewards. The Router uses all route means in a finite-action sum; the Specialist compares each trajectory with other executions of the same route. Weighting each route by its live probability corrects for the allocated sample counts. This changes how the budget is distributed and how credit is assigned, while keeping the workflow, reward, models, and total number of executions fixed.

The distinction is intuitive even in a two-route example. Suppose route A has mean reward 0.9, while two executions of route B obtain 0.25 and 0.15. The Specialist should distinguish the two B executions; their local LOO advantages are $+0.10$ and -0.10, regardless of A's reward. The Router, in contrast, should retain the difference between A and B. A route-level reward offset is therefore removable from Specialist credit but informative for routing. CAGE assigns these two comparisons to separate gradient blocks.

We analyze what this separation guarantees and what it costs at small sample counts. Fixed-policy diagnostics then test coverage and the Specialist's response to route-specific reward offsets. In the coverage check, IID routing misses at least one route on 29 of 32 questions, whereas CAGE covers every route. Mathematical-reasoning evaluations examine downstream reward, with implementation limits described in Section 6.3; they do not establish an end-to-end advantage.

Our contributions are:

1. We derive separate Router and Specialist estimators that use the same covered batch and preserve the original current-policy objective under the stated sampling assumptions.
2. We prove conditional unbiasedness, route-shift invariance, and coverage. The variance analysis identifies both the Router noise removed by exact summation and the finite-sample cost of estimating Specialist local baselines.
3. We test coverage and route-shift invariance with fixed-policy and full Specialist LoRA-gradient diagnostics. Budget sensitivity, four-arm validation, and a three-seed external test distinguish these mechanisms from reward gains.

2 Related Work

Routed and multi-agent language-model systems. RouteLLM learns model routing from preferences (Ong et al., 2025); Mixture-of-Agents composes layered model outputs (Wang et al., 2025b); and MasRouter and Router-R1 learn collaboration or multi-round routing (Yue et al., 2025; Zhang et al., 2025). MARFT formulates heterogeneous multi-agent fine-tuning (Liao et al., 2025), while Agent Lightning separates execution from RL training (Luo et al., 2025). Stronger-MAS's AT-

GRPO tree-samples identical agent–turn contexts (Zhao et al., 2026). CAGE instead keeps the workflow fixed and estimates its end-to-end current-policy gradient from a fixed complete-execution budget.

Policy gradients and group-relative baselines. REINFORCE is the standard score-function estimator (Williams, 1992); critic-free LM variants revisit leave-one-out baselines (Ahmadian et al., 2024), and GRPO uses group-relative advantages (Shao et al., 2024). BPO branches intermediate sandbox states and centers sibling returns (He et al., 2026). CAGE retains independent complete executions, groups by known entry route, and corrects forced sampling by p_r/n_r.

Stratified GRPO is the closest local-centering precedent: it post-hoc partitions i.i.d. trajectories by endogenous structure and blends self-inclusive, within-stratum standardization with global normalization (Zhu et al., 2025). CAGE instead preallocates explicit routes, uses p_r/n_r-weighted Specialist LOO, and exactly estimates a separate finite Router block. Local normalization is therefore prior art, but not the complete routed-objective estimator.

Multi-agent credit assignment. COMA uses a centralized counterfactual critic (Foerster et al., 2018), and learning-aware gradients model other agents' updates (Meulemans et al., 2025). SSVPO uses sequential Shapley credit (Li et al., 2026); C3 replays alternative messages from restored contexts (Chen et al., 2026); and CollabUIAgents and LangMARL add process rewards or dense language feedback (He et al., 2025; Yao et al., 2026). CAGE fits no critic and adds no reward: it compares independent same-route executions and sums the Router. Ben-Gal & Tong (2026) derive marginal-contribution training signals under filtered feedback, including routing where only the selected candidate's reward is observed. CAGE observes every allocated run and needs no learned counterfactual contribution model.

Stratification and marginalization. Stratified stochastic-gradient sampling (Lu et al., 2021), Rao–Blackwellization (Casella & Robert, 1996), and expected policy gradients (Ciosek & Whiteson, 2018) establish the component principles. CAGE's contribution is their graph-specific coupling with route-local credit and fixed-budget coverage, not any primitive alone.

3 Problem Formulation

3.1 Routed workflow model

Let $x \sim \mathcal{D}$ denote a request. A trainable Router $\pi_\phi(r \mid x)$ selects one of K routes. Conditional on r, an independently parameterized Specialist policy π_θ generates language actions along the route-specific graph, yielding $\tau \sim \pi_\theta(\cdot \mid x, r)$. A fixed answer reader and verifier produce a terminal reward $R(x, r, \tau)$. In the main setting, $K = 3$ and the routes contain one, two, and three Specialist nodes. The estimator requires only that the Router action be finite and enumerable; a downstream route may contain an arbitrary fixed sequence or graph of stochastic language actions. The implementation uses one Specialist parameter block shared across its node invocations, with node- and history-specific contexts; Router parameters ϕ and Specialist parameters θ are disjoint. Operationally, the Router distribution is the softmax of the next-token logits restricted to three distinct single-token route labels; it is not obtained by free-form text parsing.

The graph information used by CAGE is deliberately limited. Each Router action indexes an executable subgraph, and the resulting route label defines a sampling stratum; within that subgraph, the Specialist score sums actions over all visited nodes. CAGE does not infer node-level causal contributions or require the environment to replay a counterfactual prefix. Its claims are therefore about route-level sampling and policy-gradient credit, not arbitrary credit assignment on every internal edge.

Writing $p_r = \pi_\phi(r \mid x)$ and $\mu_r = \mathbb{E}[R \mid x, r]$, the per-request objective is

$$J_x(\phi, \theta) = \sum_{r=1}^{K} p_r \mu_r. \tag{1}$$

The two score-function gradients have different statistical structure:

$$\nabla_\phi J_x = \sum_r p_r \mu_r \nabla_\phi \log p_r, \tag{2}$$

$$\nabla_\theta J_x = \sum_r p_r \mathbb{E}[R \nabla_\theta \log \pi_\theta(\tau \mid x, r)]\,. \tag{3}$$

The first sum contains only K enumerable structural actions. The second contains high-dimensional stochastic language trajectories within each route. Throughout, *credit assignment* means the stochastic advantage that multiplies each policy's score term in these estimators; we do not claim to recover latent intermediate rewards or causal contribution scores.

3.2 Two finite-budget mismatches

Suppose an update can afford N complete workflows for a request. Under independent routing, realized route counts are random even though their expectations follow p_r. Router estimation therefore mixes uncertainty about which route was sampled with uncertainty about how that route executed. Route r is entirely absent with probability $(1 - p_r)^N$.

A global leave-one-out baseline creates a separate finite-batch issue. It compares a trajectory with samples from routes that can have different reward means or graph depths. The resulting residual contains between-route variation unrelated to the current Specialist action. These observations motivate treating the finite Router action and within-route language actions separately; they do not imply bias or universal variance inferiority of global RLOO.

