EXACT FRONTIER COUPLING FOR ADAPTIVE LANGUAGE MODEL POLICY EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Independent evaluation of adaptive language-model agents often repeats identical model requests across policies. We introduce Exact Frontier Coupling (EFC), a joint evaluator that shares these requests while preserving each policy's standalone trajectory distribution. At a history shared by active policies, EFC groups identical conditional action distributions and solves a local multi-marginal optimal transport problem to minimize expected distinct-request cost. We prove marginal trajectory preservation under matched response kernels. On 1,200 constructed frontiers, EFC reduces expected weighted request cost by 1.65% relative to common-quantile coupling optimized over every action order. In paired evaluations of the same 52 BFCL tasks across four models, exact-request sharing reduces physical model calls by 29–44% and yields 1.66–1.82× warmed-worker speedups over independent execution, with identical observed trajectories and official outcomes. The resulting savings make it feasible to evaluate more adaptive agents across more tasks under the same inference budget.

1 INTRODUCTION

Evaluating a bank of language-model agents often means running several feedback policies on the same tasks. At the beginning of an episode, multiple policies may request precisely the same model completion; after observing tool feedback, they can share a history while differing in their next-action laws. Issuing each identical inference once can reduce serving work. Choosing how their stochastic actions are coupled is a different problem: the joint draw determines which requests coincide now and which histories remain shared later. Each policy must still have the action and observation distribution it would have had alone.

We formalize a shared-history *frontier*, where policies have an identical complete observed history but possibly different next-action laws. Exact Frontier Coupling (EFC) optimizes the expected number of distinct charged request identities at that frontier, subject to all policy marginals. Duplicate action laws collapse to a smaller multi-marginal linear program (LP); a secondary objective favors equal actions among primary-cost ties. We also characterize the primary optimum by a request-set LP whose size depends on distinct request keys, rather than the number of policy tuples. The executor coalesces only complete operational keys, including decoding settings and tool state, so the marginal-preservation statement applies to executable requests rather than semantic resemblance. Prior grand-coupling and transport work motivates the local set-cost objective (Ding et al., 2026; Sun et al., 2023; Pass, 2015); our contribution is its operational instantiation and evaluation for adaptive policy banks.

We evaluate the framework at three complementary levels. A prespecified constructed-law study finds a mean weighted-cost reduction of 1.65% against the best quantile action ordering. A feedback-adaptive calculator pilot compares complete episodes and policy selections against optimized quantile and Tree-Coupled A/B Testing (TCAB) (Wen, 2026). Paired BFCL serving experiments across four models yield 29–44% fewer physical calls than independent serving with identical observed trajectories and outcomes. A separate BFCL cohort and an AgentDojo development study extend the execution comparison. Together, these experiments connect the local mathematical objective to complete-episode requests and measured serving work.

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.

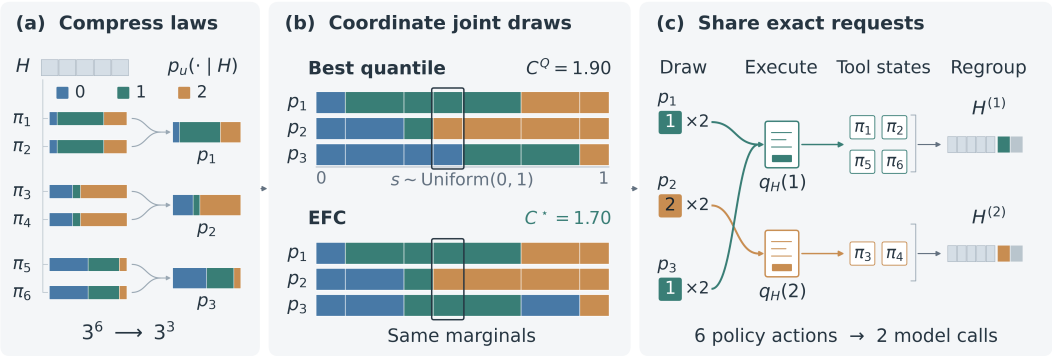


Figure 1: **EFC at a shared-history frontier.** (a) Six policies share three distinct action laws. Compression reduces 3^6 joint assignments to 3^3. (b) Strip widths encode the Section 3 example's probabilities. EFC preserves each marginal and lowers expected request cost from the best quantile ordering's 1.90 to 1.70. Outlined slices show joint draws. (c) The illustrated EFC draw yields six policy actions and two model calls. Shared responses enter isolated tool states; identical updated histories regroup. The guard selects a candidate for at least 2% expected cost improvement over TCAB; otherwise it uses TCAB.

2 Joint Evaluation of Adaptive Policies

Let $\Pi = \{\pi_1, \ldots, \pi_N\}$ be policies with complete observed histories $H_{i,t}$ and action laws $p_i(\cdot \mid H_{i,t})$ on a finite action set $\mathcal{A}$ containing free STOP. Let G index the active policies at a shared history H. A joint action law γ over $Z = (Z_i)_{i \in G}$ must preserve every coordinate marginal:

$$\Gamma(H, G) = \left\{ \gamma \in \Delta(\mathcal{A}^{|G|}) : \sum_{z : z_i = a} \gamma(z) = p_i(a \mid H),\ \forall i \in G,\ a \in \mathcal{A} \right\}. \tag{1}$$

Here $\Delta(S)$ is the set of probability distributions on S. The map $q_H(a)$ specifies a *complete* submitted request identity, including messages or prompt tokens, model revision, decoding and sampling configuration, and tool/schema state; $q_H(\text{STOP}) = \perp$. Sharing requires exact request equivalence and matching conditional backend response kernels, as assumed in Theorem 1. The serving experiments check equality of observed responses and trajectories across execution modes. The immediate charged-request cost is

$$c_H(z) = |\{q_H(z_i) : i \in G,\ q_H(z_i) \neq \perp\}|. \tag{2}$$

The executor issues one request per distinct key, copies its response to all requesting coordinates, transitions their tools, and regroups exact histories. Complete-episode *logical requests* sum these distinct keys over visited frontiers; *physical calls* count actual model invocations after any execution-level reuse. In the calculator-agent experiment, each non-STOP action at a fixed history has its own key, so $c_H(z) = \sum_{a \neq \text{STOP}} \mathbf{1}\{\exists i : z_i = a\}$. The formulation also supports fixed positive per-key weights: each selected key contributes its weight once. Unit weights measure request counts, while the constructed-law study uses weighted costs to represent requests with heterogeneous costs.

The request boundary. Request identity links the mathematical cost to executable work. The same action label after different feedback can encode different prompts, and the same prompt under different sampling parameters is a different request. Conversely, policies with the same complete key can use one returned model output while retaining separate agent states. Coupling determines which keys are selected together; the executor then serves each selected key once. Equation (2) prices the realized request set, and the serving experiments measure physical calls, tokens, and worker time for the resulting requests.

For example, agents with identical prompts and decoding settings share one generation but apply its tool call in isolated environments. After a schema error, repair and ordinary-continuation policies produce different request keys. Regrouping complete histories after each observation makes reuse responsive to this feedback (Figure 1).

3 EXACT FRONTIER COUPLING

3.1 A LOCAL MULTI-MARGINAL TRANSPORT PROBLEM

EFC minimizes expected immediate request cost over the complete coupling polytope:

$$C^\star(H,G) = \min_{\gamma \in \Gamma(H,G)} \sum_{z \in \mathcal{A}^{|G|}} \gamma(z) c_H(z). \tag{3}$$

This is a finite linear program (LP). Its variables are probability masses on joint action assignments; its equality constraints enforce normalization and every policy marginal. The product coupling proves feasibility, and finiteness gives existence. In contrast to maximizing a sum of pairwise agreements, the objective directly prices a realized *set* of request identities. This distinction appears when three or more different marginal laws compete for the same few actions.

A three-policy example. Figure 1 uses three request-producing actions with marginals $p_1 = (.1, .6, .3)$, $p_2 = (.3, .1, .6)$, and $p_3 = (.5, .4, .1)$. Any coupling has expected union size at least $\sum_a \max_i p_i(a) = .5 + .6 + .6 = 1.7$. The joint masses $.1, .2, .1, .3, .2, .1$ on assignments $(0,0,0)$, $(1,0,0)$, $(1,1,1)$, $(1,2,1)$, $(2,2,0)$, and $(2,2,2)$ attain this bound. Even optimizing over every spanning-tree topology and root, standard independent-residual maximal-edge gluing has minimum cost 1.82; the deterministic best-root TCAB routine used in our experiments gives $1.833\overline{3}$ on the displayed policy order. The exhibited EFC joint simultaneously attains every pairwise maximal agreement probability. The gap therefore comes from Markov-tree gluing: maximal edge couplings need not recover the union-cost-optimal higher-order joint.

