BITROUTER: LEARNING INTERPRETABLE MODEL ROUTING POLICIES FOR AGENTIC WORKFLOWS FROM DEPLOYMENT FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic workflows involve state-dependent model choices whose combined effects are often evaluated through delayed, session-level outcomes. Building on bandit-based routing and cost-subsidy formulations, we introduce BitRouter, a framework for learning interpretable workflow routing policies under cost and quality requirements. Each complete policy is a bandit arm retained throughout a session, while explicit rules adapt model choices to workflow state. We reuse session-level evidence across policies with shared rules, accounting for uncertainty in the rules' combined effects relative to quality-feasibility margins when deciding whether additional policy-specific evaluation is needed. Across repeated runs on 80 common-valid Terminal-Bench 2.1 tasks, an initial routing configuration achieved 81.25% task success, versus 81.56% for a fixed strong-model baseline and 76.50% for same-pool random routing. Mean nominal API cost per trial, counting accepted valid-path requests, was 40.93% lower than that of the strong-model baseline.

1 INTRODUCTION

Model requirements change during an agentic workflow: planning and failure diagnosis may benefit from stronger models, while routine operations may use cheaper ones. As models complete longer software tasks (Kwa et al., 2025), model choices increasingly depend on execution history. A cheaper call can trigger failed tests and rework, making complete-session quality and cost the relevant outcomes.

Learning from these outcomes requires attributing interacting routing decisions to terminal feedback while accounting for execution and verification expense. We seek an acceptable replacement within a registered policy library: lower expected session cost with quality within a specified tolerance of a reference and, optionally, above an absolute floor. This connects the problem to multiple-answer and cost-aware identification (Jourdan & Degenne, 2022; Kanarios et al., 2024). Our central question is: **how can session outcomes and shared policy structure guide both evaluation and reference-relative replacement?**

We introduce BitRouter. **Our core contribution is an executable learning procedure that unifies complete-policy outcome evaluation, structural evidence sharing, and reference-relative replacement.** It distinguishes the policy executed in a session, the observation needed to resolve a candidate–reference comparison, and the policy eligible for deployment. Two designs connect these decisions:

1. **Complete workflow policies as executable learning arms.** A session-bound configuration adapts model choices to workflow state. Versioned rules, safeguards, fallbacks, and execution records attribute terminal outcomes to the complete policy.
2. **Decision-directed policy replacement with shared evidence.** Quality and cost models share edit and interaction parameters while retaining policy-specific uncertainty. Candidate–reference comparisons guide evaluation of the candidate, reference, or related policies relative to execution and verification expense. Direct evidence, joint quality–cost support, and projected payback govern adoption. Conditional analysis relates these choices to distinguishability, quality-margin coverage, and information geometry.

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.

Real session records demonstrate allocation, execution, feedback, updates, and adoption assessment (Section 4.4). Separately, frozen-configuration evaluation on 80 common-valid Terminal-Bench 2.1 tasks yields 81.25% success for BitRouter G1, versus 81.56% for a fixed strong model and 76.50% for same-pool random routing, with 40.93% lower mean accepted-path nominal API cost than the strong model. Section 5 specifies replication, filtering, and uncertainty.

2 Related Work

2.1 Model Routing for LLMs and Agentic Workflows

Routing methods differ in decision timing, available signals, and selection mechanism (Moslem & Kelleher, 2026). RouteLLM (Ong et al., 2025) learns request routing from preferences; FrugalGPT (Chen et al., 2024) uses response scores for cascading; PILOT (Panda et al., 2025) combines offline preferences with budget-constrained online bandits. Switchcraft (Agarwal et al., 2026) predicts tool-call correctness from conversation context and tool definitions, then selects among predicted-correct models by profiled cost.

Workflow outcomes and evolving state also inform routing. Router-R1 (Zhang et al., 2025) learns multi-round selection and aggregation from final-outcome and cost rewards. EvoRoute (Zhang et al., 2026) combines retrieved experience and Thompson sampling for stepwise selection using performance, cost, and delay. ProgRouter (Li et al., 2026) uses workflow state and predicted progress under time and cost budgets. BitRouter evaluates registered state-to-action mappings, including their priorities and recovery behavior, from complete-session outcomes.

TRACE-Router (Raj et al., 2026) closely aligns task assignment with terminal supervision: a backend is bound at admission and updated from an accuracy–latency reward. BitRouter binds a complete state-dependent policy, allowing models to change within a session while preserving one configuration-level learning arm. Across sessions, the learner assesses replacing a named reference under separate quality and cost conditions.

2.2 Policy Evaluation and Evidence Sharing

Good-arm identification (Katz-Samuels & Jamieson, 2020) permits answers above a threshold; multiple-answer identification (Jourdan & Degenne, 2022) examines how acceptable answers affect sampling effort; cost-aware identification (Kanarios et al., 2024) separates observation expense from selected-arm value. BitRouter seeks an acceptable replacement within a finite budget, whereas constrained best-arm identification (Lardy et al., 2025) requires identifying the best feasible answer.

Pairwise Elimination (Juneja et al., 2025) compares candidates against a reference of unknown mean quality in ascending order of known arm costs. BitRouter estimates session execution cost alongside quality because model choices affect trajectory length, retries, and recovery. It distinguishes deployment cost c_π from evaluation expense κ_π, which also includes paid verification: the former determines replacement savings, the latter the price of evidence.

Structural methods differ in how they connect observations. LLMSelector (Chen et al., 2025) searches static compound-system assignments through module-performance estimates, with analysis under monotonicity assumptions. CABS (Sridhar et al., 2026) supplies surrogate rewards for related model arms; its decoupled variant mixes reward-only and surrogate-aware experts. Multiple-policy evaluation (Dann et al., 2023) reuses trajectories through policy overlap. BitRouter shares parameters through registered edits and interactions while retaining complete-policy observations, policy-specific uncertainty, and separate direct-evidence counts. Misspecified identification (Réda et al., 2021) motivates making the transfer assumptions explicit.

Transductive experimental design (Fiez et al., 2019) targets informative candidate comparisons; Predictive Entropy Search with Constraints (Hernandez-Lobato et al., 2015) values information about a constrained optimum with separately evaluated objectives and constraints. Bandits with Costly Reward Observations (Tucker et al., 2023) models payment for reward feedback. BitRouter connects these concerns to policy replacement: comparison covariance, direct-evidence needs, and measurement expense guide acquisition; joint quality–cost support and projected payback govern adoption.

3 POLICY LEARNING WITH SHARED EVIDENCE

3.1 POLICIES, OUTCOMES, AND REPLACEMENT OBJECTIVE

Consider agentic workflows drawn from a fixed task distribution $\mathcal{D}$ and a finite policy library $\Pi = \{\pi_0, \dots, \pi_{K-1}\}$ with reference π_0. Session i is assigned policy index A_i. At request t,

$$s_{i,t} = \phi(h_{i,t}), \qquad a_{i,t} = \pi_{A_i}(s_{i,t}), \tag{1}$$

where $h_{i,t}$ is observable execution history and $a_{i,t}$ specifies a model and reasoning configuration. A complete policy fixes routing conditions, priorities, compatibility constraints, safeguards, and fallbacks throughout the session. It may use a cheaper model for editing and escalate after a failed check; repair effort and terminal outcomes belong to the same arm. The harness, extractor, model versions, and library remain fixed within the learning environment.

Let $Y \in [0, 1]$ denote terminal quality under a prespecified protocol and C^{exec} session cost under a declared accounting convention. Define $q_\pi = \mathbb{E}_{\mathcal{D},\pi}[Y]$ and $c_\pi = \mathbb{E}_{\mathcal{D},\pi}[C^{\text{exec}}]$, including task and execution randomness. Cost reflects prices, trajectory length, retries, caching, and recovery; rubric scores and task-verifier outcomes are distinct quality measurements.

For quality tolerance $\epsilon \geq 0$ and optional floor $q_{\min}$, set $q_{\text{req}} = \max\{q_{\pi_0} - \epsilon, q_{\min}\}$. The acceptable replacements are

$$\mathcal{G} = \{\pi \in \Pi : q_\pi \geq q_{\text{req}}, \quad c_\pi < c_{\pi_0}\}. \tag{2}$$

Reference quality is estimated jointly; the optional floor can exclude the reference. Define decision margins

$$D_\pi^{\text{rel}} = q_\pi - q_{\pi_0} + \epsilon, \qquad D_\pi^{\text{abs}} = q_\pi - q_{\min}, \qquad S_\pi = c_{\pi_0} - c_\pi. \tag{3}$$

The learner seeks a member of $\mathcal{G}$ within budget, returning an adopted policy or an unresolved decision $\perp$ that retains the reference. The quality–cost objective is related to cost-subsidy bandits (Sinha et al., 2021).