4 CAGE

4.1 Live-policy stratified allocation

For a fixed budget $N \geq 2K$, CAGE assigns at least two executions to every route and distributes the remaining budget according to the live Router:

$$n_r = 2 + \mathrm{LargestRemainder}((N - 2K)p_r)\,, \qquad \sum_r n_r = N. \tag{4}$$

Allocation occurs before observing target-batch rewards. In the primary study, $N = 8$ and $K = 3$. The coverage floor modifies proportional allocation for a low-probability route; the gradient weights below correct for this change and retain the objective in Equation 1. The integer counts are stop-gradient sampling decisions. Two is the smallest floor that gives every trajectory another same-route sample for LOO; it is a feasibility constraint, not a variance-optimal allocation. With $N = 8$ and $K = 3$, only two residual executions track p_r.

4.2 Exact Router marginalization

For route r, let $\widehat{\mu}_r = n_r^{-1} \sum_{i=1}^{n_r} R_{r,i}$. Rather than sampling the discrete Router action, CAGE uses

$$\widehat{g}_\phi^{\mathrm{CAGE}} = \sum_r p_r \widehat{\mu}_r \nabla_\phi \log p_r. \tag{5}$$

The sum is exact over routes; only each route's conditional reward is estimated by Monte Carlo. A common baseline could be subtracted without changing the estimator because $\sum_r p_r \nabla_\phi \log p_r = 0$.

4.3 Route-local weighted RLOO

For trajectory i on route r, define

$$b_{r,-i} = \frac{1}{n_r - 1} \sum_{j \neq i} R_{r,j}. \tag{6}$$

The Specialist estimator is

$$\widehat{g}_\theta^{\mathrm{CAGE}} = \sum_r \frac{p_r}{n_r} \sum_{i=1}^{n_r} (R_{r,i} - b_{r,-i})\, \nabla_\theta \log \pi_\theta(\tau_{r,i} \mid x, r). \tag{7}$$

Samples are centered only against executions of the same route. The p_r/n_r factor preserves the live-policy objective under forced allocation. Trajectory log probability sums generated tokens at every Specialist node on the selected route and includes the terminal EOS action.

Why the components are coupled. Coverage makes every local LOO comparison well-defined and supplies a reward estimate for every Router action. Weighting then restores the desired mixture: an unweighted average over the N allocated executions has expected Specialist update $\sum_r (n_r/N)\nabla_\theta \mu_r$, rather than $\sum_r p_r \nabla_\theta \mu_r$. Finally, a separate Router estimator retains between-route value differences that local centering would remove; exact summation uses all available route means without additional label draws. This construction links coverage to the two credit targets. Alternatives include IID sampling with fallback baselines and other reward-independent covered allocations.

Why local centering cannot replace Router credit. For disjoint Router and Specialist blocks, population route centering gives

$$\mathbb{E}[(R - \mu_r)\nabla_\phi \log p_\phi(r \mid x) \mid r] = 0. \tag{8}$$

Pure local centering therefore removes route selection; a global/local blend retains only a sampled global component. Self-inclusive empirical mean centering alone shrinks the conditional Specialist estimate by $1-1/n_r$, whereas LOO does not. CAGE centers only the Specialist block and uses Equation 5 for route selection. The identity concerns population centering; finite-sample SAN/global blends need not be equivalent to it.

For example, consider two equally likely routes with deterministic rewards $\mu_1 = 0$ and $\mu_2 = 1$. Every route-local residual is zero, correctly leaving no Specialist learning signal. Yet for Router softmax logits z, the objective gradient is $\nabla_z J_x = (-1/4, 1/4)$: routing should move toward route 2. Equation 5 retains this signal; centering the Router within each route would erase it.

4.4 Implementation and complexity

The estimator is evaluated as weighted negative-log-probability losses, which is algebraically equivalent to materializing and summing trajectory score gradients but permits length-bucketed micro-batching. Concretely, all coefficients are computed from the pre-update policy and rewards and treated as stop-gradient quantities:

$$\mathcal{L}_\phi = -\sum_r \mathrm{sg}[p_r \widehat{\mu}_r] \log p_r,$$

$$\mathcal{L}_\theta = -\sum_r \sum_i \mathrm{sg}\left[\frac{p_r}{n_r}(R_{r,i} - b_{r,-i})\right] \log \pi_\theta(\tau_{r,i} \mid x, r).$$

Backpropagating through a coefficient would define a different surrogate and is not part of CAGE. Conditional unbiasedness concerns these raw estimators before optimizer preconditioning and gradient clipping; an AdamW update need not itself be an unbiased estimate of gradient descent. Per request, CAGE requires K Router scores and the same N complete workflow executions as the baseline. Estimator-side bookkeeping is $O(K + N)$; generation remains the dominant cost. Algorithmic details appear in Appendix A.

5 Theoretical Analysis

All statements condition on x, the current parameters, and an allocation chosen before rewards are observed. Executable routes have $p_r > 0$, as under a finite softmax Router; trajectories are sampled independently within each route, and standard score-function regularity conditions hold. Scores must differentiate the distribution that generated the trajectories. Numerical actor–trainer agreement is a separate implementation requirement.

Proposition 1 (Router unbiasedness). The CAGE Router estimator is conditionally unbiased for Equation 2. Since $\mathbb{E}[\widehat{\mu}_r \mid x, r] = \mu_r$,

$$\mathbb{E}[\widehat{g}_\phi^{\mathrm{CAGE}} \mid x] = \sum_r p_r \mu_r \nabla_\phi \log p_r = \nabla_\phi J_x. \tag{9}$$

The coverage floor changes the precision of each conditional mean, not the expectation of the weighted sum.

Proposition 2 (Specialist unbiasedness). The CAGE Specialist estimator is conditionally unbiased for Equation 3. The leave-one-out baseline $b_{r,-i}$ depends only on other independently sampled trajectories. It is independent of the current score conditional on (x, r), and

$$\mathbb{E}[b_{r,-i}\nabla_\theta \log \pi_\theta(\tau_{r,i} \mid x, r) \mid x, r] = 0. \tag{10}$$

Taking expectations of Equation 7 recovers the Specialist block of the current-policy gradient. A complete derivation is in Appendix B.

Proposition 3 (Specialist route-shift invariance). Let $R'_{r,i} = R_{r,i} + c_r$ for arbitrary constants that may differ across routes. The true Specialist gradient is unchanged because $\mathbb{E}[c_r \nabla_\theta \log \pi_\theta(\tau \mid x, r)] = 0$. Every CAGE Specialist advantage is unchanged sample by sample:

$$(R_{r,i} + c_r) - \frac{1}{n_r - 1}\sum_{j \neq i}(R_{r,j} + c_r) = R_{r,i} - b_{r,-i}. \tag{11}$$

By contrast, a global leave-one-out advantage over N preallocated trajectories changes by $(Nc_r - \sum_s n_s c_s)/(N-1)$, which is generally nonzero. Route-local centering therefore excludes route-only reward offsets from finite-sample Specialist credit without asserting uniform variance dominance. These route shifts remain value signal for the separately estimated Router block and are not removed there.