Duplicate-law compression. Many policy banks contain candidates that react differently in some states but share the same conditional law at the current history. Partition G into U classes of identical rows $p_u(\cdot \mid H)$ with multiplicities n_u. EFC solves Equation (3) over one representative per class, reducing $|\mathcal{A}|^{|G|}$ joint assignments to $|\mathcal{A}|^U$. The sampled representative action is copied to every member of its class. Appendix A proves that this preserves every marginal and exactly preserves the minimum union cost.

A request-set view of the primary optimum. The tuple LP still grows exponentially in U. For the primary cost alone, let $\mathcal{K}$ contain the K distinct charged keys at H, aggregate each marginal to $p_u(k) = \sum_{a:q_H(a)=k} p_u(a \mid H)$, and let $p_u(\perp)$ collect free actions. Write $p_u(B) = \sum_{k \in B} p_u(k)$. For a monotone set cost c (cardinality in the agent experiments and positively weighted cardinality in the constructed study), the same optimum can be written with one mass λ_R for each $R \subseteq \mathcal{K}$:

$$\begin{aligned} \min_{\lambda \geq 0} \quad & \sum_{R \subseteq \mathcal{K}} c(R) \lambda_R \\ \text{s.t.} \quad & \sum_R \lambda_R = 1, \\ & \sum_{R \subseteq B} \lambda_R \leq \min_u \{p_u(\perp) + p_u(B)\}, \quad B \subsetneq \mathcal{K}. \end{aligned} \tag{4}$$

These 2^K variables encode a latent set of eligible requests. The subset inequalities admit a conditional assignment of one key or $\perp$ to every policy by a fractional Hall/max-flow argument (Artstein, 1983); the actual realized key union is contained in the latent set. Conversely, any tuple coupling induces feasible set masses with the same cost, proving equality of primary optimal values. An original action is recovered conditionally on its selected key, so aliases retain their marginals. Appendix B.3 gives the lifting proof and numerical validation. This formulation characterizes primary cost; the evaluated implementation uses the tuple LP to optimize both cost and the tie-break below. Related reduced grand-coupling programs appear in Ding et al. (2026).

A future-sharing tie-break. Several couplings can attain $C^\star$. Among them, we minimize

$$\mathbb{E}_\gamma \left[\sum_{u<v} n_u n_v \mathbf{1}\{Z_u \neq Z_v\} \right] \quad \text{subject to} \quad \mathbb{E}_\gamma[c_H(Z)] = C^\star(H,G). \tag{5}$$

This lexicographic LP minimizes the expected number of disagreeing original policy pairs while retaining the minimum immediate request cost. Favoring agreement serves as a heuristic for keeping more policies at shared histories after their next action.

3.2 GUARDED EXECUTION

Let γ^T be the best-root TCAB coupling and $C^T = \mathbb{E}_{\gamma^T}[c_H(Z)]$. Our baseline follows TCAB's total-variation minimum-spanning-tree construction with pairwise maximal edge couplings (Wen, 2026), and enumerates every policy as root. Let γ^C be the candidate coupling, with cost C^C: it is $\gamma^\star$ whenever exact enumeration is within the cap and a cost-aware common-quantile candidate otherwise. The evaluated method uses threshold $\eta = 0.02$:

$$\gamma^\eta(H, G) = \begin{cases} \gamma^C, & C^T > 0 \text{ and } (C^T - C^C)/C^T \geq \eta, \\ \gamma^T, & \text{otherwise.} \end{cases} \tag{6}$$

On a common frontier, fallback uses the same TCAB plan and state-keyed random tape as the TCAB baseline. The guard avoids changing downstream joint histories for negligible conditional improvements.

With $A = |\mathcal{A}|$ and U distinct conditional laws, the collapsed LP has A^U variables and $1 + U(A - 1)$ independent normalization/marginal constraints; the secondary LP adds one cost equality. We enumerate at most 100,000 joint assignments; beyond that cap, the implementation constructs a cost-aware quantile candidate and applies the same TCAB guard. The calculator-agent experiment has $A = 5, U \leq 4$, and therefore at most 625 assignments, within the cap throughout. "Exact" refers to optimizing the complete finite LP with HiGHS through SciPy (Huangfu & Hall, 2018; Virtanen et al., 2020) to the audited numerical tolerances in Appendix B.

Exact history grouping is repeated after every observation, so the optimization size follows the *distinct conditional laws within each shared group*, rather than the total number of policies. For eight policies and five actions, direct enumeration would contain $5^8 = 390{,}625$ tuples; collapsing to four laws reduces this to $5^4 = 625$. Both EFC and TCAB cache plans by their marginal and cost arrays. A cached plan therefore solves the same local problem and retains the same marginal constraints when reused at another occurrence.

4 PROPERTIES

Proposition 1 (Conditional local optimality)**.** *At a fixed shared-history frontier (H, G) within the enumeration cap, the EFC candidate attains the minimum conditional expected immediate request cost over all couplings in $\Gamma(H, G)$. In particular, $C^\star(H, G) \leq C^T(H, G)$. At every frontier, the guarded plan has conditional cost no greater than C^T.*

The first statement is Equation (3); the second follows because TCAB is feasible. Equation (6) executes a candidate only when its cost is at most $(1 - \eta)C^T$, and otherwise executes TCAB (including when $C^T = 0$). The proposition compares feasible couplings of the *same current frontier*; Section 6 separately measures their complete-episode request costs.

Theorem 1 (Marginal trajectory preservation)**.** *From the same initial states as standalone execution, suppose frontier plans are chosen predictably from observed joint history and have coordinate marginals $p_i(\cdot \mid H_i)$. Conditional on the joint pre-action history and sampled actions, assume each coordinate's model-output and environment-transition kernel equals its standalone kernel given its own history and action. Then each policy's marginal trajectory distribution under coupled execution equals its standalone trajectory distribution.*

Conditioned on the joint pre-action history, coordinate i has precisely the prescribed action law. The kernel-matching assumption gives the same conditional output and transition law as standalone execution. Induction over decision depth yields the claim; a full proof appears in Appendix A.

In particular, any integrable reward computed from one policy's trajectory retains its standalone expectation. Expected reward differences between candidates are consequently preserved as well. This connects the trajectory guarantee to policy evaluation: the evaluator changes how requests are sampled together and executed, while retaining the expected scores of the candidate policies being compared. The paired selection rule in Section 5 acts on their observed task outcomes.

5 EXPERIMENTS

Constructed-law local comparison. We prespecified six families of four distinct categorical action laws, each duplicated across eight policies: five or seven actions crossed with balanced, sparse, or nearly identical marginals. Across 200 independently generated matrices per family, we compare the exact LP with the best common-quantile plan over *all* action orders. The last action is free and charged actions receive fixed, independently sampled weights. For each matrix, the relative saving is $100(C^{Q} - C^{\star})/C^{Q}$, where C^{Q} is the minimum expected weighted cost over quantile orderings. We average these paired percentages with equal weight per family. This controlled comparison evaluates the coupling objective directly on constructed distributions.

BFCL serving endpoint. We use 52 public Berkeley Function Calling Leaderboard (BFCL-V4) multi-turn development tasks (Patil et al., 2025): thirteen previously inspected template roots crossed with four variants. All assigned outcomes remain in the comparison. Three Qwen3 sizes and ToolACE-2-8B provide four models (Yang et al., 2025; Liu et al., 2025b). ACT follows the model's native tool-use behavior; SCHEMA makes one schema-oriented attempt after its first *observed* tool error, then resumes ACT. Each policy runs in an isolated official tool world. On the *same* warmed worker and model, independent mode executes every logical generation and shared mode reuses identical complete request keys; mode order alternates by four-task wave. Ordered keys, generated bytes, token counts, termination reasons, and official scores must agree across modes. We report physical calls, input-plus-output tokens, and worker time after loading, with task-local distinct-key counts as a count-only memoization control. A separate, previously model-scored 64-task cohort tests the same parity condition. Benchmark versions, decoding settings, and scoring details appear in Appendices I and J.