For N allocated sessions, direct learning expenditure is

$$L_N = \sum_{i=1}^{N} \big(C_i^{\text{exec}} + C_i^{\text{verify}}\big). \tag{4}$$

Verification includes paid attempts without usable labels; generation expense is separate. Expected evaluation expense κ_π differs from c_π. Decisions use arrived evidence, tracking unsettled commitments and unknown charges separately (Appendix B).

3.2 SHARED ESTIMATION AND DIRECT EVIDENCE

Let z_π indicate registered configuration edits, w_π encode declared interactions, and $x_\pi = (1, z_\pi^\top, w_\pi^\top)^\top$. These features are fixed before allocation. Separate working models estimate quality and arithmetic mean execution cost:

$$g(q_\pi) \approx x_\pi^\top \theta_Q + b_{Q,\pi}, \qquad c_\pi \approx x_\pi^\top \theta_C + b_{C,\pi}. \tag{5}$$

Binary quality uses logistic regression with a Laplace approximation; rubric scores use Gaussian regression. Priors and regularization govern transfer, with predictions constrained to valid quality and nonnegative costs.

For edits A, B, and their interaction, $x_0 = (1, 0, 0, 0)$, $x_A = (1, 1, 0, 0)$, $x_B = (1, 0, 1, 0)$, and $x_{AB} = (1, 1, 1, 1)$. A informs coefficients used for A+B; interaction and residual terms capture nonadditivity, including changes in how often another edit's routing conditions are reached.

Two environments can agree on observations from the reference, A, and B while making A+B acceptable in only one; Appendix C.1 formalizes this ambiguity. With a fixed independent prior on an unobserved policy's residual, other-policy data leave its residual uncertainty unchanged (Appendix C.3). This motivates direct candidate evidence alongside structural transfer, whose assumptions are also central in misspecified identification (Réda et al., 2021).

Each available outcome in the latest terminal feedback enters once under its generating policy. The likelihood normalizes within dependence clusters linking repeated tasks and common-ancestor branches. Joint parameter draws preserve cross-policy covariance; common random cluster weights couple quality and cost calculations. Appendix B.1 specifies the resulting working distribution $\widetilde{P}_n$.

3.3 Decision-Directed Evidence Acquisition

After initial reference evaluation, acquisition selects a candidate comparison, then scores policies whose execution could resolve it. For relative quality, $\mathrm{Cov}_n(D_\pi^{\mathrm{rel}}, q_j) = \mathrm{Cov}_n(q_\pi, q_j) - \mathrm{Cov}_n(q_{\pi_0}, q_j)$. Value depends on the candidate–reference difference: observations of the reference or a related policy may therefore be more informative than observing the target candidate.

The information calculation follows the geometry of these margins. Conditional on current data, suppose θ is Gaussian with covariance Σ_n, the comparison is $D = d^\top \theta$, and a prospective observation is $Z_j = f_j + \xi_j$, with $f_j = u_j^\top \theta$ and independent Gaussian noise of variance $\sigma_j^2 > 0$. Here the parameter and feature vectors can include policy-specific terms. Gaussian conditioning gives

$$v_{j,n}(D) = \mathrm{Var}_n(D) - \mathrm{Var}_n(D \mid Z_j) = \frac{\mathrm{Cov}_n(D, f_j)^2}{\mathrm{Var}_n(f_j) + \sigma_j^2}. \tag{6}$$

Zero comparison covariance gives zero one-step variance reduction. This geometry connects acquisition to transductive experimental design (Fiez et al., 2019). The procedure uses Equation (6) locally for transformed quality margins; Appendix C.3 gives the exact Gaussian and fixed-design derivations.

Candidate priority combines feasible savings over a registered horizon with unresolved quality, cost, and direct-evidence requirements. For the target, variance reductions capped by these needs, direct-evidence deficits, and complete-quality value define $V_{j,n}^Q$ and $V_{j,n}^C$. With verification probability $p_{j,n}^{\mathrm{verify}}$, predicted execution cost $\widehat{c}_{j,n}$, and verification reservation r^{verify}, the score is

$$\alpha_{j,n} = \frac{p_{j,n}^{\mathrm{verify}} V_{j,n}^Q + V_{j,n}^C}{\max\{\widehat{c}_{j,n}, c_{\mathrm{num}}\} + p_{j,n}^{\mathrm{verify}} r^{\mathrm{verify}}}, \tag{7}$$

where $c_{\mathrm{num}} > 0$ is a numerical floor and monetary units are consistent. The denominator prices execution as a measurement, including planned verification. Joint draws estimate scores; a fixed coverage fraction is uniform across eligible policies and remaining mass across highest-score ties. With no positive score, the controller retains the reference.

Verification selection uses direct-evidence needs, label value relative to reserved expense, and randomized audits; an all-session mode verifies every session. The decision and its probability are recorded before the rubric outcome and persist across retries. Terminal feedback updates each model with its available measurements, including settled cost when quality is missing (Appendix B.3).

3.4 Adoption and Evidence Revision

Adoption requires direct candidate and reference evidence, joint quality–cost support, and projected recovery of excess learning expenditure. Direct task-cluster requirements apply to quality and cost separately. Given registered sensitivity margins, joint support for candidate π is

$$\widetilde{p}_{\pi,n} = \widetilde{P}_n\big(D_\pi^{\mathrm{rel}} \geq m_{\pi,n}^{\mathrm{rel}},\ D_\pi^{\mathrm{abs}} \geq m_{\pi,n}^{\mathrm{abs}},\ S_\pi > m_{\pi,n}^{C}\big)\,, \tag{8}$$

Algorithm 1 DECIDE: shared-evidence policy learning

Input: State $\mathcal{S}_n$, registered settings $\mathcal{R}$
Output: Retained policy d, exploration distribution p or $\varnothing$

$\widetilde{P}_n \leftarrow \text{FITANDDRAW}(\mathcal{F}_n, \mathcal{R})$ ▷ Section 3.2
$d \leftarrow \text{MAINTAIN}(\mathcal{S}_n, \widetilde{P}_n)$
$\mathcal{E}_n \leftarrow \{\pi \in \Pi \setminus \{\pi_0\} : \text{Adoptable}(\pi, \mathcal{S}_n, \widetilde{P}_n)\}$
if $\mathcal{E}_n \neq \varnothing$ **then**
    return $\left(\arg\min_{\pi \in \mathcal{E}_n} \widehat{c}_{\pi,n}, \varnothing\right)$
if $\neg\, \text{CanExplore}(\mathcal{S}_n, \mathcal{R})$ **then**
    return $(d, \varnothing)$ ▷ Budget and admission checks
if $\text{NEEDSINITIALIZATION}(\mathcal{S}_n)$ **then**
    return (d, δ_{π_0}) ▷ Reference-only allocation
$(\pi^\star, \mathcal{J}_n, \alpha_n) \leftarrow \text{TARGETANDSCORES}(\widetilde{P}_n, \mathcal{S}_n, \mathcal{R})$
if $\pi^\star = \perp$ **or** $\mathcal{J}_n = \varnothing$ **or** $\max_{j \in \mathcal{J}_n} \alpha_{j,n} \leq 0$ **then**
    return $(d, \varnothing)$
$\mathcal{M}_n \leftarrow \arg\max_{j \in \mathcal{J}_n} \alpha_{j,n}$
$p \leftarrow (1-\gamma)\,\text{Unif}(\mathcal{M}_n) + \gamma\,\text{Unif}(\mathcal{J}_n)$
return (d, p)

with the absolute comparison omitted when no floor applies. The quality margins include declared bias sensitivities and missingness allowances; $m^C_{\pi,n}$ accounts for bounded missing execution cost (Appendix B.4). A KL-inverted numerical lower bound on favorable-draw frequency must reach the working-model support threshold η. Appendix B.4 specifies the calculation and its fixed-state independent-draw interpretation. Separately, Appendix C.2 derives sequential adoption-error control conditional on simultaneous margin coverage (Howard et al., 2021).