Proposition 4 (Finite-budget coverage). For $N \geq 2K$, Equation 4 gives at least two trajectories per route and exactly N trajectories in total. This is a construction-level guarantee. Whether allocating samples to a low-probability route improves learning is an empirical question.

Proposition 5 (Marginalization versus the sampled-Router ablation). Fix x and the policy parameters. Let $\mathcal{B}$ be the covered batch, and let the sampled-Router arm draw M independent labels $u_j \sim p$ conditional on $\mathcal{B}$, reusing its route means: $\widehat{g}_\phi^{\text{sampled}} = M^{-1}\sum_j \widehat{\mu}_{u_j} s_{u_j}$, where $s_r = \nabla_\phi \log p_r$. Its conditional expectation is exactly $\widehat{g}_\phi^{\text{CAGE}}$. Writing $V_a = \operatorname{tr}\operatorname{Var}(\widehat{g}_\phi^a \mid x)$, total variance gives

$$V_{\text{sampled}} - V_{\text{CAGE}} = \frac{1}{M}\mathbb{E}_{\mathcal{B}}[\operatorname{tr}\operatorname{Var}_{u\sim p}(\widehat{\mu}_u s_u \mid \mathcal{B}, x)|x] \geq 0. \tag{12}$$

This comparison holds for the actual integer allocation, including its floor. It isolates extra label-sampling noise at the same policy and batch law, not variance across different trained checkpoints or improvement in final reward. Appendix B.4 gives the proof.

Comparison with IID. Against IID routing with a fixed common baseline, exact proportional allocation removes a nonnegative route-choice variance term (Appendix B.5). CAGE's integer floor changes that allocation, so this is not a universal ordering against IID/global RLOO. The floor trades precision on frequent routes for usable samples on rare ones.

Specialist variance and the cost of local baselines. The two-sample floor ensures feasibility, but how much noise does estimating a local baseline add? Write $S_r = \nabla_\theta \log \pi_\theta(\tau \mid x, r)$, $Z_r = R - \mu_r$, $\sigma_r^2 = \operatorname{Var}(R \mid x, r)$, $g_r = \mathbb{E}[Z_r S_r]$, $F_r = \mathbb{E}[S_r S_r^\top]$, and $A_r = \operatorname{Cov}(Z_r S_r)$. With independent samples across routes and finite indicated moments, the sample-covariance identity gives

$$\operatorname{Cov}(\widehat{g}_\theta^{\text{CAGE}} \mid x) = \sum_r p_r^2 \left[\frac{A_r}{n_r} + \frac{\sigma_r^2 F_r + g_r g_r^\top}{n_r(n_r - 1)}\right]. \tag{13}$$

The first term uses the population route mean as baseline; the second is the positive-semidefinite cost of estimating that mean by LOO. Writing $B_r = \sigma_r^2 F_r + g_r g_r^\top$, the unweighted route covariance is $A_r/2 + B_r/2$ at $n_r = 2$, versus $A_r/3 + B_r/6$ at $n_r = 3$. These coefficients quantify the sample-count dependence, not the absolute noise: that also depends on the reward–score moments A_r, B_r. Both matrices are invariant to route-specific reward offsets (Appendix B.3).

6 EXPERIMENTAL SETUP

6.1 DATA, MODELS, WORKFLOW, AND REWARD

The empirical evidence comprises HardMath development evaluations and a frozen external Omni-MATH test. As in verifier-scored mathematical reasoning (Hendrycks et al., 2021), terminal answers provide the learning signal.

The primary training source is the public SparkleRL HardMath corpus (Wang et al., 2025a). Before paired-method outcomes are observed, preprocessing rejects exact duplicate questions, missing rationales, and rows whose reference answer and public solution disagree under the same bidirectional symbolic verifier used for reward. This leaves 4,361 automatically screened candidates from 6,501 raw rows. This filter does not establish semantic correctness: later inspection found erroneous references, incomplete questions, and multipart-target mismatches in the frozen training data, which were not repaired in the reported runs. A frozen base-policy calibration motivates HardMath as a nonsaturated training regime; it is not a guarantee of learnability for each selected question. The frozen V2 development populations contain 320 training questions and 128 disjoint validation questions. Stage A and Stage B reuse these populations; Stage B varies training seeds and question permutations, not the development questions. The confirmatory design uses ten 32-question training batches. The external test population is a deterministic 500-question Omni-MATH subset (Gao et al., 2025) with difficulty at least five, proportional difficulty coverage, symbolically consistent labels, and no exact HardMath overlap. Its selection is frozen before trained checkpoints are evaluated.

The Router and Specialist use disjoint Qwen3.5-4B policy parameters (Qwen Team, 2026) initialized from the same frozen base checkpoint. Each has a rank-16 LoRA adapter with alpha 32, zero dropout, and adapters on all language-transformer linear projections. The Router selects among routes containing one, two, or three node invocations of the shared Specialist policy, followed by an identical frozen answer reader. Reward is exact symbolic equivalence under `math-verify` 0.9.0 (Kydlíček, 2024).

6.2 EQUAL-WORKFLOW TRAINING AND EVALUATION

The primary baseline samples eight routes independently from the same live Router and uses one global leave-one-out baseline. CAGE and the baseline share paired initial weights, question order, workflow count, sampling temperature, action horizon, and one update per batch. Equal complete-workflow count does not imply identical node calls or generated tokens when route depths differ, so both are reported explicitly and this is not described as equal compute. Stage A screened $\{2 \times 10^{-6}, 5 \times 10^{-6}\}$ for both methods on one tuning seed, selecting by each method's own validation reward, never the cross-method contrast. Reuse of existing 2×10^{-6} runs follows a post-outcome amendment (Appendix C.1); Stage-A outcomes are excluded from the primary effect. The completed grid selected 2×10^{-6} for both methods (Appendix C.1). The Stage-B design uses three new paired seeds and ten training batches per run, with nine nonzero updates. It covers CAGE and IID only; frozen-base and single-agent controls were not evaluated.

Both regimes sample at temperature 0.7 and top-p 0.95, disable explicit thinking mode, and retain solution tokens plus EOS under a 16,384-token action safety cap. We optimize both adapters with AdamW, zero weight decay, a two-step linear warmup followed by cosine decay to zero, and global gradient clipping at 1.0, without PPO ratio clipping or a reference-KL penalty. The implementation reconstructs temperature/top-p scores; its departure from the sampling distribution and the scope of the resulting comparisons are specified in Section 6.3.