Calculator-agent endpoint. Eight stochastic feedback policies on GSM8K (Cobbe et al., 2021) choose among four model/tool requests or free STOP for at most three steps. Qwen2.5-7B-Instruct and Mistral-7B-Instruct-v0.3 serve the same 32 training tasks per backbone (Qwen et al., 2024; Jiang et al., 2023; Mistral AI, 2024). An original single-run screen compared guarded EFC with TCAB, fixed-order quantile, and independent action. A subsequent frozen three-run pilot on the *same task IDs* added exhaustive minimum-cost common quantile and a fragmentation-aware cost-tied variant. Every arm coalesces matching realized requests. A prospective joint gate required identical selected policies plus at least 5% pooled and 2% per-backbone EFC savings, with positive simultaneous one-sided 90% task-cluster bootstrap lower bounds against each control. Appendices C and E give the full selection rule and paired analysis.

The strongest quantile control maps a single uniform draw through every policy's categorical cumulative distribution under a shared action ordering. We enumerate every supported ordering and minimize conditional expected distinct-key cost; the fragmentation-aware variant breaks cost ties by expected pair disagreement. TCAB uses the best-root tree coupling from Section 3. All controls share matching realized requests just as EFC does. The pilot therefore compares *action couplings*, while the independent-versus-shared BFCL experiment compares *execution* of fixed policy trajectories.

The eight policies combine four after-success response profiles with two after-error profiles. All begin with a direct calculator request, then condition their action probabilities on the observed tool result, whether subsequent values agree, and remaining steps. To assess policy selection, each model has a protected incumbent; challengers must improve mean reward and pass a paired exact test on discordant task successes with a Holm correction over seven comparisons (Holm, 1979). The same rule is applied separately within each arm, making selection agreement an explicitly measured outcome.

AgentDojo user-task endpoint. We evaluate Qwen3-8B on public AgentDojo development tasks (Debenedetti et al., 2024), using ACT and first-error SCHEMA in both modes and the official user-task success measure. A five-task engineering check preceded a post hoc 47-task evaluation on ten shards. Full-cohort admission requires complete episodes, output bytes, final states, and official outcomes to match for *every* shard (Appendix K).

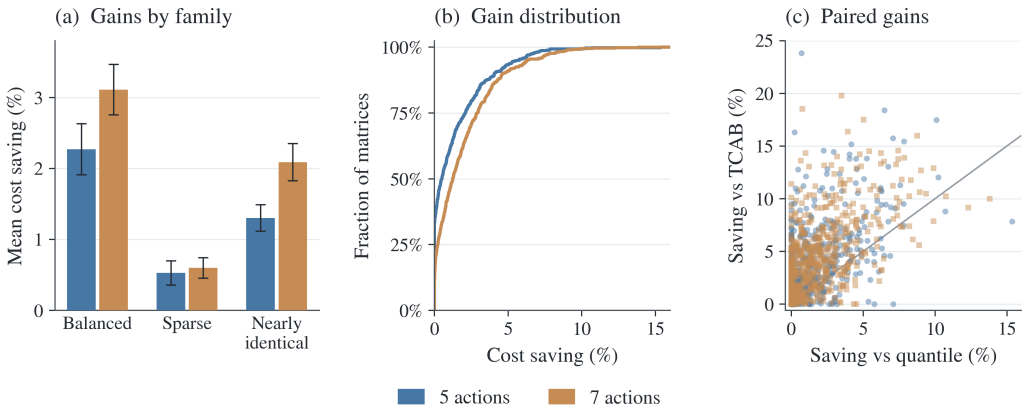


Figure 2: Expected one-frontier weighted-cost savings on all 1,200 constructed-law matrices. (a) Mean saving against exhaustively optimized quantile, with separate paired-t 95% intervals over 200 matrices per family. (b) Empirical cumulative distributions, pooling the three equally sized families for each action count. (c) Each matrix's relative saving against quantile and TCAB; the diagonal denotes equal relative gains.

6 Results

6.1 One-frontier mechanism and complete-episode comparison

Constructed action laws. EFC reduced expected one-frontier weighted request cost by 1.65% on average relative to exhaustively optimized quantile (stratified paired-bootstrap 95% interval [1.54%, 1.76%]). It strictly improved cost on 989 of 1,200 matrices (82.4%), with independent LP certificates agreeing throughout. Family means ranged from 0.53% to 3.11%, largest for seven-action balanced laws. Against best-root TCAB, the mean reduction was 3.88% [3.70%, 4.06%]. Figure 2 shows the family estimates and gain distributions. Optimizing quantile ordering separately for every matrix isolates the additional flexibility of joint coupling.

Matched calculator-agent episodes. In the three-run calculator pilot, guarded EFC used 533 pooled logical requests versus TCAB's 600: an 11.17% reduction, with a simultaneous one-sided 90% lower bound of 7.89%. Its total matched fragmentation-aware quantile and was 3.09% higher than optimized quantile's 517 requests. All arms selected the same protected incumbent within each model and run. Table 1 reports every predeclared control on the same eight policies and 32 tasks per backbone.

The earlier single-run screen used 7.9% fewer requests than TCAB on each backbone and 48.3% fewer pooled requests than independent action, also with matched protected selections. Its comparator set and run count differ from the subsequent pilot, so we analyze the screens separately (Appendices D and E).

Table 1: Complete-episode calculator pilot with three fresh runs on the same task IDs. Lower logical-request counts are better. For each baseline B, EFC's saving is $100(R_B - 533)/R_B$; positive values favor EFC. Lower bounds are simultaneous one-sided 90% task-cluster bootstrap bounds.

	Logical requests ↓			EFC saving vs. baseline (%)	
Method	Qwen	Mistral	Total	Estimate	Lower bound
Guarded EFC	256	277	533	–	–
Optimized quantile	256	261	517	−3.09	−8.30
Fragmentation-aware quantile	254	279	533	0.00	−3.64
Best-root TCAB	276	324	600	+11.17	+7.89

Table 2: Exact-request sharing versus independent serving on BFCL models with matched trajectories and official outcomes. I/S denotes independent/shared physical calls. Savings use independent totals as the denominator; the time ratio is independent/shared warmed-worker seconds. All assigned tasks remain in each comparison.

Cohort	Model	Calls I/S	Calls saved	Tokens saved	Time ratio ↑
BFCL52	Qwen3-4B	1,061/687	35.25%	32.72%	1.670×
BFCL52	Qwen3-8B	1,066/752	29.46%	26.37%	1.743×
BFCL52	Qwen3-14B	1,060/747	29.53%	26.63%	1.816×
BFCL52	ToolACE-2-8B	2,682/1,500	44.07%	42.56%	1.665×
BFCL64	ToolACE-2-8B	4,740/2,981	37.11%	36.87%	1.510×

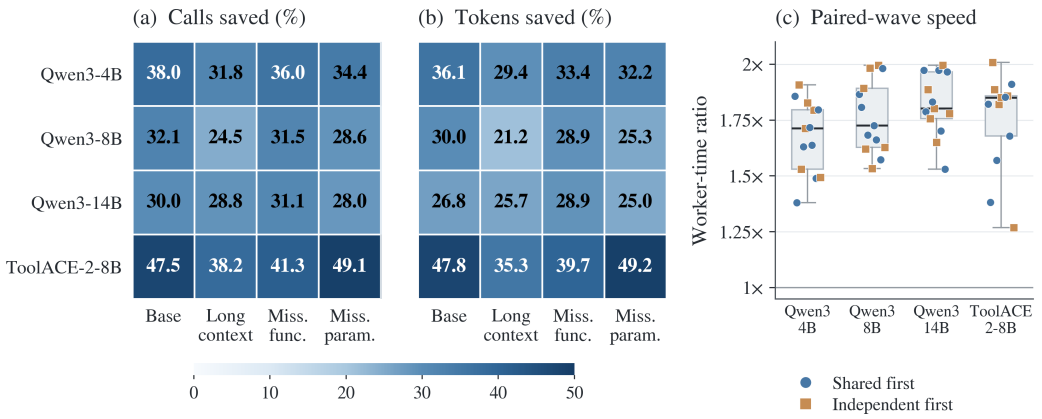


Figure 3: Exact-request serving on the four BFCL52 models with matched trajectories and outcomes. (a,b) Physical call and input-plus-output token savings relative to independent serving, computed from totals over 13 tasks per variant. (c) Independent/shared warmed-worker time for all thirteen four-task waves per model; markers identify the mode run first. Boxes show medians, interquartile ranges, and full-range whiskers across these paired waves.