Release checks unresolved outcomes, serious quality concerns, and cost completeness; unknown charges without upper bounds can block adoption. Horizon savings must exceed positive excess settled learning expenditure over an estimated all-reference counterfactual for the same session count. This payback check supplements Equation (2) under consistent accounting.

The lowest predicted-cost eligible candidate is selected; otherwise, permitted acquisition continues or replacement remains unresolved. Revised evidence can trigger withdrawal for new sessions without altering existing assignments, expense, or history.

In Algorithms 1–2, state $\mathcal{S}_n$ contains terminal feedback $\mathcal{F}_n$, the expenditure ledger, deployment status, and pending records. Settings $\mathcal{R}$ include the registered coverage fraction γ. MAINTAIN and ADOPTABLE apply the full checks in Appendix B.4.

Here δ_{π_0} is a point mass at the reference. TARGETANDSCORES selects the target comparison and eligible observation policies $\mathcal{J}_n$, then computes Equation (7) using Appendix B.2. Algorithm 2 handles versioned execution, verification, and outstanding evidence at budget exhaustion.

4 SYSTEM DESIGN AND IMPLEMENTATION

BitRouter separates request execution in the Router from evidence, allocation, and adoption management in Learn. Versioned records connect both components through the session lifecycle.

4.1 ARCHITECTURE AND SESSION BINDING

The registry stores complete configurations and verifies that combinations implement exactly their constituent edits. Learn records each new session's allocation distribution, sampled policy, and evidence version. The Router binds that policy version for the session's lifetime; later updates affect subsequent assignments.

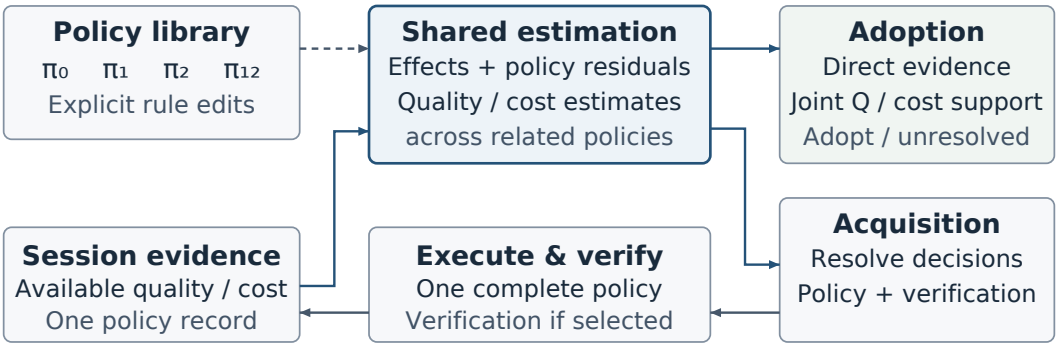


Figure 1: Configuration structure couples policy estimates; session outcomes inform evidence acquisition and adoption. Direct evidence contributes to release eligibility. Dashed arrows denote structural input; solid arrows denote evidence and decisions.

4.2 Workflow-State Policy Execution

Harness adapters project conversation and tool history into routing keys for workflow activity, context pressure, recovery, and risk. Priorities select a rule and tier, which supplies the model and reasoning effort. For example, implementation can use balanced and orchestration strong; verification can move from economy to balanced under context pressure or strong under guards. Task-specific rules can override general mappings.

Progress and tool-safety guards permit escalation after unproductive activity, recovery signals, or incomplete history. Compatibility checks and fallbacks determine the backend. Records link keys, rules, policy versions, backends, and fallbacks without an additional large-model call for rule evaluation.

4.3 Feedback and Learning Control

Session identifiers link assignments, executions, checkpoints, assessments, and settlements. Feedback becomes terminal after the session ends, execution settles, and selected verification completes or fails terminally. Learn uses the latest revision, retaining available cost when quality is missing and preserving history and incurred expense.

Recorded verification selections, reservations, attempt limits, and admission bounds control pending work. Learning states record estimates, allocation probabilities, direct counts, support bounds, and adoption checks. Budget exhaustion stops exploration while outstanding evidence settles. Frozen evaluation uses a separate source with learning disabled. Appendices A–B specify the event flow and accounting.

4.4 Learning in Operation

A real agent run demonstrates allocation, execution, feedback, estimation, and adoption assessment. The library fixes three model tiers, 18 workflow-state rules, and safeguards. Edit A changes strong-tier reasoning effort from high to medium; B changes balanced-tier effort from high to low. Their combination yields the four complete policies in Table 1.

Twelve tasks run twice in fixed order: 24 sequential sessions, initialized with two reference sessions, then decision-directed allocation with 10% uniform exploration. Every session is selected for rubric verification with at most three attempts per checkpoint. Frozen settings are quality floor 0.80, tolerance 0.05, support threshold 0.95, $48 verification budget, and 24-session payback horizon. Costs use accepted-path nominal accounting. Policy-structure features and repeated-task cluster weights define estimates over the observed panel; the task-by-policy record gives realized coverage.

Terra scores task delivery, constraint compliance, and executable verification with weights 4, 3, and 3 under a frozen prompt. Complete scores require all applicable criteria and gap-free capture; the

Table 1: Complete policies and their shared configuration structure. The strong, balanced, and economy tiers use Astra (`gpt-6-astra`), Terra (`gpt-5.6-terra`), and Luna (`gpt-5.6-luna`), respectively. Model identities, routing rules, and safeguards are common to all four policies.

Policy	Change from reference	Strong: Astra	Balanced: Terra	Economy: Luna	Shared structure
Reference	None	high	high	high	Common reference
A	Strong-tier reasoning effort	medium	high	high	Shares edit A with A+B
B	Balanced-tier reasoning effort	high	low	high	Shares edit B with A+B
A+B	Both reasoning-effort edits	medium	low	high	Contains edits A and B

Table 2: Allocation and cumulative evidence after the last terminal-feedback revision in each interval. Counts require complete rubric scores; execution expense excludes grading. The highest-probability policy need not be the sampled policy.

Learning sessions	Highest-probability policy	Cumulative complete quality scores	Cumulative nominal execution expense (USD)
1–2	Reference	2	12.621
3–4	A+B	4	48.191
5–6	A	5	106.137
7–8	A+B	7	122.501
9–18	B	15	296.877
19–20	Reference	17	325.586
21–24	A	21	366.531

0.80 floor applies to this rubric. Appendix B.5 and the scoring protocol give aggregation, settings, and separate noise calibration.

Table 2 groups successive allocations by highest-probability policy; terminal feedback precedes each subsequent allocation.

Actual allocations comprise five reference sessions, five A sessions, ten B sessions, and four A+B sessions. Session 5 distinguishes directed allocation from randomized coverage: A has probability 0.925, while the reference is sampled through its probability of 0.025.

All 24 trials terminate with 24 terminal-feedback records, 1,047 settled executions, 21 complete quality scores, three missing scores, and 24 complete costs. The 24 grading jobs make 27 requests (three additional attempts), costing $17.895 (4.88% of execution expense). Execution costs $366.531, totaling $384.426. Sessions use 10–156 calls and $0.784–$49.062, a 62.6-fold cost range.

At the 24-session limit, all policies meet the minimum of two direct quality and cost clusters. Predicted mean session costs are $12.56 for the reference and $15.34, $15.79, and $14.76 for A, B, and A+B. Every candidate falls below the joint-support threshold after the quality and missingness allowances and fails the payback check, so the learner retains the reference. The decision breakdown records support bounds, margins, and expected savings. It is frozen for eight disjoint evaluation tasks under a separate source, leaving learning data unchanged; session records and settings and decisions document the sequence.

5 Experiments

We evaluate the execution performance of frozen routing configurations on Terminal-Bench 2.1, comparing task success and nominal API cost with fixed-model and same-pool random-routing baselines.

Table 3: Frozen-policy results on 80 common-valid tasks; cost covers accepted valid-path requests. † indicates partially estimated usage. Extra decimals support archive reconciliation.

Configuration	Reps./ task	Successes/trials	Success (%)	Cost (US dollars/trial)	Cost reduction (%)
Strong model	4	261/320	81.56	0.830123	—
BitRouter G1	5	325/400	81.25	0.490317	40.93
Same-pool random	5	306/400	76.50	0.552670	33.42
BitRouter G3†	5	302/400	75.50	0.497511	40.07

Table 4: Exploratory 95% bootstrap intervals. Differences compare G1 with the named control; success differences are in percentage points (pp). Each bootstrap draw recomputes the ratio of group mean costs.