Checkpoint selection uses only the frozen 128-question validation population with earliest-step tie breaking. Selected and final checkpoints are reported separately. After all arms and seeds completed, each selected policy was evaluated on the 500-question Omni-MATH population by drawing eight routes IID from that checkpoint's live Router and executing the corresponding workflows. Thus, for both training methods, $Y_{m,s,q} = 8^{-1} \sum_{j=1}^{8} R_{m,s,q,j}$ estimates the same natural current-policy workflow value.

6.3 STATISTICS AND EMPIRICAL SCOPE

For training seed s and held-out question q, let $Y_{m,s,q}$ be the mean terminal reward over eight workflows. We report the three paired effects $\Delta_s = Q^{-1}\sum_q (Y_{\mathrm{CAGE},s,q} - Y_{\mathrm{IID},s,q})$, their mean, and a paired question-bootstrap interval. Resampling keeps methods and all three training seeds together for each question. The interval is conditional on these trained seeds, not an estimate of training-seed population uncertainty.

We distinguish fixed-policy mechanism checks from training-based comparisons. The former test coverage and samplewise parameter-gradient sensitivity. For the latter, every historical Stage-B run used target-dependent support repair when a sampled token fell outside the reconstructed nucleus, and the fused Router also exhibited cross-process reconstruction discrepancies. Consequently, these runs do not establish sampling-consistent execution of the analyzed estimator. We report their rewards as implementation-limited comparisons, not as a test of CAGE's theoretical unbiasedness or definitive learning efficacy. Appendix C.1 gives the numerical scope and follow-up checks. Workflow, node, and token counts characterize execution cost; parameter-gradient dispersion and total training-compute efficiency were not measured.

7 RESULTS

7.1 COVERAGE UNDER A SMALL WORKFLOW BUDGET

We first test the sampling mechanism at a common initial policy on 32 fixed training questions with $K = 3$ routes and $N = 8$ workflows per question. IID sampling leaves at least one route absent on 29 questions and a singleton route on seven; CAGE has neither failure. The low-probability route B is absent on 28 IID questions and appears once on the remaining four, whereas CAGE allocates two executions to it on every question. Thus the missing route-local comparison under IID becomes available on every CAGE question.

Table 1: Analytic coverage sensitivity at the same 32 fixed Router distributions. Entries count expected empty or singleton question–route cells among 96 cells. CAGE guarantees zero of both for $N \geq 2K$. Only $N = 8$ was executed in this diagnostic; the other columns are analytic, not additional training runs.

Workflows per question N	6	8	12	16	24
IID: expected empty cells	31.28	27.97	23.96	21.33	17.62
IID: expected singleton cells	14.27	11.81	9.81	9.35	9.50
CAGE: empty / singleton cells	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0

Table 1 evaluates $\sum_{x,r}(1-p_r(x))^N$ and $\sum_{x,r} N p_r(x)(1-p_r(x))^{N-1}$ without new rollouts. Even at $N = 24$, IID has 17.62 expected empty cells: increasing the budget alone does not ensure coverage of rare routes. The floor addresses that failure directly, at the cost of diverting samples from high-probability routes. Execution accounting and the distinction between coverage and answer quality appear in Appendix F.

Actual parameter-gradient check. On one runtime-selected question, we rescore eight independently generated saved workflows per arm at a frozen 4B Specialist, including all 32,464,896 LoRA parameters. Adding route offsets $(-1, 0, +1)$ leaves the CAGE gradient exactly unchanged; the IID/global-RLOO change has norm 5.98 times its original gradient norm. This verifies Equation 11 in parameter space for this batch, not lower gradient variance or improved reward. Appendix F.1 details the post-observation question choice and full-gradient calculation; no generation or training was added.

7.2 DEVELOPMENT-SET COMPARISONS

Before held-out access, we evaluated final step-10 checkpoints on 128 frozen HardMath questions with paired route uniforms. CAGE, its sampled-Router ablation, IID, and SAN obtained .45605, .45312, .44824, and .44531; every paired interval included zero. The sampled-Router contrast had

20 wins, 20 losses, and 88 ties, so it does not establish a reward benefit or equivalence. Full intervals, audit details, and compute appear in Appendix E.

Both methods selected a learning rate of 2×10^{-6} in Stage A. Stage B completed three paired seeds and 60 evaluations. CAGE's mean reward is .46842 versus .46257 at each run's selected checkpoint, and .45964 versus .44889 at fixed step ten; all three seed differences are positive under both rules (Appendix E). Reusing these questions for rate and checkpoint selection makes the external test the relevant transfer check.

7.3 Frozen external-test comparison

All selected and final checkpoint pairs were evaluated on the same frozen 500 Omni-MATH questions, using eight live-Router IID workflows per question and paired seed domains. Table 2 reports the prespecified selected-checkpoint analysis and fixed-final sensitivity analysis, with the seed-conditional paired bootstrap defined in Section 6.3.

Table 2: Frozen external-test reward. Differences are CAGE minus IID in percentage points (pp). The selected row is primary; the final row is a prespecified sensitivity analysis. Historical training used the disclosed support-repair scorer, so these are not repaired-run results.

Checkpoint rule	IID	CAGE	Δ (pp)	Paired 95% interval (pp)
Validation-selected	.50800	.50900	+0.100	$[-0.592, +0.808]$
Fixed final	.51217	.50992	-0.225	$[-0.900, +0.425]$

The selected-checkpoint seed effects are $+0.775$, $+0.450$, and -0.925 pp for seeds 88501–88503. Thus the small positive aggregate is neither uniform across seeds nor distinguishable from zero under the conditional interval. At fixed final checkpoints the corresponding effects are -0.550, $+0.175$, and -0.300 pp. The external test therefore does not establish a CAGE reward advantage, despite the consistent direction on the reused validation set.

8 Limitations

The analysis covers finite K with $N \geq 2K$; empirical coverage is limited to $K = 3$. CAGE assumes an enumerable Router; exact summation is costly for many graphs or token-level routing, and its two-sample floor can waste budget on low-probability redundant routes. It neither discovers workflows nor adds intermediate credit, and may help little when route values are similar, rewards are saturated, or IID coverage is reliable. Evidence is limited to verifier-scored mathematics; open-ended tasks add judge and environment noise. The implementation constraints in Section 6.3 leave multi-step efficacy under sampling-consistent execution unresolved; the available recovery checks do not close that gap. Training data also contain the semantic defects described in Section 6. Neither the coverage guarantee nor route-shift invariance implies a universal reduction in total gradient variance: allocation must balance route coverage against the precision of high-probability route estimates.

9 Conclusion

CAGE separates between-route Router credit from within-route Specialist credit while preserving the current-policy objective. The analysis establishes conditional unbiasedness, coverage, and route-shift invariance, and quantifies the finite-sample cost of local baselines. Coverage measurements and a one-question full-LoRA-gradient check support the sampling and local-credit mechanisms. Translating these properties into reliable end-to-end gains remains an empirical question.