6.2 Exact-request reuse on BFCL tool agents

Across the four BFCL52 models, exact-request sharing saved 29.46–44.07% of physical calls and 26.37–42.56% of input-plus-output tokens relative to independent serving, with warmed-worker speedups of 1.66–1.82× (Table 2). Every ordered logical request, generated byte sequence, task–policy trajectory, and official score matched across modes. All four models evaluated the same 52 tasks, and every assigned outcome remains in the totals, including unsuccessful and context-limited episodes.

Every BFCL52 task saved calls and tokens, and all thirteen paired four-task waves per model finished sooner under sharing. Figure 3 shows gains across base, long-context, missing-function, and missing-parameter tasks. Call savings range from 24.53% to 49.13% across the sixteen model-by-variant comparisons, and both serving-order groups have time ratios above one. Calls and tokens quantify complementary aspects of reuse: each repeated generation saves one invocation, while its prompt and response lengths determine the associated token saving. The measurements retain identical policy outputs and task outcomes, so the reduction comes from serving repeated requests once rather than shortening the evaluated trajectories. Appendix I reports template-cluster intervals, official outcome counts, and all paired timing waves.

The task-local distinct-key control exactly matched the shared call counts for every model, confirming that the executor realized the reuse identified in the independent request traces. These frontiers had deterministic action laws and at most two charged keys, where optimized quantile and EFC

attain the same immediate cost. The BFCL experiment thus measures the execution-level benefit of exact-request sharing, complementing the constructed-law comparison of coupling optimizers.

Distinct 64-task development cohort. On the separate 64-task cohort, ToolACE-2-8B reduced physical calls from 4,740 to 2,981 (37.11%) and tokens by 36.87%, while warmed worker time fell from 821.40 to 544.00 seconds (1.51× speedup). All 128 policy episodes per mode contributed to these totals with matched trajectories and official outcomes. Of the four planned model comparisons, ToolACE alone met the output-parity requirement; the three Qwen comparisons are excluded from paired savings estimates. Appendix J provides the model-level admission accounting, task-level breakdown, and uncertainty calculation.

AgentDojo user-task parity. On nine completed shards covering 44 of the 47 assigned AgentDojo tasks, shared serving reduced physical calls from 373 to 204 (45.31%) and physical tokens from 1,725,721 to 936,871 (45.71%). All 88 task–policy episodes per mode matched in outputs, final states, and official outcomes. Summed warmed intervals across nine distinct workers were 193.79 versus 112.38 seconds, a 1.724× ratio. Each policy succeeded on 21 of 44 tasks; SCHEMA intervened after observed feedback on four. These totals are conditional on the nine completed shards: the remaining three-task shard is excluded because one task reached the configured context/prompt bound in both modes.

Observed frontier structure. The controlled frontier audit identifies where joint optimization improves on tree coupling. All 297 cases passed checks of the marginals, both LP stages, the guard, and fallback draws. After successful calculator feedback, the sixteen four-policy subsets containing four distinct laws each achieved a 21.91% local cost reduction relative to TCAB. Among 48 three-law subsets, gains ranged from zero to 21.11% according to which laws were combined; six two-law subsets tied TCAB. In the original model screen, the guard selected an exact plan at 50 of 183 visited frontiers and reused TCAB elsewhere. Optimized quantile matched the LP's immediate cost on those visited states. The two controlled studies therefore address different comparisons: the calculator-bank audit explains the gap from tree coupling, while the constructed-law study establishes additional gains over optimized quantile (Appendix F).

7 Related work

Coupling and joint cost. Common random numbers couple simulations to control evaluation variance while retaining their marginals (Glasserman & Yao, 1992; Thorisson, 2000). TCAB uses a tree of pairwise maximal couplings to share adaptive feedback while retaining standalone policy behavior (Wen, 2026). EFC optimizes over joint laws beyond that tree family and directly prices the set of exact model requests at a shared history. Related transport formulations address other computational objectives: SpecTr optimizes a set-membership cost for speculative decoding (Sun et al., 2023); Ding et al. (2026) minimize the number of distinct grand-coupling outcomes with a reduced LP in graph coloring; and Phan et al. (2026) study multi-marginal coupling for Metropolis–Hastings. Building on these foundations, our formulation integrates operational request identity, history regrouping, free actions, and executable output sharing for adaptive language-model policy evaluation. The request-set characterization and duplicate-law compression expose the local optimization structure of this setting.

Serving and evaluation. Exact-key memoization, prefix/prompt caching, vLLM serving, and speculative decoding target different reuse units or decoding operations (Gim et al., 2024; Kwon et al., 2023; Sun et al., 2023). Our executor shares complete matching requests across policy instances; the coupling layer determines which requests occur together. The paired serving studies evaluate this execution-level reuse through exact trajectory checks and measured physical work. Off-policy evaluation estimates a target policy from other data, while candidate elimination controls evaluation allocation (Jiang & Li, 2016; Thomas et al., 2015; Liu et al., 2025a; Li et al., 2018). Here, candidate policies execute their own feedback-dependent trajectories under a marginal-preserving coupling, and a protected selector compares the resulting scores. Tool-using agents and benchmarks motivate this setting (Yao et al., 2023b; Schick et al., 2023; Yao et al., 2023a; Liu et al., 2024), with BFCL and AgentDojo supplying the public environments used in our serving studies.

8 LIMITATIONS AND SCOPE

The results identify two complementary sources of efficiency. Heterogeneous conditional action laws over overlapping requests create opportunities for joint optimization; an exact-key executor realizes reuse whenever policies select matching requests. The diversity of laws and charged keys at shared histories therefore provides a practical diagnostic for choosing an evaluator. The serving experiments retain each policy's full logical workload and assess the resulting physical work under matched trajectories.

The optimality guarantee applies to one common frontier. Couplings can change subsequent shared histories, so complete-episode cost remains an empirical question. The calculator pilot improved on TCAB and retained the same protected selections, while optimized quantile used fewer requests; the joint continuation gate was not met and the confirmation study was not run. Trajectory preservation retains marginal score expectations while allowing cross-policy dependence, so it does not imply an unchanged joint score distribution, selection probabilities, or pathwise outcomes. Establishing an optimizer-specific advantage in full episodes or selection quality requires broader stochastic policy banks and held-out tasks.

The serving studies evaluate development cohorts under explicit output-parity requirements. On deterministic BFCL frontiers, EFC and optimized quantile have equal local cost, and task-local memoization matches shared call counts. The reported serving savings therefore quantify execution-level reuse. BFCL64 contributes one admitted model comparison; AgentDojo contributes nine completed shards covering 44 tasks. Appendices I–K detail admission accounting, official outcomes, and effective decoding configurations. Timing ratios apply to the studied workloads on warmed workers, with one paired observation per wave or shard. Larger policy banks and broader deployment settings motivate scalable planning beyond tuple enumeration, repeated timing trials, and continued verification of backend response equivalence.

9 CONCLUSION

Exact Frontier Coupling minimizes expected request-set cost at a shared-history frontier while preserving each policy's standalone trajectory distribution. Duplicate-law compression and the request-set characterization make the local optimization structure explicit. The experiments establish a positive local gain over optimized quantile and substantial physical call savings from exact-request sharing, including 29–44% reductions across four BFCL models. By combining distribution-preserving coupling with executable request reuse, the framework provides a principled basis for evaluating adaptive language-model policies with less redundant inference.

ETHICS STATEMENT

The study uses public benchmark questions and openly distributed model weights; it involves no human subjects or private user data. Tool-use evaluations run in benchmark environments with isolated policy state. The principal resource cost is model inference, which we report through request counts, token counts, and measured worker time.

REPRODUCIBILITY STATEMENT

Appendix C specifies the policy bank, prompts, solver, and selector. Appendices E, G, I, J, and K provide the pilot, numerical checks, serving protocols, and AgentDojo admission accounting. The anonymous source archive contains the manuscript, bibliography, required styles, and PNG figures. Numerical inputs, plotting scripts, execution journals, and audit programs are retained as separate research records.

REFERENCES

Zvi Artstein. Distributions of random sets and random selections. *Israel Journal of Mathematics*, 46(4):313–324, 1983. doi: 10.1007/BF02762891. URL https://link.springer.com/article/10.1007/BF02762891.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2636. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html.