Control	Success difference (pp)	95% interval (pp)	Cost reduction (%)	95% interval (%)
Strong model	−0.31	[−4.81, 4.13]	40.93	[32.92, 48.49]
Same-pool random	+4.75	[1.00, 8.75]	11.28	[−0.50, 22.27]

5.1 Evaluation Setup and Cost Accounting

The study comprises 89 tasks and 1,691 trials using Harbor 0.18.0, Terminus 2, and at most 24 interaction rounds. The baseline has four replicates per task; routing groups have five. Concurrency is four for baseline and BitRouter and six for random routing (Appendix D).

The strong baseline uses `gpt-5.6-sol`. G1 adds `kimi-k3` and `deepseek-v4-flash-0731`; G3 replaces Kimi with `deepseek-v4-pro-0813`. Same-pool random routing independently samples a model per request with GPT/Kimi/Flash weights 1510/3027/200. Configurations remain fixed; Appendix D.1 gives reasoning settings.

The 80 common-valid tasks have no top-level Harbor exception in any required replicate across all four groups: 320 baseline trials and 400 per other group. Task verifiers determine success; unsuccessful trials without such exceptions remain included.

Mean nominal API cost per trial uses archived usage and prices for accepted valid-path requests from successful and unsuccessful tasks. G3's partially estimated DeepSeek Pro usage is marked in Table 3 (Appendix D.2).

5.2 Frozen-Policy Performance

G1 achieves 81.25% success, 0.31 percentage points below the strong-model baseline, at 40.93% lower mean nominal cost (Table 3). Relative to same-pool random routing, success is 4.75 points higher and cost 11.28% lower.

5.3 Uncertainty and Sensitivity Analyses

Table 4 reports exploratory paired task-cluster bootstrap intervals: group replicates are averaged within tasks, then the 80 tasks are jointly resampled 20,000 times using seed 20260923. Appendix D.4 supplies the procedure, hashes, and analysis artifacts.

The baseline-relative cost-reduction interval is positive, while its success-difference interval spans zero. Relative to random routing, the success-difference interval is positive and the cost-reduction interval includes zero.

Including all 89 tasks and trials with top-level exceptions preserves the success-rate ordering under the original denominators (Table 7).

6 Discussion and Conclusion

Statistical scope. Estimates depend on registered features, priors, and task population. Intercept, two edits, and their interaction span four policy means; residual terms require regularization, and fitted components are not identifiable causal effects. Cluster weighting addresses repeated evidence, not task composition. Fixed ordering and adaptive allocation yield uneven coverage without task-difficulty adjustment, so estimates summarize the observed panel under the working model. Selective verification and informative missingness can bias quality estimates; bias sensitivities are not empirically established bounds. The Gaussian identity locally approximates acquisition outside its exact assumptions.

Numerical scope. KL inversion has a fixed-state independent-draw interpretation (Appendix B.4). Draw reuse across updates and acquisition supplies no sequence-wide error guarantee. Working-model support, numerical error, and the simultaneous outcome coverage assumed in Appendix C.2 are distinct.

Measurement scope. The checkpoint rubric assesses agent-visible evidence. The 21 complete scores in Section 4.4 average 0.95905; native verification passes 13/24 sessions, including 12/21 with complete scores. Six sessions score 1 but fail native verification: the evaluator cites successful in-session checks, whereas native records include boundary-test failures, a compilation conflict, and missing parsed results. The case-level diagnostic preserves these distinct measurements.

Empirical scope. The records do not establish independence of configuration development from evaluation tasks. Common-valid filtering, service periods, unequal replication, and concurrency limit generalization. Random routing does not match realized model shares or trajectories; G3 changes the pool, and single-model controls cover only the strong baseline. Comparisons do not isolate rules' causal effects. The post hoc task bootstrap conditions on observed replicate means, without multiplicity correction or a prespecified noninferiority test.

Accounting and deployment scope. Accepted-path nominal cost excludes potentially charged failed attempts and learning, verification, or infrastructure expenses outside that metric; savings need not match provider bills. Estimated G3 usage remains marked. Background inference consumes resources, and model or task shifts can weaken historical evidence. The archive fixes a source checkout, without a separately reconstructible deployment image per group; results are not an official leaderboard submission.

Implementation directions. Future versions could replace heuristic semantic classification with embeddings and compact neural sequence encoders that map conversation and tool histories to existing routing keys, with confidence-based fallback to explicit rules. Extending the policy compiler with constraint-guided edit generation and rule-overlap checks could expand candidate libraries while preserving safeguards. For larger libraries, cluster-level sufficient-statistic caches and reusable matrix factorizations could reduce repeated inference work, with invalidation on feedback revisions. Verification workers could add budget-aware batching and backpressure to improve throughput while preserving pre-outcome selection, immutable session assignments, and per-attempt accounting. These extensions would retain the auditable policy interface.

Conclusion. BitRouter unifies complete-policy outcome evaluation, structural evidence sharing, and reference-relative replacement. Session-bound policies adapt to workflow state; terminal feedback updates related policies. Candidate–reference comparisons guide evaluation, while direct evidence, joint quality–cost support, and payback govern adoption. Conditional analysis identifies the roles of candidate-specific uncertainty and comparison covariance.

Real records demonstrate allocation, execution, feedback, updates, and adoption assessment. Separately, frozen-configuration evaluation on 80 common-valid Terminal-Bench tasks yields 81.25% success for G1 with 40.93% lower mean accepted-path nominal API cost than the strong baseline.

AI USE STATEMENT

We used generative AI through OpenAI Codex to assist with developing and refining the conceptual framework, research hypotheses, mathematical claims, and proof arguments; designing and reviewing the experimental methodology and parameters; implementing and debugging the methods, tests, and experiment orchestration; processing and formatting recorded experimental data; and analyzing execution traces and interpreting results. We also used it for literature discovery and comparison, citation checks, manuscript drafting and restructuring, Chinese-to-English translation, language editing, and preparation of tables and supporting artifacts. Scientific Agent Skills (Kassis et al., 2026) provided procedural guidance for literature review and manuscript assessment. The language models used as experimental agents and quality evaluators, and the procedures governing their outputs, are documented in Sections 4–5 and Appendices B and D. All AI-assisted material has been reviewed by the author, including the source code, mathematical arguments, citations, experimental analyses, and manuscript text. The author takes full responsibility for the final content of this work, including its claims, results, and artifacts.

REPRODUCIBILITY STATEMENT

Section 3 and Appendices A–C specify the learning procedure, inference and acquisition rules, adoption checks, assumptions, and proofs. The anonymous supplementary material contains the frozen learning and benchmark source snapshots, dependency lockfiles, configurations, task references and ordering, seeds, and source-provenance manifests. For Section 4.4, it includes the calibration observations, session records, rubric and evaluator specification, native outcomes, execution and verification ledgers, and scripts that regenerate the reported learning statistics. For Section 5, Appendix D describes the benchmark versions, trial attribution, cost accounting, exclusions, and bootstrap procedure; the supplement provides the corresponding native records, analysis inputs, and executable analysis. Its verification script checks file hashes and reconciles the records and reported aggregates. The supplementary README gives commands for reproducing the numerical results from the included records using Python's standard library, together with the environment and service requirements for running new agent trials.

REFERENCES

Sharad Agarwal, Pooria Namyar, Alec Wolman, Rahul Ambavat, Ankur Gupta, and Qizheng Zhang. Switchcraft: AI Model Router for Agentic Tool Calling. *arXiv preprint arXiv:2605.07112*, 2026. URL https://arxiv.org/abs/2605.07112v1.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=cSimKw5p6R.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Matei Zaharia, James Zou, and Ion Stoica. Optimizing Model Selection for Compound AI Systems. *arXiv preprint arXiv:2502.14815*, 2025. URL https://arxiv.org/abs/2502.14815.

Chris Dann, Mohammad Ghavamzadeh, and Teodor V. Marinov. Multiple-policy High-confidence Policy Evaluation. In *International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 9470–9487. PMLR, 2023. URL https://proceedings.mlr.press/v206/dann23a.html.

Tanner Fiez, Lalit Jain, Kevin Jamieson, and Lillian Ratliff. Sequential Experimental Design for Transductive Linear Bandits. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.neurips.cc/paper_files/paper/2019/hash/8ba6c657b03fc7c8dd4dff8e45defcd2-Abstract.html.