Reproducibility statement

Sections 3–6 and the appendix specify the estimators, optimization, data, selection, evaluation, and estimand. Proofs and implementation details appear in the appendix; the anonymous supplement provides an aggregate evidence summary and checksums. Full execution journals are not included.

AI use statement

Generative AI tools, including OpenAI Codex, were used to identify and review related literature, provide feedback on research methodology and experiments, implement and test software, diagnose execution failures, analyze and interpret results, prepare figures, and draft and polish the manuscript. They also assisted in formulating mathematical claims and developing and checking proofs, including the Specialist covariance decomposition. Numerical results were checked against frozen artifacts or independently reproducible calculations. The authors remain responsible for reviewing the AI-assisted work and for all content, claims, and artifacts.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

Elai Ben-Gal and Stela Tong. Marginal-contribution policy gradients under filtered feedback for multi-agent LLMs, 2026. URL https://arxiv.org/abs/2604.22785v2. Version 2.

George Casella and Christian P. Robert. Rao–blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996. doi: 10.1093/biomet/83.1.81.

Yanjun Chen, Yirong Sun, Hanlin Wang, Jinghan Wang, Xinming Zhang, Xiaoyu Shen, Wenjie Li, and Wei Zhang. Exact is easier: Credit assignment for cooperative LLM agents, 2026. URL https://arxiv.org/abs/2603.06859.

Kamil Ciosek and Shimon Whiteson. Expected policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11607. URL https://ojs.aaai.org/index.php/AAAI/article/view/11607.

Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11794.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f9e1e8b56c7e363985ebeb0e9dd1a85c-Abstract-Conference.html.

Bowei He, Yankai Chen, Xiaokun Zhang, and Xue Liu. Branching policy optimization: Sandbox-native language agent reinforcement learning, 2026. URL https://arxiv.org/abs/2607.14171.

Zhitao He, Zijun Liu, Peng Li, Yi R. Fung, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. Advancing language multi-agent learning with credit re-assignment for interactive environment generalization. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=SoEmgM1ioC.

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. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2103.03874.

Hynek Kydlíček. Math-Verify: Math verification library, 2024. URL https://github.com/huggingface/Math-Verify. Version 0.9.0.

Yugu Li, Zehong Cao, Jianglin Qiao, and Siyi Hu. SSVPO: Effective step-level credit assignment for RL training of language models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4da4dd613cc9e63932d643b9c6e8f86f-Abstract-Conference.html.

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. MARFT: Multi-agent reinforcement fine-tuning, 2025. URL https://arxiv.org/abs/2504.16129.

Yucheng Lu, Youngsuk Park, Lifan Chen, Yuyang Wang, Christopher De Sa, and Dean Foster. Variance reduced training with stratified sampling for forecasting models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7145–7155, 2021. URL https://proceedings.mlr.press/v139/lu21d.html.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent lightning: Train any ai agents with reinforcement learning, 2025. URL https://arxiv.org/abs/2508.03680.

Alexander Meulemans, Seijin Kobayashi, Johannes von Oswald, Nino Scherrer, Eric Elmoznino, Blake A. Richards, Guillaume Lajoie, Blaise Aguera y Arcas, and Joao Sacramento. Multi-agent cooperation through learning-aware policy gradients. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/718a3e764e63a130323c5b5abf4ac332-Abstract-Conference.html.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5503a7c69d48a2f86fc00b3dc09de686-Abstract-Conference.html.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5. Model card: https://huggingface.co/Qwen/Qwen3.5-4B.

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, 2024. URL https://arxiv.org/abs/2402.03300.

Jiayu Wang, Yifei Ming, Zixuan Ke, Caiming Xiong, Shafiq Joty, Aws Albarghouthi, and Frederic Sala. Beyond accuracy: Dissecting mathematical reasoning for LLMs under reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-2235. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6058d0c628a03fd95dfe5c72cbdf9e64-Abstract-Conference.html.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y. Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5434be94e82c54327bb9dcaf7fca52b6-Abstract-Conference.html.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

Huaiyuan Yao, Longchao Da, Xiaoou Liu, Charles Fleming, Tianlong Chen, and Hua Wei. LangMARL: Natural language multi-agent reinforcement learning, 2026. URL https://arxiv.org/abs/2604.00722.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. MasRouter: Learning to route LLMs for multi-agent system. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15549–15572, 2025. doi: 10.18653/v1/2025.acl-long.757. URL https://aclanthology.org/2025.acl-long.757/.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-R1: Teaching LLMs multi-round routing and aggregation via reinforcement learning, 2025. URL https://arxiv.org/abs/2506.09033.

Yujie Zhao, Lanxiang Hu, Yang Wang, Minmin Hou, Hao Zhang, Ke Ding, and Jishen Zhao. Stronger-mas: Multi-agent reinforcement learning for collaborative LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/f3da4165893c2465fd7e8df453c41ffa-Abstract-Conference.html.

Mingkang Zhu, Xi Chen, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Stratified GRPO: Handling structural heterogeneity in reinforcement learning of LLM search agents, 2025. URL https://arxiv.org/abs/2510.06214.

A PER-REQUEST CAGE ALGORITHM

For each request x:

1. Evaluate $p_r = \pi_\phi(r \mid x)$ for all $r \in \{1, \dots, K\}$.
2. Allocate $n_1, \dots, n_K$ with Equation 4.
3. Independently sample n_r trajectories on every route and record their terminal rewards.
4. Compute the Router estimate in Equation 5 from per-route sample means.
5. Compute the Specialist estimate in Equation 7 from route-local leave-one-out advantages and weights p_r/n_r.

Largest remainder first takes the floor of each residual quota $(N-2K)p_r$, then assigns the remaining integer units to routes in descending order of fractional remainder under a fixed tie-breaking rule. Counts therefore sum to N exactly and are functions only of the live Router probabilities.