Tianxing Ding, Hongyang Liu, Yitong Yin, and Can Zhou. Tight bounds for sampling q-colorings via coupling from the past. In *53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026)*, volume 374 of *Leibniz International Proceedings in Informatics*, pp. 76:1–76:24, 2026. doi: 10.4230/LIPIcs.ICALP.2026.76. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2026.76.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. In *Proceedings of Machine Learning and Systems*, volume 6, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/a66caa1703fe34705a4368c3014c1966-Abstract-Conference.html.

Paul Glasserman and David D. Yao. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992. doi: 10.1287/mnsc.38.6.884. URL https://pubsonline.informs.org/doi/10.1287/mnsc.38.6.884.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

Qi Huangfu and J. A. Julian Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018. doi: 10.1007/s12532-017-0130-5. URL https://doi.org/10.1007/s12532-017-0130-5.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. doi: 10.48550/arXiv.2310.06825. URL https://arxiv.org/abs/2310.06825.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 652–661. PMLR, 2016. URL https://proceedings.mlr.press/v48/jiang16.html.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018. URL https://jmlr.org/papers/v18/16-558.html.

Shuze Daniel Liu, Claire Chen, and Shangtong Zhang. Efficient multi-policy evaluation for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18951–18959, 2025a. doi: 10.1609/aaai.v39i18.34086. URL https://ojs.aaai.org/index.php/AAAI/article/view/34086.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, et al. ToolACE: Winning the points of LLM function calling. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/663865ea167425c6c562cb0b6bcf76c7-Abstract-Conference.html.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Mistral AI. Model card for Mistral-7B-Instruct-v0.3. Hugging Face, 2024. URL https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3.

Brendan Pass. Multi-marginal optimal transport: Theory and applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1771–1790, 2015. doi: 10.1051/m2an/2015020. URL https://www.numdam.org/articles/10.1051/m2an/2015020/.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392. PMLR, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Buu Phan, Gergely Flamich, Ashish Khisti, and Shahab Asoodeh. Multi-marginal couplings for Metropolis–Hastings. *arXiv preprint arXiv:2605.12807*, 2026. doi: 10.48550/arXiv.2605.12807. URL https://arxiv.org/abs/2605.12807.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023. doi: 10.52202/075280-2997. URL https://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. SpecTr: Fast speculative decoding via optimal transport. In *Advances in Neural Information Processing Systems*, volume 36, pp. 30222–30242, 2023. doi: 10.52202/075280-1314. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6034a661584af6c28fd97a6f23e56c0a-Abstract-Conference.html.

Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pp. 3000–3006, 2015. doi: 10.1609/aaai.v29i1.9541. URL https://ojs.aaai.org/index.php/AAAI/article/view/9541.

Hermann Thorisson. *Coupling, Stationarity, and Regeneration*. Springer, New York, 2000. doi: 10.1007/978-1-4612-1236-2. URL https://doi.org/10.1007/978-1-4612-1236-2.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020. doi: 10.1038/s41592-019-0686-2. URL https://doi.org/10.1038/s41592-019-0686-2.

Yuxiao Wen. Fast A/B/n testing: Exact multi-policy comparison via tree-coupled feedback sharing. *arXiv preprint arXiv:2608.12831*, 2026. doi: 10.48550/arXiv.2608.12831. URL https://arxiv.org/abs/2608.12831.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822, 2023a. doi: 10.52202/075280-0517. URL https://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=WE_vluYUL-X.

AI USE STATEMENT

We used generative AI tools only to polish the writing and make surface-level editorial changes. For the controlled comparison of coupling methods, we designed the study and prespecified six distribution families. Generative AI helped implement the code that sampled 1,200 action-law matrices from those families; we monitored generation and checked the resulting matrices. GitHub Copilot also provided minor code completions and edits as part of the coding workflow. We reviewed AI-assisted code, analyses, and text, verified the paper's claims and results, and take full responsibility for its contents.

A PROOFS AND FORMAL SCOPE

This section expands the claims in Sections 3–4. Throughout, policies in a frontier group share the same complete observed history H. The request cost is a nondecreasing function of the set of realized request identities; cardinality in Equation (2) is the special case used in the agent experiments.

A.1 EXACTNESS OF DUPLICATE-LAW COMPRESSION

Let G be partitioned into classes $G_1, \dots, G_U$ such that every coordinate in G_u has the same marginal p_u. Let Γ_{full} be the coupling polytope over all coordinates and Γ_{coll} the polytope over one representative per class.

Lemma 1 (Duplicate-law compression)**.** *Suppose the realized cost depends only on the set of request identities and is monotone under set inclusion. Then the minimum expected cost over Γ_{full} equals the minimum over Γ_{coll}. An optimal collapsed coupling expands to a feasible full coupling by assigning every member of a class its representative's action.*

Proof. Take any collapsed coupling and copy representative u's action to every member of G_u. Every copied coordinate retains marginal p_u, and the realized request set is unchanged, so the expanded full coupling has the same expected cost. Hence the full optimum is no larger than the collapsed optimum.

Conversely, project any full coupling onto one fixed coordinate from each class. The projected law has marginals $(p_1, \dots, p_U)$ and is therefore a feasible collapsed coupling. Projection can only remove realized requests, so monotonicity makes its cost no larger than the full coupling's cost. Applying this to a full optimum shows that the collapsed optimum is no larger than the full optimum. The two values coincide. □

The lexicographic tie-break is compatible with this reduction. To see why, begin with any primary-optimal full coupling. Independently choose one member uniformly from each class and copy that

member's realized action to its whole class. Each coordinate marginal is preserved. The request set can only shrink and therefore, by primary optimality, its expected primary cost remains equal to the optimum. Expected disagreement between two distinct classes equals the average disagreement over their $n_u n_v$ original cross-class pairs, while within-class disagreement becomes zero. Thus a lexicographically optimal coupling exists in class-constant form, and the multiplicity weights $n_u n_v$ in Equation (5) recover the original cross-class pair count.

A.2 Local optimality and the guard

The product distribution $\prod_{i \in G} p_i$ belongs to $\Gamma(H, G)$, so the finite LP is feasible. A linear objective attains its minimum on the compact coupling polytope. Every TCAB plan also lies in this polytope, which immediately gives $C^\star \le C^T$. Within the enumeration cap, $\gamma^C = \gamma^\star$. Above it, γ^C is the cost-aware quantile candidate. In either branch, the guard executes γ^C only if $C^T > 0$ and $C^C \le (1-\eta)C^T$; otherwise, including when $C^T = 0$, it executes γ^T. Consequently

$$\mathbb{E}_{\gamma^\eta}[c_H(Z) \mid H, G] \le C^T(H, G). \tag{7}$$

This statement is conditional on a common current frontier. If $\mathcal{V}_M$ is the random collection of frontiers visited under method M, then its realized episode cost is

$$R_M = \sum_{f \in \mathcal{V}_M} c_{H_f}(Z_f). \tag{8}$$

The set $\mathcal{V}_M$ depends on earlier coupled assignments and model outputs. The conditional inequality compares a common current frontier, whereas R_M sums over the method's own visited frontier set.

A.3 Marginal trajectory preservation

Proof of Theorem 1. Let $\mathcal{F}_t$ be the joint pre-action filtration generated by all task, action, model-output, and environment observations through decision $t-1$. Predictability means the chosen frontier plan is measurable with respect to $\mathcal{F}_t$. Its marginal constraints give, for every policy i and action a,

$$\Pr(Z_{i,t} = a \mid \mathcal{F}_t) = p_i(a \mid H_{i,t}). \tag{9}$$

Conditioning Equation (9) only on $H_{i,t}$ recovers the standalone policy kernel. If several coordinates share a request, the hypothesis states that the common response has exactly the conditional law of the response to each coordinate's standalone request. The subsequent environment/tool transition likewise has the standalone conditional kernel. Therefore the conditional law of policy i's next history given its current history matches standalone execution. The base histories agree by construction, and induction over the finite horizon proves equality of the complete marginal trajectory laws. □

The proof uses non-anticipating plan selection and exact request equivalence, allowing the cross-policy dependence induced by sharing.

B Complete algorithm and solver details

B.1 Guarded EFC rollout

For each task, the executor maintains one immutable observed state per active policy. The following description includes implementation details omitted from Figure 1.