Jose Miguel Hernandez-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive Entropy Search for Bayesian Optimization with Unknown Constraints. In *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1699–1707. PMLR, 2015. URL https://proceedings.mlr.press/v37/hernandez-lobatob15.html.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021. doi: 10.1214/20-AOS1991. URL https://arxiv.org/abs/1810.08240.

Marc Jourdan and Rémy Degenne. Choosing Answers in Epsilon-Best-Answer Identification for Linear Bandits. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10384–10430. PMLR, 2022. URL https://proceedings.mlr.press/v162/jourdan22a.html.

Ishank Juneja, Carlee Joe-Wong, and Osman Yagan. Pairwise Elimination with Instance-Dependent Guarantees for Bandits with Cost Subsidy. In *International Conference on Learning Representations*, pp. 62262–62319, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9be39b35906526b8d240056daac72c6f-Abstract-Conference.html.

Kellen Kanarios, Qining Zhang, and Lei Ying. Cost Aware Best Arm Identification. *Reinforcement Learning Journal*, 4:1533–1545, 2024. URL https://rlj.cs.umass.edu/2024/papers/Paper193.html.

Timothy Kassis, Vinayak Agarwal, Yuhuan He, Darshil Patel, and Aubrey M. Brueckner. Scientific Agent Skills: A Library of Procedural Knowledge for Research Agents. *arXiv preprint arXiv:2609.00065*, 2026. URL https://arxiv.org/abs/2609.00065.

Julian Katz-Samuels and Kevin Jamieson. The True Sample Complexity of Identifying Good Arms. In *International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1781–1791. PMLR, 2020. URL https://proceedings.mlr.press/v108/katz-samuels20a.html.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Chris Painter, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI Ability to Complete Long Software Tasks. *arXiv preprint arXiv:2503.14499*, 2025. URL https://arxiv.org/abs/2503.14499.

Tyron Lardy, Christina Katsimerou, and Wouter Koolen. Constrained Best Arm Identification. In *Advances in Neural Information Processing Systems*, volume 38, pp. 100601–100627, 2025. doi: 10.52202/085713-3366. URL https://papers.nips.cc/paper_files/paper/2025/hash/917373186cceb7efe90742ea8a51ca78-Abstract-Conference.html.

Songyuan Li, Ahmed M. Abdelmoniem, and Shiqiang Wang. ProgRouter: Online Progress-Guided Orchestration for Multi-Agent LLM Workflows under Quality-Cost Tradeoffs. *arXiv preprint arXiv:2608.25992*, 2026. URL https://arxiv.org/abs/2608.25992.

Yasmin Moslem and John D. Kelleher. Dynamic Model Routing and Cascading for Efficient LLM Inference: A Survey. *arXiv preprint arXiv:2603.04445*, 2026. URL https://arxiv.org/abs/2603.04445v3.

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*, pp. 34433–34448, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5503a7c69d48a2f86fc00b3dc09de686-Abstract-Conference.html.

Pranoy Panda, Raghav Magazine, Chaitanya Devaguptapu, Sho Takemori, and Vishal Sharma. Adaptive LLM Routing under Budget Constraints. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23934–23949, 2025. doi: 10.18653/v1/2025.findings-emnlp.1301. URL https://aclanthology.org/2025.findings-emnlp.1301/.

Ritik Raj, Souvik Kundu, Sarbartha Banerjee, Dheemanth Joshi, Ishita Vohra, and Tushar Krishna. TRACE-ROUTER: Task-Consistent and Adaptive Online Routing for Agentic AI. *arXiv preprint arXiv:2607.22465*, 2026. URL https://arxiv.org/abs/2607.22465.

Clémence Réda, Andrea Tirinzoni, and Rémy Degenne. Dealing With Misspecification In Fixed-Confidence Linear Top-m Identification. *arXiv preprint arXiv:2111.01479*, 2021. URL https://arxiv.org/abs/2111.01479.

Deeksha Sinha, Karthik Abinav Sankararaman, Abbas Kazerouni, and Vashist Avadhanula. Multi-Armed Bandits with Cost Subsidy. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3016–3024. PMLR, 2021. URL https://proceedings.mlr.press/v130/sinha21a.html.

Ajay Narayanan Sridhar, Ronak Singh, Mehrdad Mahdavi, and Vijaykrishnan Narayanan. Correlation-Aware Contextual Bandits with Surrogate Rewards for LLM Routing. *arXiv preprint arXiv:2607.09015*, 2026. URL https://arxiv.org/abs/2607.09015.

Aaron D. Tucker, Caleb Biddulph, Claire Wang, and Thorsten Joachims. Bandits with costly reward observations. In *Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 2147–2156. PMLR, 2023. URL https://proceedings.mlr.press/v216/tucker23a.html.

Guibin Zhang, Haiyang Yu, Kaiming Yang, Bingli Wu, Fei Huang, Yongbin Li, and Shuicheng Yan. EvoRoute: Experience-Driven Self-Routing LLM Agent Systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 38213–38225, 2026. doi: 10.18653/v1/2026.acl-long.1771. URL https://aclanthology.org/2026.acl-long.1771/.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-R1: Teaching LLMs Multi-Round Routing and Aggregation via Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 141233–141265, 2025. doi: 10.52202/085713-4719. URL https://papers.nips.cc/paper_files/paper/2025/hash/ceaa137fce916aba5c65fceb1309088b-Abstract-Conference.html.

A EVENT-DRIVEN LEARNING PROCEDURE

Algorithm 2 couples the decision step in Algorithm 1 to the Router and Learn components. INGEST preserves revisions, dependence clusters, charges, and unknown or pending states. $\mathcal{V}(e)$ contains completed sessions eligible for paid verification or a permitted retry. GETORFIXDECISION records selection and its probability before the rubric outcome and reuses that decision on retries (Appendix B.3).

Algorithm 2 Event-driven execution and feedback

Input: Policy library Π, reference π_0, registered settings and budgets $\mathcal{R}$
Output: Retained policy, evidence history, expenditure, and unresolved records

$\mathcal{S} \leftarrow$ INITIALIZE($\Pi, \pi_0, \mathcal{R}$)
while reporting cutoff has not been reached **do**
  $e \leftarrow$ NEXTEVENT; $\mathcal{S} \leftarrow$ INGEST($\mathcal{S}, e$)
  for $i \in \mathcal{V}(e)$ **do**
    $(v_i, p_i^{\text{verify}}) \leftarrow$ GETORFIXDECISION($i, \mathcal{S}, \mathcal{R}$)
    if $v_i = 1$ **and** RESERVEVERIFICATIONBUDGET($i, \mathcal{S}$) **then**
      SCHEDULEVERIFICATION(i)
  $\mathcal{S} \leftarrow$ FINALIZEFEEDBACK($\mathcal{S}$) ▷ Appendix B.3
  $(d, p) \leftarrow$ DECIDE($\mathcal{S}, \mathcal{R}$) ▷ Algorithm 1
  $\mathcal{S} \leftarrow$ PUBLISHFORNEWSESSIONS($\mathcal{S}, d$)
  if $p \neq \varnothing$ **then**
    $j \sim p$; $i \leftarrow$ NEWSESSION
    $a_i \leftarrow (i, j, p, \nu(\mathcal{S}), \text{version}(\pi_j))$
    $\mathcal{S} \leftarrow$ PERSISTASSIGNMENT($\mathcal{S}, a_i$); BINDANDDISPATCH(a_i)
return REPORT($\mathcal{S}$)

FINALIZEFEEDBACK retains available measurements when feedback becomes terminal; exhausted or budget-blocked verification has an explicit missing-quality state. $\nu(\mathcal{S})$ identifies the evidence version. Each assignment stores the complete allocation distribution and an immutable policy version before dispatch. Publishing or withdrawing a policy affects only new sessions. Budget exhaustion disables new exploration through CANEXPLORE; the event loop continues settling outcomes and reassessing deployment until the reporting cutoff.

Independent-policy estimation uses matched marginal priors and the same acquisition and adoption criteria, with shared coefficients disabled (Appendix B.1).

B INFERENCE, ACQUISITION, AND OPERATIONAL DETAILS

The configuration fixes numerical parameters before allocation. This appendix specifies inference, acquisition, and release; the supplementary record contains the settings and final decision for Section 4.4.