B PROOF DETAILS

B.1 ROUTER BLOCK

Conditioning on x and the reward-independent allocation,

$$\mathbb{E}[\widehat{g}_\phi^{\mathrm{CAGE}} \mid x] = \mathbb{E}\left[\sum_r p_r \widehat{\mu}_r \nabla_\phi \log p_r \,\middle|\, x\right] \tag{14}$$

$$= \sum_r p_r \mathbb{E}[\widehat{\mu}_r \mid x, r] \nabla_\phi \log p_r \tag{15}$$

$$= \sum_r p_r \mu_r \nabla_\phi \log p_r. \tag{16}$$

B.2 SPECIALIST BLOCK

For each route and trajectory, $b_{r,-i}$ is measurable with respect to the other independently sampled trajectories. Hence

$$\mathbb{E}[b_{r,-i} \nabla_\theta \log \pi_\theta(\tau_{r,i} \mid x, r) \mid x, r] \tag{17}$$

$$= \mathbb{E}[b_{r,-i} \mid x, r]\, \mathbb{E}[\nabla_\theta \log \pi_\theta(\tau_{r,i} \mid x, r) \mid x, r] = 0, \tag{18}$$

where the final equality is the zero-mean score identity. Therefore,

$$\mathbb{E}[\widehat{g}_\theta^{\mathrm{CAGE}} \mid x] = \sum_r \frac{p_r}{n_r} \sum_{i=1}^{n_r} \mathbb{E}[R_{r,i} \nabla_\theta \log \pi_\theta(\tau_{r,i} \mid x, r)] \tag{19}$$

$$= \sum_r p_r \mathbb{E}[R \nabla_\theta \log \pi_\theta(\tau \mid x, r)]\,. \tag{20}$$

B.3 FINITE-SAMPLE SPECIALIST COVARIANCE

Fix a route and suppress its index. For $n \geq 2$ independent pairs (R_i, S_i), the local LOO estimate is the unbiased sample covariance

$$\widehat{g} = \frac{1}{n} \sum_i (R_i - b_{-i}) S_i = \frac{1}{n-1} \sum_i (R_i - \overline{R})(S_i - \overline{S}) = \binom{n}{2}^{-1} \sum_{i<j} h_{ij}, \tag{21}$$

where $h_{ij} = \frac{1}{2}(R_i - R_j)(S_i - S_j)$. Let $Z = R - \mu$, $g = \mathbb{E}[ZS]$, $F = \mathbb{E}[SS^\top]$, and $A = \mathrm{Cov}(ZS)$; the score identity gives $\mathbb{E}[S] = 0$. The first-order centered component is $h_1(Z, S) = \mathbb{E}[h_{12} \mid Z, S] - g = \frac{1}{2}(ZS - g)$. Direct expansion gives

$$\mathrm{Cov}(h_{12}) = \tfrac{1}{2}\mathbb{E}[Z^2 SS^\top] + \tfrac{1}{2}\sigma^2 F,$$

$$\mathrm{Cov}(h_1) = \tfrac{1}{4} A.$$

The degenerate remainder $h_2 = h_{12} - g - h_1(Z_1, S_1) - h_1(Z_2, S_2)$ has zero conditional mean given either observation. Its covariance is therefore $\mathrm{Cov}(h_2) = \frac{1}{2}(\sigma^2 F + gg^\top)$. Orthogonality of these components and counting the overlapping pairs yield

$$\mathrm{Cov}(\widehat{g}) = \frac{4}{n}\,\mathrm{Cov}(h_1) + \frac{2}{n(n-1)}\,\mathrm{Cov}(h_2) = \frac{A}{n} + \frac{\sigma^2 F + gg^\top}{n(n-1)}. \tag{22}$$

Finally, independent route batches and the weights p_r give Equation 13. The required moments are finite $\mathbb{E}[Z^2\|S\|^2]$, $\mathbb{E}[Z^2]$, and $\mathbb{E}[\|S\|^2]$. The oracle estimator $n^{-1}\sum_i Z_i S_i$ has covariance A/n; LOO pays the additional positive-semidefinite term to avoid knowing μ. This comparison uses the population mean, which need not be the variance-minimizing scalar baseline. For $n = 2$ its coefficient is $1/2$, whereas for larger n it decreases as $1/[n(n-1)]$. A route-constant reward shift leaves Z, and hence every term of this identity, unchanged. This is a finite-sample identity for the analyzed estimator, not a measurement on the trained language model.

B.4 Same-batch sampled-Router comparison

Condition throughout on x and fixed policy parameters. Given $\mathcal{B}$, the route means are fixed and the M labels are independent, so

$$\mathbb{E}[\widehat{g}_\phi^{\mathrm{sampled}} \mid \mathcal{B}, x] = \sum_r p_r \widehat{\mu}_r s_r = \widehat{g}_\phi^{\mathrm{CAGE}},$$
$$\mathrm{Var}(\widehat{g}_\phi^{\mathrm{sampled}} \mid \mathcal{B}, x) = M^{-1}\,\mathrm{Var}_{u\sim p}(\widehat{\mu}_u s_u \mid \mathcal{B}, x).$$

The law of total covariance is the sum of the covariance of the conditional mean and the expected conditional covariance. Taking traces gives Equation 12. Finite second moments suffice for this identity; it needs neither proportional counts nor an unbiased reward model. It establishes a marginalization property, not sampling–score fidelity or empirical learning superiority. In the frozen ablation, $M = 8$.

B.5 Router variance comparison

Let $s_r = \nabla_\phi \log p_r$ and $\sigma_r^2 = \mathrm{Var}(R \mid x, r)$. Independence across preallocated strata gives

$$\mathrm{tr}\,\mathrm{Var}(\widehat{g}_\phi^{\mathrm{CAGE}} \mid x) = \sum_r \frac{p_r^2 \sigma_r^2}{n_r}\|s_r\|_2^2. \tag{23}$$

Under ideal proportional allocation $n_r = Np_r$, the IID estimator with a fixed common baseline c has additional trace variance

$$\frac{1}{N}\,\mathrm{tr}\,\mathrm{Var}_{r\sim p}((\mu_r - c)s_r) \geq 0. \tag{24}$$

The following derivation identifies exactly where proportional allocation is needed.

Let c be any fixed baseline conditional on x, and define $g_\phi = \sum_r p_r \mu_r s_r$. One IID workflow contributes $Y = (R - c)s_r$, so

$$\mathrm{tr}\,\mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^N Y_i \middle| x\right) = \frac{1}{N}\left[\sum_r p_r\left(\sigma_r^2 + (\mu_r - c)^2\right)\|s_r\|_2^2 - \|g_\phi\|_2^2\right]. \tag{25}$$

The common baseline does not change the CAGE estimator because $\sum_r p_r s_r = 0$. If $n_r = Np_r$ is integer for every route, subtracting Equation 23 from Equation 25 gives

$$\frac{1}{N}\left[\sum_r p_r(\mu_r - c)^2\|s_r\|_2^2 - \|g_\phi\|_2^2\right] = \frac{1}{N}\,\mathrm{tr}\,\mathrm{Var}_{r\sim p}((\mu_r - c)s_r)\,, \tag{26}$$

which is nonnegative. The equality concerns a fixed common baseline and exact proportional allocation. It is not a variance ordering for a random global leave-one-out baseline or for the minimum-count allocation in Equation 4.

C REPRODUCIBILITY SPECIFICATION

Table 3 records the staged confirmatory HardMath design. Stage-A optimizer selection and Stage-B three-seed training, validation, checkpoint selection, and frozen external-test aggregation are complete for the historical implementation. A repaired seed-88503 continuation is tracked separately and is not mixed into that aggregate.

Table 3: Confirmatory training and evaluation design.