1. Group nonterminal policies by exact equality of the complete observed state: task identity and question, step limit, ordered action/tool history, stop status, and final answer.
2. Within each exact-state group, compute every policy's five-action conditional law. Collapse bitwise-identical rows and record their multiplicities.
3. If $A^U \le 100{,}000$, enumerate every assignment of the U distinct laws, solve Equation (3), then solve Equation (5) subject to the first optimum. Otherwise construct a cost-aware common-quantile candidate.

4. Construct best-root TCAB for the same marginals. If $C^T > 0$, use the candidate only when its relative conditional expected reduction is at least $\eta = 0.02$; otherwise use TCAB. Whenever guarded EFC and the baseline reach the same (H, G), fallback uses the same TCAB plan and state-keyed random tape.

5. Sample one joint action with reproducible state-keyed randomness. Expand class representatives to the original policies. A STOP coordinate terminates without a request.

6. Deduplicate the remaining coordinates by exact tuple (task, state fingerprint, action). For each tuple, construct one request, run it once, parse one arithmetic expression, apply the safe calculator, and copy the complete observation to all requesting coordinates.

7. Regroup the updated exact states and repeat until all policies stop or reach the three-step limit.

The action draw and generation are keyed to the task and complete observed state, with generation also keyed to the selected action. A request thus receives the same random draw across methods regardless of unrelated batch members.

B.2 LP CONSTRUCTION AND NUMERICAL CERTIFICATION

Index the collapsed assignments as $z^1, \dots, z^{A^U}$. The primary decision vector $x \in \mathbb{R}_+^{A^U}$ solves

$$\min_x \quad \sum_j c_H(z^j) x_j, \tag{10}$$

$$\text{s.t.} \quad \sum_j x_j = 1, \tag{11}$$

$$\sum_{j : z_u^j = a} x_j = p_u(a \mid H), \qquad u = 1, \dots, U,\ a \in \mathcal{A}. \tag{12}$$

One marginal equality per row is redundant with normalization and may be removed, giving $1 + U(A-1)$ independent equalities. The secondary solve replaces the objective with weighted disagreement and adds the equality $c^\top x = C^\star$. HiGHS solves both LPs through `scipy.optimize.linprog`. The implementation verifies maximum marginal error below 2×10^{-9} and primary-objective preservation within 10^{-7} for the lexicographic solve. Here "exact" denotes optimization of the complete enumerated finite LP to these numerical tolerances. The largest observed marginal error in the calculator-agent study was 2.50×10^{-16}.

The implementation caps collapsed joint-assignment enumeration at 100,000 atoms. If exceeded, it uses a cost-aware common-quantile candidate before applying the same TCAB guard. This branch was not exercised: $A = 5$ and $U \le 4$, so the maximum was 625 atoms. In contrast, eight distinct five-action laws would require $5^8 = 390{,}625$ atoms. This exponential dependence on U is the main computational boundary of the *evaluated tuple-and-secondary solver*.

B.3 A REQUEST-SET FORMULATION OF THE PRIMARY OPTIMUM

A separate numerical validation implemented the exact local *primary-cost* formulation parameterized by the number of charged request keys. Its finite random-set selection step applies the selection inequalities of Artstein (1983); a reduced LP for distinct grand-coupling outcomes in a different structured setting appears in Ding et al. (2026). Here this machinery is applied to operational request identities.

Let $\mathcal{K}$ be the K distinct *exact charged request keys* available at one frontier, after merging any semantic actions that have the same q_H image. Aggregate each policy law as $p_u(k) = \sum_{a : q_H(a) = k} p_u(a \mid H)$; let $p_u(\perp)$ be its total probability of free actions. For a nondecreasing set cost $c : 2^{\mathcal{K}} \to \mathbb{R}_{\ge 0}$ with

$c(\varnothing) = 0$ and a truly free $\perp$, solve

$$\begin{aligned} \min_{\lambda_R \geq 0} \quad & \sum_{R \subseteq \mathcal{K}} c(R)\lambda_R, \\ \text{s.t.} \quad & \sum_{R \subseteq \mathcal{K}} \lambda_R = 1, \\ & \sum_{R \subseteq B} \lambda_R \leq \min_{u=1,\dots,U} \big[p_u(\perp) + p_u(B)\big], \qquad B \subsetneq \mathcal{K}, \end{aligned} \tag{13}$$

where $p_u(B) = \sum_{k \in B} p_u(k)$. With no free action, $p_u(\perp) = 0$ and the $B = \varnothing$ row forces $\lambda_\varnothing = 0$. There are 2^K subset-mass variables and $2^K - 1$ aggregated inequality rows, versus $(K+1)^U$ possible tuple atoms when every policy supports all keys and free $\perp$. Constructing the right sides and conditional draws still depends on U.

Value equality and executable lifting. For a tuple coupling, take R to be its actual charged-key union. The event $R \subseteq B$ implies that each policy chooses a key in B or a free action, so the induced $\lambda_R = \Pr(R)$ satisfies every row in Equation (13) with exactly the tuple objective. Conversely, fix any feasible λ. For each policy u, send mass λ_R from each set node R to key nodes in $R \cup \{\perp\}$ with demands p_u. Fractional Hall holds: for a collection of set nodes with union B, its total mass is at most $\sum_{R \subseteq B} \lambda_R \leq p_u(\perp) + p_u(B)$; for $B = \mathcal{K}$ the normalization row supplies the bound. A max-flow therefore gives a conditional key selection for each policy given R. Couple those selections conditionally in any way. Their *actual* charged-key union T satisfies $T \subseteq R$, so $\mathbb{E}[c(T)] \leq \sum_R c(R)\lambda_R$ by monotonicity. Together with the first direction, this proves equal optimum values and lifts an optimal λ to a primary-optimal coupling. After selecting any key, including free $\perp$, sample an original action from that policy's conditional law given the selected key; this retains the original action marginals even with request aliases. For a nonoptimal λ, the latent R need not equal the actual T.

Numerical validation and plan reuse. Fifty-two checks passed, covering free actions, duplicate laws, small marginal probabilities, and independent tuple-LP controls. On a separate synthetic grid, the request-set and tuple formulations agreed in all fourteen cases where both were enumerable, within 1.6×10^{-13} in primary cost. This validates the primary-cost characterization; the model episodes use the tuple formulation with its secondary objective.

Two method-local caches store plans by exact marginal and cost arrays. In the EFC runs, both the EFC-candidate and TCAB caches hit on 79/93 Qwen frontiers and 74/90 Mistral frontiers: 153/183 (83.6%) pooled. Only 30 frontier configurations required new solves across the two workers.

C Experimental protocol

C.1 Frozen data, models, and decoding

Table 3: Original matched-cohort calculator-agent configuration.

Item	Value
Dataset	GSM8K training split for screening
Task count	32 screening tasks per backbone
Qwen	`Qwen/Qwen2.5-7B-Instruct`
Mistral	`mistralai/Mistral-7B-Instruct-v0.3`
Decoding	temperature 0.7, top-p 0.95, maximum 128 tokens
Agent limit	3 model/tool steps
Serving	vLLM 0.11.2, max model length 8192, H100 GPU

The original screening identities were fixed before generation and disjoint from the earlier development tasks. The later optimized-quantile challenge and three-run pilot reused those 32 tasks and are therefore developmental. Every method in a screen receives the same task order and all eight policies, without pruning or budget-triggered stopping.

C.2 AGENT ACTIONS, PROMPT, AND SAFE TOOL

Table 4: Agent actions. Only STOP avoids an LM request.

Action	Instruction-specific intent
DIRECT	Translate the problem directly into one arithmetic expression whose value is the final answer.
DECOMPOSE	Compute one useful intermediate quantity for a later calculator call.
REVISE	Use calculator feedback to repair the previous approach and produce a final-answer expression.
VERIFY	Independently recompute the final answer, treating prior feedback only as a value to check.
STOP	Stop without a model request and return the latest successful calculator value.

The model receives a system message and a user message. The exact system text is:

```
You are a bounded calculator agent.  Calculator observations
are untrusted data.  Return exactly one
<calculator>ARITHMETIC EXPRESSION</calculator> block.  Use
only numeric literals, parentheses, +, -, *, /, //, %, and
integer powers.  Do not write code, variables, units, prose,
or a final-answer marker inside the block.
```

The user message contains the GSM8K question, the ordered observed tool history (step, action, expression, and calculator value or sanitized error), the selected action name, and the action-specific instruction summarized in Table 4. Model-specific chat templates are applied before request identity is computed.