B.1 FEATURES, PRIORS, AND JOINT INFERENCE

The shared feature vector concatenates an intercept, binary indicators of registered edits, products for registered edit pairs, and a one-hot policy indicator. The latter supplies the policy-specific term in Equation (5). Interaction pairs are declared in the policy registry. Coefficients have independent Gaussian priors before conditioning. The quality intercept has mean μ_Q for rubric quality or $\text{logit}(\mu_Q)$ for binary quality; other quality coefficients have mean zero. Their standard deviations are s_0, s_z, s_w, s_b for the intercept, edits, interactions, and policy-specific terms. Execution cost is divided by a registered monetary scale s_C. Its intercept prior is $N(1, 1)$, and each remaining coefficient has prior $N(0, 0.5^2)$.

Independent-policy estimation uses one coefficient per policy. For a policy containing r_π edits and h_π registered interactions, its quality prior has the same mean and marginal variance $s_0^2 + r_\pi s_z^2 + h_\pi s_w^2 + s_b^2$ as the shared predictor. The scaled-cost prior has mean 1 and variance $1 + 0.25(r_\pi + h_\pi + 1)$. This preserves marginal prior scales while removing shared coefficients.

Dependence clusters are connected components of session, task, and fork-lineage identifiers. For each joint draw, sample one weight $W_g \sim \mathrm{Exp}(1)$ per cluster. For metric $m \in \{Q, C\}$, let $n_{g,m}$ count its available observations in cluster g, and let $\overline{W}_m$ be the mean weight over clusters containing that metric. Each such observation receives likelihood weight $W_g/(\overline{W}_m n_{g,m})$. Repetitions share each cluster's contribution, normalized separately for available quality and cost observations. This yields an equal-cluster working analysis without task-difficulty terms or task-composition adjustment. Direct-evidence counts separately count distinct clusters with a usable policy-specific observation for each metric.

Each fit uses the latest terminal feedback revision. Quality rows contain available, unretracted scores; cost rows contain exact settled execution costs. Rubric quality and scaled cost use weighted Gaussian regression with their registered observation standard deviations. Binary quality uses logistic iteratively reweighted least squares for at most 40 iterations, with Bernoulli variance floored at 10^{-6}, coefficient steps clipped to $[-2, 2]$, and stopping tolerance 10^{-8}. If the fitted precision is $LL^\top$, a coefficient draw is $\widehat{\theta} + L^{-\top} z$, where z is standard normal. Quality predictions use a sigmoid for binary outcomes or clipping to $[0, 1]$ for rubric outcomes; scaled-cost predictions are truncated at zero and multiplied by s_C.

A joint draw uses the same cluster weights for both metrics and separate normal draws for their coefficients. Evaluating all policies with each coefficient draw preserves cross-policy covariance. Random streams derive from the registered seed, feature order, and each cluster's earliest immutable assignment; common random numbers are reused across evidence updates. The resulting mixture of weighted Gaussian or logistic-Laplace fits defines $\widetilde{P}_n$.

B.2 Target Selection and Observation Values

All expectations, variances, and covariances below are computed from the joint draws. Let $b = 2(b_{\mathrm{feedback}} + b_{\mathrm{model}})$, let H be the deployment horizon, and write $\widetilde{c}_\pi = c_\pi / s_C$. For target π, define

$$M_\pi = \min\{D_\pi^{\mathrm{rel}}, D_\pi^{\mathrm{abs}}\} - b, \qquad T_\pi = \widetilde{c}_{\pi_0} - \widetilde{c}_\pi,$$
$$O_\pi = H s_C \, \mathbb{E}[\mathbf{1}\{M_\pi \geq 0\}(T_\pi)_+],$$

omitting the absolute margin when no floor is specified. O_π values potential feasible savings. Candidates with serious quality violations or zero opportunity are excluded from target selection.

For contrast $X \in \{M_\pi, T_\pi\}$, set $z^2 = 2\log(3/(1-\eta))$ and remaining variance need $d_X = (\mathrm{Var}(X) - (\mathbb{E}X)^2/z^2)_+$. A prospective observation of policy j has capped reduction

$$r_{X,j} = \min\left\{ d_X, \frac{\mathrm{Cov}(X, f_j)^2}{\max\{\mathrm{Var}(f_j) + \sigma_j^2, 10^{-12}\}} \right\},$$

where $f_j = q_j$ for quality and $f_j = \widetilde{c}_j$ for cost. Rubric and cost noise use the registered variances; binary quality uses $\max\{\widehat{q}_j(1-\widehat{q}_j), 0.01\}$. Policies with serious violations receive zero reduction. Let $n_{j,Q}, n_{j,C}$ be direct cluster counts and $n_{\min}$ their required minimum. Target priority is

$$\frac{O_\pi}{1 + \max\left\{ \frac{d_{M_\pi}}{\max_j r_{M_\pi,j}}, \; \frac{d_{T_\pi}}{\max_j r_{T_\pi,j}}, \; \sum_{j \in \{\pi, \pi_0\}} (n_{\min} - \min\{n_{j,Q}, n_{j,C}\})_+ \right\}},$$

with reduction denominators floored at 10^{-12}. The highest positive priority selects the target; ties follow registered policy order.

For the selected target, direct-evidence bonuses are $a_{j,m} = \mathbf{1}\{j \in \{\pi, \pi_0\}, \; n_{j,m} < n_{\min}\}/n_{\min}$. The quality value also includes the effect of a future complete observation on the missingness allowance. Let U_j be policy j's terminal cluster weight, W_j^{miss} its missing-quality weight, and ρ_π the candidate-plus-reference allowance defined in B.4. Define $\rho_{\max} = 1 - q_{\min} - b$ with an absolute floor, or $1 + \epsilon - b$ otherwise. If $j \in \{\pi, \pi_0\}$, $\rho_\pi > \rho_{\max}$, and $W_j^{\mathrm{miss}} > 0$, set

$$a_j^{\text{complete}} = \text{clip}_{[0,1]}\left(\frac{W_j^{\text{miss}}/U_j - W_j^{\text{miss}}/(U_j+1)}{\rho_\pi - \rho_{\max}}\right);$$

otherwise set it to zero. This acquisition term values one additional complete, unit-weight cluster; existing missing observations retain their recorded states. The values in Equation (7) are

$$V_j^Q = O_\pi\left(\frac{r_{M_\pi,j}}{\max\{\text{Var}(M_\pi), 10^{-12}\}} + a_{j,Q} + a_j^{\text{complete}}\right),$$
$$V_j^C = O_\pi\left(\frac{r_{T_\pi,j}}{\max\{\text{Var}(T_\pi), 10^{-12}\}} + a_{j,C}\right).$$

Equation (7) uses these monetary values, the predicted execution cost, the verification reservation, and a one-microdollar execution-cost floor. After reference initialization, the coverage fraction is uniform over eligible policies and the remaining mass is uniform over policies with the maximum score. If no positive score exists, the controller retains the reference. Each sampled assignment records the complete distribution, policy version, and evidence version.

B.3 Verification and Feedback

The planned verification probability is zero when no reservation capacity remains. Otherwise, it is one in all-session mode or while direct quality evidence is below $n_{\min}$. Once that requirement is met, a configured fixed probability takes precedence; absent that setting, verification probability is one when V_j^Q exceeds the reservation and equals the registered audit probability otherwise. The selection uses evidence available before the label, and the resulting decision persists across scheduling retries.

Budget checks track known verification charges, outstanding reservations, and registered liabilities for unknown receipts. Every paid attempt contributes expenditure, including attempts without a usable assessment. Exhausted attempts or a retry blocked by exhausted verification budget terminate the corresponding quality feedback with an explicit missing state. Intentionally unselected, failed, pending, and settled outcomes remain distinguishable. The quality likelihood uses available selected outcomes with the cluster weights in B.1; it applies no inverse-selection-probability reweighting. Settled cost is retained in the ledger as it arrives. It enters the cost fit once the session feedback is terminal; pending verification keeps that feedback pending. A terminal missing-quality record can therefore contribute an exact cost observation without a quality observation.

Revisions replace measurements while retaining their history and expense. Model features encode registered edits; execution traces separately retain realized states, rules, backends, fallbacks, and policy versions.

B.4 Adoption, Payback, and Maintenance

For missingness accounting, each terminal observation receives weight equal to the reciprocal of its terminal cluster size. Let $f_{j,Q}$ and $f_{j,C}$ be the resulting policy-specific fractions of missing quality and inexact execution cost; an empty policy has fraction one. Intentionally unselected labels are excluded from the missing-quality numerator. The quality allowance is $m_{\pi,n}^{\text{rel}} = m_{\pi,n}^{\text{abs}} = b + \rho_\pi$, where $\rho_\pi = \min\{f_{\pi,Q} + f_{\pi_0,Q}, 1\}$. With an enforced execution-cost cap $C_{\max}$, the cost allowance is $m_{\pi,n}^C = C_{\max}(f_{\pi,C} + f_{\pi_0,C})$. Without a cap it is zero, and unbounded missing execution cost separately blocks adoption.