Item	Specification
Policies	Disjoint Router and Specialist Qwen3.5-4B policy blocks; one Specialist block shared across node invocations; rank-16 LoRA, alpha 32, zero dropout; 32,464,896 trainable parameters per block
Route graph	Three enumerable routes with one, two, or three Specialist nodes; identical frozen answer reader after every route
Training population	320 frozen HardMath development questions, reused across Stages A and B; 10 batches with 9 nonzero updates; 32 questions per batch, once per run
Workflow budget	Eight complete workflows per question; 256 per update; same budget for CAGE and IID/global-RLOO
Generation	Temperature 0.7; top-p 0.95; thinking disabled; solution tokens and terminal EOS are policy actions; 16,384-token action cap
Optimization	AdamW, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, zero weight decay, two-step linear warmup then cosine decay to zero, clip norm 1.0; one on-policy update per batch; no PPO clipping or reference KL
Screened rates	Both CAGE and IID/global-RLOO screen $\{2 \times 10^{-6}, 5 \times 10^{-6}\}$ on the frozen development data; the completed 2×10^{-6} runs are reused under the disclosed amendment; each selects by its own validation reward; both selected 2×10^{-6}; tuning runs excluded from the primary effect
Validation	128 frozen HardMath questions; earliest-step tie break; selected and final checkpoints reported separately
Held-out	500 frozen Omni-MATH questions with difficulty at least five; both methods use eight IID live-Router draws per question with paired seed domains
Reward	Frozen dense Qwen3.5-4B answer reader, at most 128 generated tokens; exact bidirectional symbolic equivalence under `math-verify` 0.9.0

C.1 EXECUTION AND SELECTION PROVENANCE

Sampling consistency and reconstruction. Every historical Stage-B run invoked target-dependent support expansion when rescoring sampled tokens. This prevents identifying the executed updates with the sampling-consistent estimators analyzed in Section 5. A separate repaired one-step workflow passed a frozen numerical-consistency threshold, but its continuation exposed a second issue: the default fused Router produced different probabilities across fresh processes with identical saved inputs. A recurrence-only correction did not resolve reconstruction across all tested recovery paths.

A subsequent reference implementation bound the recurrent, causal-convolution, gated-normalization, and attention operators explicitly. For each arm, all 32 tested Router logits and probabilities matched exactly across three fresh processes and the actual recovery order. A nonzero backward pass and one diagnostic Router-only update also passed, including restoration of the updated adapter and optimizer. These zero-generation checks establish bounded reconstruction consistency, not training efficacy. No complete amended multi-step training result is available for the reported comparison; neither the partial trajectories nor these checks are pooled with the historical three-seed results.

On the accepted repaired step-one checkpoints, a post-observation 32-question four-arm extension obtained rewards .3750 (CAGE), .3711 (IID), .3594 (SAN), and .3945 (uniform-covered allocation with exact Router credit). Its single-seed intervals cross zero, and the strongest mean belongs to the uniform-covered arm. That evaluator predated the Router reference binding and loaded the default fused recurrence, so its route draws are execution-specific rather than evidence for the amended

backend's reconstruction. We retain the numbers as an exploratory implementation-boundary check, not as repaired efficacy or a replacement for the full main experiment.

Stage A evaluated both candidate rates on one tuning seed and the frozen development population. Each method selected its own learning-rate–checkpoint pair by validation reward, breaking ties by earlier step and then lower rate. The existing 2×10^{-6} trajectories are reused under a disclosed post-outcome operational amendment; the 5×10^{-6} trajectories use the same scoring implementation. This is a symmetric grid, not a claim that all reuse decisions preceded outcomes. Stage-A tuning outcomes do not enter the primary effect. Stage B was frozen before selection and uses fresh adapter initializations for three paired seeds, with validation-only checkpoint selection.

All 40 conditions completed evaluation on 128 questions with eight workflows per question (160 shards). Both methods selected a peak rate of 2×10^{-6}: CAGE at step two with mean reward 0.45703, and IID/global RLOO at step eight with mean reward 0.46875. These are maxima over each method's own development grid, not independent estimates of comparative efficacy. The higher selected IID mean does not establish held-out superiority, just as the four-arm fixed-endpoint validation does not establish CAGE superiority. Only the selected rate transfers to Stage B, not the tuned checkpoint or its selected step; each new run selects its checkpoint separately.

The original contract declared linear warmup followed by a constant rate, but every Stage-A candidate executed two warmup steps followed by cosine decay to zero. Step ten had nonzero gradients but zero learning rate and no parameter change. A versioned amendment froze this same executed schedule for Stage B before its first output. Stage A was retained because the deviation was shared across both methods and rates. We report ten training batches and nine nonzero updates, rather than ten effective parameter updates.

The scorer reconstructs temperature-scaled, top-p-truncated and renormalized token probabilities, differentiating with detached nucleus membership. If the local nucleus excludes an actor-sampled token, it expands the support to include that token. In the four-arm matrix this affected 0.02311%, 0.02533%, 0.02482%, and 0.02465% of rescored tokens for IID/global RLOO, SAN/global blend, sampled Router, and CAGE, respectively. The prior 16-token parity probe matched exactly; neither that probe nor observed-token inclusion proves equality of the complete actor and trainer distributions across training. Frequency alone gives no gradient-bias bound. Earlier v1r3 trajectories used unprocessed-softmax scores and are exploratory, not evidence of the analyzed estimator's unbiasedness.

D BASELINE ESTIMATORS

For IID routes $r_i \sim p$ and rewards R_i, the strong baseline uses the global leave-one-out advantage

$$A_i^{\text{global}} = R_i - \frac{1}{N-1} \sum_{j \neq i} R_j. \tag{27}$$

Its Specialist estimate is $N^{-1} \sum_i A_i^{\text{global}} \nabla_\theta \log \pi_\theta(\tau_i \mid x, r_i)$, and its Router estimate uses the same coefficients with $\nabla_\phi \log p_\phi(r_i \mid x)$. This explicit sampled-Router baseline differs from CAGE's all-route sum in Equation 5. The nearest-prior comparison additionally implements post-hoc route strata and the frozen SAN/global blend below.

SAN-style comparator. Let $\bar{R}_r, \sigma_r$ be the self-inclusive within-route mean and population standard deviation; $\bar{R}, \sigma$ are their global counterparts. The coefficient multiplying both the Specialist score and the sampled Router score is

$$c_i = \frac{1}{N} \left[0.8 \frac{R_i - \bar{R}_{r_i}}{\sigma_{r_i} + \epsilon} + 0.2 \frac{R_i - \bar{R}}{\sigma + \epsilon} \right], \qquad \epsilon = 10^{-8}. \tag{28}$$

A singleton has zero local contribution and retains the 0.2-weighted global term. Empty strata contribute no observations. The stabilizer is our frozen implementation convention. This is a route-based adaptation of SAN, not a reproduction of the original search-agent training system.