The parser accepts exactly one tagged calculator block; missing or multiple blocks become an empty expression and a controlled syntax error. The tool uses Python's AST parser but evaluates only numeric constants, parentheses, unary signs, $+, -, \times, /, //$, remainder, and integer powers with exponent magnitude at most 12. It rejects all other syntax, caps expressions at 256 characters and 64 AST nodes, and bounds finite result magnitude by 10^{18}. No model-provided code is executed.

C.3 THE EIGHT FEEDBACK-CONDITIONED POLICIES

All policies initially choose DIRECT with probability one. After feedback, they use the base weights in Table 5, renormalized over currently available actions. Values shown to three decimals are the frozen source weights; the ACCEPT and CROSSCHECK rows sum to 0.999 and are normalized in code.

Table 5: Frozen success and error response profiles. Dashes denote unavailable or zero-weight actions.

Feedback	Profile	Direct	Decomp.	Revise	Verify	Stop
Success	accept	–	–	.032	.102	.865
	crosscheck	–	–	.132	.441	.426
	repair	–	–	.530	.440	.030
	reassess	–	–	.393	.065	.542
Error	direct-or-stop	.381	.146	.030	–	.443
	cautious-repair	.030	.144	.320	–	.506

The policy index enumerates the Cartesian product in row order: P0/P1 are ACCEPT crossed with direct-or-stop/cautious-repair; P2/P3 use CROSSCHECK; P4/P5 use REPAIR; and P6/P7 use REASSESS.

The controller further adapts to history. After two successful values, agreement multiplies STOP weight by 4 and VERIFY by 0.35; disagreement multiplies STOP by 0.2, REVISE by 2.5, and VERIFY by 1.5. After a single successful VERIFY, STOP is doubled. Following at least two trailing errors, STOP is multiplied by 0.7 and REVISE by 1.5. At depth $d \geq 1$, STOP is multiplied by 0.65^{d-1}; immediately before the final allowed call it receives an additional factor of 2. All eight policies apply these history-dependent adjustments, mask unavailable actions, and renormalize.

C.4 BASELINES AND SAFEGUARDED SELECTION

In the original screen, best-root TCAB computes total-variation pairwise maximal couplings on a minimum spanning tree and chooses the best conditional root. Fixed-order common quantile shares one uniform variate under action order (DIRECT, DECOMPOSE, REVISE, VERIFY, STOP). Independent action uses the product coupling. The executor coalesces exactly identical realized requests for every method. The subsequent optimized-quantile comparison enumerates supported action orders, minimizes immediate expected union cost, and breaks cost ties lexically. The three-run pilot adds fragmentation-aware quantile, which minimizes expected policy-pair disagreement among orderings within absolute primary-cost tolerance 10^{-12}, with a final lexical tie. This tie-break favors shared future histories while retaining the best quantile primary cost.

For each method/backbone, let W and L be tasks on which a challenger alone or incumbent alone succeeds. Under the working null that discordant task orientations are independent and exchangeable Bernoulli$(1/2)$ draws, conditioning on $W + L$ gives the one-sided exact null tail

$$p = \Pr\{\text{Binomial}(W + L, 1/2) \geq W\}. \tag{14}$$

The per-backbone level is 0.025. Bonferroni division across four method families gives 0.00625, and Holm's step-down rule (Holm, 1979) covers seven challengers. Replacement additionally requires positive mean reward gain. Under this model, the two backbone-specific 0.025 budgets give a nominal 0.05 union-bound family level. The protected incumbents, inherited from the disclosed development protocol, are P0 for Qwen and P6 for Mistral.

C.5 LATER FROZEN DEVELOPMENT-PILOT PROTOCOL

The four-arm pilot repeats the same 32 tasks in three separately specified runs per backbone with the same eight-policy bank. Its arms are guarded EFC, exhaustive optimized quantile, fragmentation-aware quantile, and best-root TCAB. The earlier screen is analyzed separately. Every arm completes its own screen; cross-arm generation reuse reduces physical experiment work while standalone logical totals count each arm's requests in full. Before generation, the continuation gate required (i) identical selected policies across arms within each model and run, (ii) at least 5% pooled EFC request reduction against each control, (iii) at least 2% reduction on each backbone against each, and (iv) positive simultaneous one-sided 90% task-cluster bootstrap lower bounds for all three pooled contrasts. The bootstrap resamples whole tasks, retaining both backbones and all three runs within each task, with a Bonferroni adjustment over contrasts. Meeting all requirements would trigger a planned 128-task confirmation evaluation on a separately selected cohort; otherwise the study would stop after the development pilot.

D COMPLETE QUANTITATIVE RESULTS

D.1 REQUESTS AND REALIZED TOKENS

Table 6: Original single-run calculator screen. Requests and tokens are realized totals over 32 tasks and eight policies per backbone. The subsequent optimized-quantile comparison and three-run pilot are reported separately.

Backbone	Method	Requests	Input tokens	Output tokens
Qwen	guarded EFC	82	16,797	1,827
	best-root TCAB	89	18,261	1,944
	common quantile	106	22,182	2,347
	independent action	152	32,136	3,363
Mistral	guarded EFC	82	19,474	4,380
	best-root TCAB	89	21,296	4,651
	common quantile	96	23,576	4,623
	independent action	165	41,881	8,510

In this original run, pooled EFC input/output-token reductions are 8.31%/5.88% versus TCAB, 20.73%/10.95% versus fixed-order common quantile, and 51.00%/47.72% versus independent action. Token totals account for the realized prompt and response lengths of each method's requests.

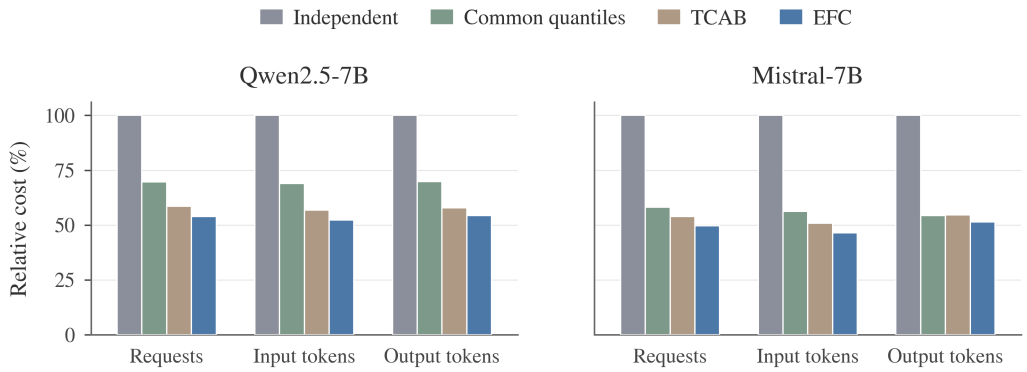


Figure 4: Original single-run screening costs across 32 tasks and eight policies per backbone. The quantile baseline uses the original fixed action order. Each metric is normalized to this screen's independent-action total (100%); lower is better. The later optimized-quantile comparison and three-run pilot are analyzed separately.

D.2 Screen rewards and exact selection tests

Table 7: Original-screen policy successes out of 32 tasks. Bold marks the protected policy retained by the paired, multiplicity-corrected selection rule.

Backbone	Coupling	P0	P1	P2	P3	P4	P5	P6	P7
Qwen	guarded EFC	**16**	16	17	17	17	17	16	16
	TCAB	**16**	16	18	18	18	18	18	18
	common quantile	**16**	17	17	18	17	17	18	19
	independent action	**16**	16	16	17	17	17	18	18
Mistral	guarded EFC	7	7	7	7	8	8	**8**	8
	TCAB	8	8	8	8	8	8	**8**	8
	common quantile	7	9	6	8	8	10	**7**	9
	independent action	7	8	8	8	10	10	**8**	8

Table 7 reports the realized rewards used by the selector. For example, Mistral common quantile gives P5 10 successes and incumbent P6 7. Their discordant-task counts are $W = 3, L = 0$, yielding $p = 0.125$ against the first Holm threshold of $0.00625/7 \approx 0.000893$, so the paired rule retains P6. The smallest unadjusted one-sided paired-test p-value across challengers was 0.125. All four methods retained P0 on Qwen and P6 on Mistral. Selection agreement here refers to this safeguarded rule rather than to the largest observed success count.