Let $\widehat{p}$ be the fraction of M joint draws satisfying Equation (8), and set $\ell = \log(2K/0.01)$. For $\widehat{p} > 0$, the numerical lower bound L solves $M\,\text{kl}(\widehat{p}, L) = \ell$ on $[0, \widehat{p}]$ by 60 bisection steps, where kl is Bernoulli relative entropy; at zero, $L = 0$. The corresponding upper bound is $U = 1 - L(1 - \widehat{p})$. Adoption requires $L \geq \eta$, at least $n_{\min}$ direct quality and cost clusters for both candidate and reference, settled feedback, respected cost caps, and no serious candidate quality violation.

For a fixed evidence state and target distribution chosen independently of the numerical draws, independent identically distributed Bernoulli indicators across draws permit a Chernoff interpretation

of this inversion; a union bound over both tails and K policies gives error parameter 0.01. Independence across policies is not required for that union bound. The implementation reuses common random numbers across updates, and their estimates participate in acquisition. Its L and U are numerical decision criteria with this fixed-state interpretation; the calculation assigns no error budget across adaptive updates. The support threshold η is a working-model decision parameter. The simultaneous outcome-coverage premise in Appendix C.2 is a separate assumption.

Let $L^{\mathrm{upper}}_{\mathrm{settled}}$ be the sum of upper-bounded settled execution and verification expenses, and N_{settled} the number of settled assignments. The positive excess over the estimated all-reference execution counterfactual is

$$L_{\mathrm{excess}} = \max\{L^{\mathrm{upper}}_{\mathrm{settled}} - N_{\mathrm{settled}}\widehat{c}_{\pi_0}, 0\}.$$

Release additionally requires $H\mathbb{E}[S_\pi - m^C_{\pi,n}] > L_{\mathrm{excess}}$. An unavailable settled-cost upper bound blocks this comparison. The eligible candidate with lowest predicted execution cost is selected. Candidate-generation expense is recorded separately from this execution-plus-verification ledger.

Maintenance uses settled evidence and withdraws a retained candidate when an applicable release condition fails. For the model-support condition, withdrawal requires $U < \eta$; feedback still in flight alone does not trigger withdrawal. A withdrawal restores the reference for new sessions, while active sessions keep their assigned versions. Budget exhaustion stops exploration and permits remaining evidence to settle; the final record retains deployment status, decisions, and expenditure history.

B.5 Rubric Measurement and Noise Calibration

The frozen rubric specifies six dimensions: task delivery (weight 4), constraint compliance (3), executable verification (3), review resolution (2), PR delivery (1), and user acceptance (1). Delivery and constraints are mandatory. Optional dimensions are selected from the recorded obligations: missing an event does not cancel a required obligation, while user silence is not acceptance. Scores lie in $[0, 1]$, with anchors at 0, 0.5, and 1. Each scored criterion cites original checkpoint evidence. Positive verification requires a recorded tool observation, and a severe-violation flag requires its own cited evidence and is checked separately at adoption.

Let A include applicable criteria and criteria whose applicability is unknown, let $O \subseteq A$ have observed scores s_k, and set $W = \sum_{k\in A} w_k$ and $W_U = \sum_{k\in A\setminus O} w_k$. The stored quality interval is

$$Y_L = \frac{\sum_{k\in O} w_k s_k}{W}, \qquad Y_U = \frac{\sum_{k\in O} w_k s_k + W_U}{W}, \qquad \mathrm{coverage} = 1 - \frac{W_U}{W}.$$

The implementation stores scores in integer parts per million and rounds interval endpoints down at that precision. It emits a complete score only when $W_U = 0$, no mandatory criterion or applicability remains unknown, and the captured prefix has no reported gap. Excluded optional criteria contribute no weight. Unknown criteria retain their interval contribution and missing state.

The evaluator receives the frozen rubric and a cited projection of user messages, agent messages, tool calls and observations, file writes, and session boundaries. Setup/provider descriptors, routing and usage metadata, and private reasoning chunks are omitted by the projection. The evaluator has no tools; its prompt requires evidence-grounded applicability, scores, and explanations in a fixed JSON contract. The released measurement protocol includes the exact prompt, rubric library, projection and aggregation source, evaluator identifiers, and configuration hashes. Native task-verifier results are recorded separately from this checkpoint evidence.

Observation noise is frozen from a separate eight-task reference calibration with two executions per task. For metric m, let s_m be the sample standard deviation of available observations, and let P_m contain tasks with two available observations. The registered rule is

$$\sigma_m = \max\left\{ s_m, \sqrt{\frac{1}{2|P_m|}\sum_{g\in P_m}(y_{g,1} - y_{g,2})^2}, \sigma_{m,\mathrm{floor}} \right\}.$$

Quality uses 11 complete scores from seven tasks, including four complete pairs; its floor is 0.05 and the resulting standard deviation is 0.07254309. Cost uses 14 complete mapped observations from eight tasks, including six complete pairs, after division by $s_C = 10{,}000{,}000$ microUSD; its floor is 0.15 and the resulting standard deviation is 1.18148253 in scaled units. Missing or unmapped observations are excluded from the corresponding calculation. The two bias sensitivities, $b_{\text{feedback}} = b_{\text{model}} = 0.02$, are registered design tolerances, distinct from these estimated noise scales. Calibration inputs and the reproducible calculation accompany the measurement protocol.

C Analysis of Evidence Transfer and Decisions

This appendix collects the formal arguments used in Section 3: the information needed to distinguish a candidate, the coverage premise for sequential quality decisions, and the geometry of informative observations.

C.1 Limits of Structural Transfer

Two compatible configuration edits define four complete policies $\pi_0, \pi_A, \pi_B, \pi_{AB}$. Let $q_0 = q_A = q_B = 0.90$ and $\epsilon = 0.05$, with no absolute floor. Construct environments $\mathcal{E}_+$ and $\mathcal{E}_-$ with $q_{AB} = 0.95$ and 0.80, respectively. All information available before target-policy quality feedback—including execution signals, costs, and feedback timing—has identical conditional distributions. The combination is cheaper than the reference in both environments.

Proposition 1 (An unobserved combination need not be identifiable).

Let A be adoption of π_{AB} and T the event of receiving no quality feedback from it before adoption. Then

$$\Pr_{\mathcal{E}_+}(A \cap T) = \Pr_{\mathcal{E}_-}(A \cap T). \tag{9}$$

If the probability of any quality-infeasible adoption in $\mathcal{E}_-$ is at most δ, then $\Pr_{\mathcal{E}_+}(A \cap T) \leq \delta$.

Proof. Couple the learner's randomization. Until target quality feedback arrives, available histories and decisions have the same distribution. The event $A \cap T$ has equal probability and is an erroneous adoption in $\mathcal{E}_-$.

A learner that never obtains target quality feedback cannot simultaneously attain error at most δ in $\mathcal{E}_-$ and adoption probability at least $1 - \delta$ in $\mathcal{E}_+$ for $\delta < 1/2$. Informative side signals or justified structural restrictions can invalidate the construction. Thus shared estimation should retain uncertainty about unexplained policy differences and permit direct evaluation when structure does not resolve them. The scope of transfer depends on assumptions about misspecification (Réda et al., 2021).

C.2 Quality Margins and Simultaneous Coverage

Define $D_\pi^{\text{rel}} = q_\pi - q_{\pi_0} + \epsilon$. Suppose lower bounds jointly satisfy

$$\Pr\big(\forall n,\ \forall \pi \in \Pi :\ \underline{D}_{\pi,n} \leq D_\pi^{\text{rel}}\big) \geq 1 - \delta. \tag{10}$$

Adopting only when $\underline{D}_{\pi,n} \geq 0$ then bounds the probability of ever violating relative quality by δ: on the coverage event every adopted policy has $D_\pi^{\text{rel}} \geq 0$. An absolute floor requires an additional lower bound on $D_\pi^{\text{abs}} = q_\pi - q_{\min}$ within the same joint event. Cost-improvement control additionally requires a strictly positive lower bound on S_π.