Sampled-Router ablation. This arm retains CAGE's covered executions and weighted Specialist LOO, but draws $M = 8$ additional categorical route labels independently of their rewards:

$$\widehat{g}_{\phi}^{\text{sampled}} = \frac{1}{M} \sum_{j=1}^{M} \widehat{\mu}_{u_j} \nabla_{\phi} \log p_{u_j}, \qquad u_j \overset{\text{iid}}{\sim} p. \tag{29}$$

It reuses the covered route means and requires no additional workflow execution. Conditionally on those means its expectation equals the exact Router sum.

E VALIDATION EXECUTION ACCOUNTING

Table 4: Completed Stage-B validation on 128 reused development questions. Selected steps are IID/CAGE; differences are CAGE minus IID in percentage points. Selected means use the checkpoint-selection data, not held-out data.

		Validation-selected			Fixed step ten		
Seed	Steps	IID	CAGE	Δ (pp)	IID	CAGE	Δ (pp)
88501	6/9	.45898	.46387	+0.49	.44727	.45313	+0.59
88502	2/2	.46484	.46973	+0.49	.45215	.45898	+0.68
88503	9/8	.46387	.47168	+0.78	.44727	.46680	+1.95
Mean	–	.46257	.46842	+0.59	.44889	.45964	+1.07

Table 5: Natural-policy validation on 128 questions, eight workflows per question, and seed 87501. Δ is CAGE minus the row method. Paired 95% question-bootstrap intervals use 10,000 replicates and are conditional on this seed; every interval includes zero.

Method	Reward ↑	Δ	95% interval
IID + global RLOO	0.44824	+0.00781	[−0.01172, +0.02832]
IID + SAN/global blend	0.44531	+0.01074	[−0.01172, +0.03418]
CAGE with sampled Router	0.45312	+0.00293	[−0.01465, +0.02051]
Full CAGE	**0.45605**	–	–

Every arm executes 1,024 workflows. Different learned Router probabilities produce different node counts despite common route uniforms. Unscorable submissions receive zero reward and remain in the denominator.

Method	Specialist node calls	Scorable submissions
IID + global RLOO	2,451	1,024/1,024
IID + SAN/global blend	2,401	1,023/1,024
CAGE with sampled Router	2,469	1,021/1,024
Full CAGE	2,387	1,023/1,024

Table 6: Execution accounting for the frozen four-arm validation.

F FIXED-POLICY MECHANISM ACCOUNTING

The coverage diagnostic uses the same initial policy and 32 training questions for both methods, with 256 complete workflows per arm. CAGE uses 524 Specialist calls and 2,385,538 generated action tokens (including EOS); IID uses 600 calls and 2,910,482 tokens. Length truncation affects 19 and 34 nodes, respectively. These are rollout-only counts, excluding scoring, backward passes, the fixed answer reader, recovery checks, and failed work. They do not establish total-compute efficiency. Known question and reference defects were left unchanged, so this diagnostic supports a statement about route counts, not about answer quality.

For a second diagnostic, fix CAGE's observed route means and Router probabilities and compare exact coefficients to those from eight independent categorical Router draws. With a matched, common route-mean baseline, the mean summed coefficient variance is 0.00121533 for sampling and zero for exact summation conditional on those inputs. With raw route means, the sampled value is 0.01887468; the two representations must not be mixed in a variance comparison. These are coefficient-space calculations, not measurements of parameter-gradient variance or uncertainty in the estimated route values.

The budget analysis in Table 1 holds all 32 Router distributions fixed and evaluates binomial empty/singleton probabilities. It introduces neither new model samples nor new trained checkpoints. In particular, the nonmonotonic singleton count is compatible with cells moving from empty to singleton as the budget increases.

F.1 Actual Specialist gradient under route reward shifts

This diagnostic reuses one short question from the saved initial-policy rollouts: eight CAGE workflows with route counts $(3, 2, 3)$ and eight independently generated IID workflows with counts $(4, 1, 3)$. The question was chosen post observation for short runtime, not by its shift sensitivity. Before diagnostic execution, we froze the saved inputs, estimators, offsets $(c_A, c_B, c_C) = (-1, 0, +1)$, and numerical gates. This is one question with 16 workflows, not 16 independent questions.

We reconstruct the seeded Qwen3.5-4B Specialist with an FP32 reference backend and recorded pre-action sampling supports. For each workflow, we differentiate the sum of its node log probabilities over all 32,464,896 trainable LoRA coordinates. These full FP32 vectors are used to form a double-precision Gram matrix G of trajectory gradients. With original and shifted estimator coefficients a, a', squared norms are computed as $\|\widehat{g}\|^2 = a^\top G a$ and $\|\widehat{g}' - \widehat{g}\|^2 = (a' - a)^\top G (a' - a)$. Thus no projected gradient or approximate population reference is used. The CAGE coefficients agree exactly before and after the offset, making its gradient difference exactly zero; a second backward pass is unnecessary.

Table 7: Full Specialist LoRA-gradient route-shift diagnostic. Each relative change uses that arm's own original gradient norm.

Estimator	$\|\widehat{g}\|^2$	$\|\widehat{g}' - \widehat{g}\|^2$	Relative change
CAGE local RLOO	492097.55	0	0
IID global RLOO	21804.74	778740.19	5.976

The maximum absolute difference between saved and rescored workflow-mean action log probabilities was 4.47×10^{-7}. This is an aggregate rescoring check, not a maximum per-token error bound. The diagnostic used 33,689 CAGE and 44,511 IID saved action tokens, with no new generation, reward requests, or optimizer steps. It tests within-batch sensitivity to fixed offsets, not between-estimator variance, MSE against a population gradient, or learning efficacy. The independently generated arms need not have identical original gradient norms.

G REPORTING CHECKLIST

THEORY–EVIDENCE MAP

Table 8: Theory–experiment map. Each component addresses a different finite-budget failure mode; the listed diagnostic is a falsifiable mechanism check, not a guaranteed improvement in terminal reward.

Design choice	Finite-budget problem	Formal consequence	Primary diagnostic
Live-policy allocation with floor	Random route counts; an executable route can be absent	At least two executions per route and exactly the fixed total	Per-route counts and IID zero-coverage frequency
Exact finite-Router sum	Categorical Router choice adds sampling noise	Removes additional categorical noise for a same-covered-batch comparator	Conditional coefficient noise; full parameter-gradient dispersion unmeasured
Weighted route-local RLOO	Route-only reward offsets enter a global finite-sample advantage	Sample-wise route-shift invariance; live-policy weighting preserves the target mixture	Full Specialist LoRA-gradient response to fixed route reward offsets

The accompanying evidence summary records aggregate provenance, decoding and budget settings, checkpoint selection, and implementation limits. Internal execution journals and dataset question text are not included in the anonymous submission package. No short-horizon tuning quantity is relabeled as held-out evidence.