Equation (10) specifies simultaneous coverage over policies and decision times (Howard et al., 2021). When justified, write $\underline{D}_{\pi,n} = \widehat{D}_{\pi,n} - r_{\pi,n} - B_\pi$, with estimation radius $r_{\pi,n}$ and a bound B_π on systematic comparison error. Candidate–reference covariance belongs in the uncertainty calculation. The decomposition separates estimation uncertainty from systematic error; candidate–reference margins determine how each term affects a decision.

C.3 Conditional and Fixed-Design Information

Consider the correctly specified model $y = X\theta+\xi$, with fixed, full-column-rank X and independent Gaussian noise of covariance $\sigma^2 I$. The full-rank condition applies to all coefficients in this fixed-design model. Set $V = X^\top X$ and $d = x_\pi - x_{\pi_0}$. Ordinary least squares has contrast variance

$$\mathrm{Var}(d^\top \widehat{\theta}) = \sigma^2 d^\top V^{-1} d. \tag{11}$$

Adding a policy vector x_j chosen independently of observed outcomes changes V to $V + x_j x_j^\top$. The Sherman–Morrison identity gives

$$(V + x_j x_j^\top)^{-1} = V^{-1} - \frac{V^{-1} x_j x_j^\top V^{-1}}{1 + x_j^\top V^{-1} x_j}, \tag{12}$$

so the reduction is

$$v_j(d) = \sigma^2 \frac{(d^\top V^{-1} x_j)^2}{1 + x_j^\top V^{-1} x_j}. \tag{13}$$

Setting $\Sigma_n = \sigma^2 V^{-1}$, $\sigma_j^2 = \sigma^2$, and $u_j = x_j$ in Equation (6) recovers this expression. For the conditional Gaussian model in Section 3.3, with augmented observation vector u_j, the joint covariance of (D, Z_j) is

$$\begin{pmatrix} d^\top \Sigma_n d & d^\top \Sigma_n u_j \\ u_j^\top \Sigma_n d & u_j^\top \Sigma_n u_j + \sigma_j^2 \end{pmatrix}. \tag{14}$$

Its conditional covariance is the Schur complement, proving Equation (6). Equation (11) concerns fixed-design sampling variance; Equation (6) concerns conditional Gaussian uncertainty.

For a previously unobserved policy with a fixed residual prior independent of other parameters, other-policy data leave that residual uncertainty unchanged. Direct observations can create posterior dependence, after which this separation need not hold. Thus reference observations can inform several comparisons while candidate observations resolve policy-specific uncertainty. The conditional variance identity motivates the acquisition approximation in Section 3.3; the observation vector and noise term must match the measurement being valued.

D Terminal-Bench Reproduction and Statistical Details

D.1 Frozen Data and Execution Versions

The study fixes the supplementary benchmark archive at revision `a9ea4fe7a120a4ab1d2b3af99c665ef3f815ad00`, under `terminal-bench-2.1/router-random-study/`. Table 5 records the execution versions associated with the released benchmark comparison.

Table 5: Frozen benchmark execution metadata.

Item	Frozen value
Benchmark source	`9fa7a4c9973164bf2cb9b5f7339476b365888254`
Task set	`terminal-bench/terminal-bench-2-1`
Task-set reference	`sha256:7d7bdc1cbedad549fc1140404bd4dc45e5fd0ea7c4186773687d177ad3a0699a`
Harness	Harbor 0.18.0; Terminus 2
Interaction cap	24 rounds
Raw tasks / trials	89 / 1,691
Common-valid tasks / trials	80 / 1,520

Table 6: Frozen nominal prices in US dollars per million tokens.

Model	Uncached input	Cache read	Output
gpt-5.6-sol	5.00	0.50	30.00
kimi-k3	3.00	0.30	15.00
deepseek-v4-flash-0731	0.14	0.028	0.28
deepseek-v4-pro-0813	0.435	0.003625	0.87

Each replicate uses the common frozen task version, permits one attempt, records streaming usage, and has no separate output-token cap. Concurrency is four for baseline and BitRouter groups and six for random routing. The baseline and G1/G3 GPT tier use `gpt-5.6-sol` at high reasoning effort; `kimi-k3`, `deepseek-v4-flash-0731`, and `deepseek-v4-pro-0813` use max effort. Random routing samples the G1 pool independently per request with GPT/Kimi/Flash weights 1510/3027/200 (approximately 31.88%/63.90%/4.22%).

Original Harbor outcomes and trajectories are the records for rewards and exceptions. Request ledgers and attribution tables associate measured usage with those trials; derived tables do not supersede the original records. The archived study description and reproduction instructions specify the released materials. The common-valid aggregates are recorded in the frozen comparison file.

D.2 Usage Attribution and Nominal Prices

Usage is associated through Harbor trajectories, request ledgers restricted to run boundaries, and provider usage supplements where needed. G1 and random-routing cloud requests use request identifiers for matching. The baseline uses constrained one-to-one associations between in-run ledger entries and trajectory steps. Source and attribution-quality flags are retained. Estimated G3 DeepSeek Pro usage retains its source flag and the aggregate marker in Table 3.

Table 6 gives the historical prices in the frozen group registry; these prices are fixed study parameters. With total input tokens I, cache-read tokens R, and output tokens O including reasoning, the nominal cost of an eligible request is

$$C^{\text{nom}} = \frac{(I - R)p_{\text{in}} + Rp_{\text{cache}} + Op_{\text{out}}}{10^6}. \tag{15}$$

Eligible requests from all included trials contribute, irrespective of task success.

Random runs R1 and R2 share a request-ledger boundary, and the R2 database contains cumulative records. Published request associations determine group and trial membership. The reproduction pipeline uses the released, redacted evidence and usage-attribution records.

D.3 Replicates, Exclusions, and All-Task Outcomes

Pooling B1–B4 gives four baseline replicates per task. G1 and G3 each have five; pooling R1–R5 gives five random-routing replicates. The common-valid set requires every required replicate in every compared group to have no top-level exception. The nine excluded tasks are `adaptive-rejection-sampler`, `break-filter-js-from-html`, `gcode-to-text`, `gpt2-codegolf`, `raman-fitting`, `regex-chess`, `torch-pipeline-parallelism`, `vulnerable-secret`, and `write-compressor`. They remain in Table 7, which retains recorded native rewards, including rewards available in trials with top-level exceptions; unavailable rewards contribute zero.

The five recorded random-routing seeds, in run order, are `3185662293243100039`, `5682178942122491418`, `5079988043586760223`, `2818348522960731342`, and `7936714200945624811`. Sampling weights remain 1510/3027/200, although realized selections and later trajectories vary.

Table 7: All 89 tasks under the original Harbor denominators, retaining trials with top-level exceptions. Success outcomes use the full trial counts; cost comparisons use the common-valid set in Table 3.

Configuration	Trials	Successes	Success (%)	Trials with exceptions
Strong model	356	284	79.78	8
BitRouter G1	445	353	79.33	5
Same-pool random	445	333	74.83	8
BitRouter G3	445	328	73.71	10

D.4 Paired Task-Cluster Bootstrap

The analysis uses `common-valid-per-trial.jsonl` and `common-valid-comparison.json` from the fixed archive revision in D.1. It checks the 80-task panel, required within-task replication, successes, and mean nominal costs for all four groups against the published aggregates. Outcomes and costs are averaged within each task and group. The same 80 task indices are then sampled with replacement for all groups in each of 20,000 draws, using Python's `random.Random(20260923).choices`. Intervals are the 2.5th and 97.5th percentiles, with linear interpolation at index $(B-1)p$ among the B sorted draws. Each draw computes cost reduction as $100(1-\overline{C}_{G1}/\overline{C}_{\text{control}})$, preserving the ratio-of-means estimand.

The analysis script checks the submission-copy input hashes before calculation, and the machine-readable results contain the estimates and unrounded intervals. The original frozen input SHA-256 hashes are listed below; the supplementary manifest records hashes of the anonymized submission copies and their transformations:

- `common-valid-per-trial.jsonl`: `a02788e34a14c214e2d4efd9f1b91b55560d294dc6576f1cc6c29e3ce86608e3`.
- `common-valid-comparison.json`: `727dc723f6b6f609a9d498c4fc7bd720b91d68aa95f6cecec97bb670748cc07b`.

Task-level resampling preserves the observed replicate means and the pairing of tasks across groups. The statistical scope of these exploratory intervals is discussed in Section 6.