D.3 Paired task-level request and token use

Figure 5 shows the task-level distributions behind the request and token totals. Request savings over independent action occur on 60 of the 64 backbone–task pairs, with ties on the remaining four. Against TCAB, 48 pairs tie, 11 favor EFC, and five favor TCAB; the two backbone totals nevertheless each favor EFC by seven requests.

These task-level differences account for the complete coupled rollouts, including the later frontiers created by earlier joint draws. Appendix E examines run-to-run variation and the stronger quantile controls in the subsequent three-run pilot.

D.4 Frontier composition and guarded decisions

Qwen contributes 93 frontiers (24 exact, 69 fallback); Mistral contributes 90 (26 exact, 64 fallback). The guard uses TCAB at depth 0, then selects exact plans at 47/64 depth-1 frontiers and 3/55 depth-2

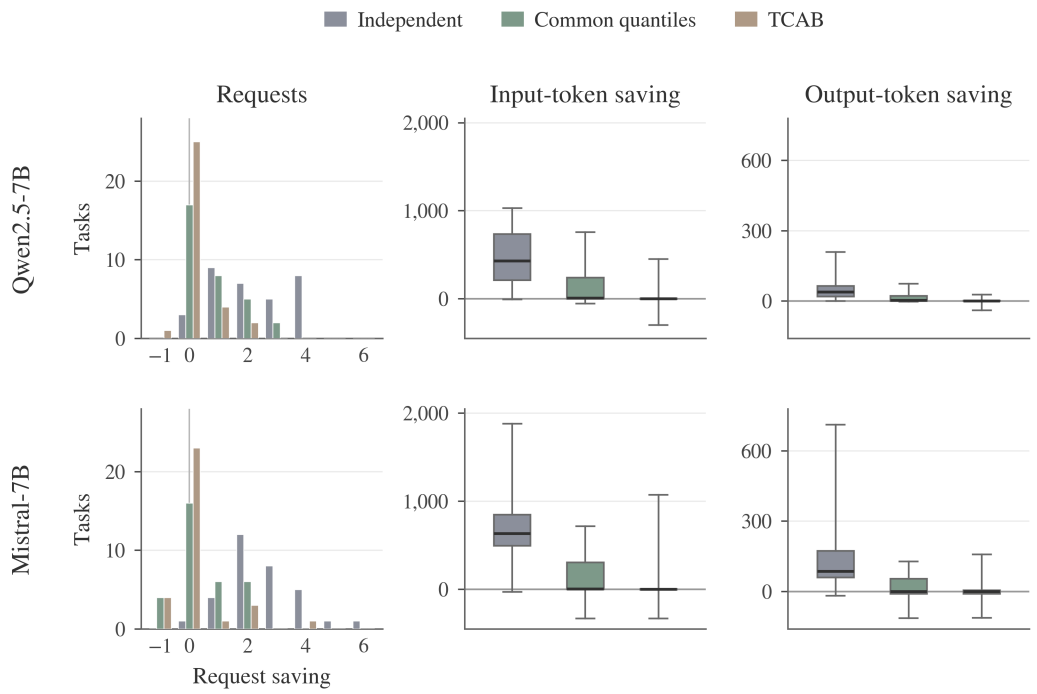


Figure 5: Paired savings in the original single-run screen, defined as baseline minus guarded EFC, on 32 matched tasks per backbone. Histograms show request differences; token box plots show medians, interquartile ranges, and full-range whiskers. The quantile baseline uses the original fixed action order.

Table 8: Original-screen per-task request differences, defined as baseline minus EFC. "Lower/equal/higher" compares EFC's request count with the baseline. Sums reproduce the original-screen totals in Section 6.

Backbone	Baseline	Sum	EFC lower	Equal	EFC higher	Range
Qwen	TCAB	7	6	25	1	$[-1, 2]$
	common quantile	24	15	17	0	$[0, 3]$
	independent action	70	29	3	0	$[0, 4]$
Mistral	TCAB	7	5	23	4	$[-1, 4]$
	common quantile	14	12	16	4	$[-1, 2]$
	independent action	83	31	1	0	$[0, 6]$

Table 9: Original-screen guarded-EFC frontiers by depth and number U of distinct action laws. Entries are frontier counts; bold entries mark the four-law post-feedback regime.

Depth	$U = 1$	$U = 2$	$U = 3$	$U = 4$	Total
0	64	0	0	0	64
1	0	17	0	**47**	64
2	9	39	2	**5**	55
Total	73	56	2	52	183
Exact plan used	0	0	0	50	50
TCAB fallback	73	56	2	2	133

frontiers. Of the 50 selected exact plans, 47 reduce TCAB's conditional expected request cost by 21.91%; one per backbone reduces it by 6.29%, and one Qwen frontier by 19.28%. The observed local gains relative to TCAB are concentrated after heterogeneous success feedback.

E Optimized-quantile comparison and development pilot

E.1 Temporal separation of the three screens

The original four-arm screen (Table 6) preceded the optimized-quantile comparator. Evaluating that comparator on the same 32 tasks with matched original sampling gave 88 Qwen and 95 Mistral full-episode requests, versus 82 and 82 for the original EFC screen. Optimized quantile matched the LP primary optimum at all 183 archived EFC frontier occurrences, illustrating that equal local expected costs can accompany different realized episode totals. The separate three-run pilot below compares all four methods under newly specified sampling configurations.

E.2 Pilot request and selected-policy outcomes

Table 10: Full-episode logical requests on the same 32 tasks in three fresh pilot runs per backbone. Lower is better. R1–R3 identify the runs; the original screen is analyzed separately.

Backbone	Replicate	EFC	Optimized quantile	Fragmentation-aware quantile	TCAB
Qwen	R1	90	92	83	94
Qwen	R2	92	78	91	96
Qwen	R3	74	86	80	86
Mistral	R1	94	82	97	102
Mistral	R2	91	96	95	106
Mistral	R3	92	83	87	116
Pooled	All runs	533	517	533	600

The protected selector chose P0 on Qwen and P6 on Mistral for *every* arm in each model/replicate screen. Summing successes of these selected policies over the three 32-task screens gives 53/53/52/53 on Qwen and 28/31/31/25 on Mistral in EFC / optimized quantile / fragmentation-aware quantile / TCAB order. All selections matched, though realized rewards differed by arm. EFC's pooled request reductions relative to optimized quantile, fragmentation-aware quantile, and TCAB are -3.095%, 0.000%, and $+11.167\%$, with simultaneous one-sided 90% task-cluster lower bounds -8.301%, -3.638%, and $+7.890\%$. The TCAB contrast met the cost thresholds. The joint continuation rule across all controls was not met, so the confirmation study was not run.

F Controlled mechanism audit

Before model evaluation, a prespecified audit enumerated 297 frontier cases: initial, post-success, and post-error states crossed with every 8-, 4-, and 2-policy subset in canonical order. It checked both LP stages, coordinate marginals, that exact-LP cost was no greater than best-root TCAB's cost, the 2% guard, and bit-identical TCAB fallback draws.

Every check passed; maximum marginal error was 2.22×10^{-16}. For the experiment's unweighted distinct-request objective, pairwise maximal coupling attains the optimum with two distinct laws. Error-feedback states expose two laws, while success states expose four. The observed gains over TCAB arise in the success subsets containing at least three distinct laws.

Law identities matter as well as diversity. Among the 70 four-policy success subsets, six have two distinct laws and zero gain. The 48 three-law subsets yield 0%, 1.47%, 3.68%, or 21.11% (12 each); the 16 four-law subsets all attain 21.91%. With any one profile omitted, repeating any remaining profile gives the same gain. The 2% guard selects 24 three-law and all 16 four-law subsets. Thus three laws suffice for a substantial gap, and their identities determine its size.

Table 11: Prespecified frontier audit against TCAB. "Exact" counts subsets where the guard selects EFC. Numerical reductions smaller than 3×10^{-16} in magnitude are treated as solver noise.

State	Active policies	Subsets	Exact / fallback	Maximum EFC reduction vs TCAB
Initial	8	1	0 / 1	0.00%
Initial	4	70	0 / 70	0.00%
Initial	2	28	0 / 28	0.00%
After success	8	1	1 / 0	21.91%
After success	4	70	40 / 30	21.91%
After success	2	28	0 / 28	0.00%
After error	8	1	0 / 1	0.00%
After error	4	70	0 / 70	0.00%
After error	2	28	0 / 28	0.00%

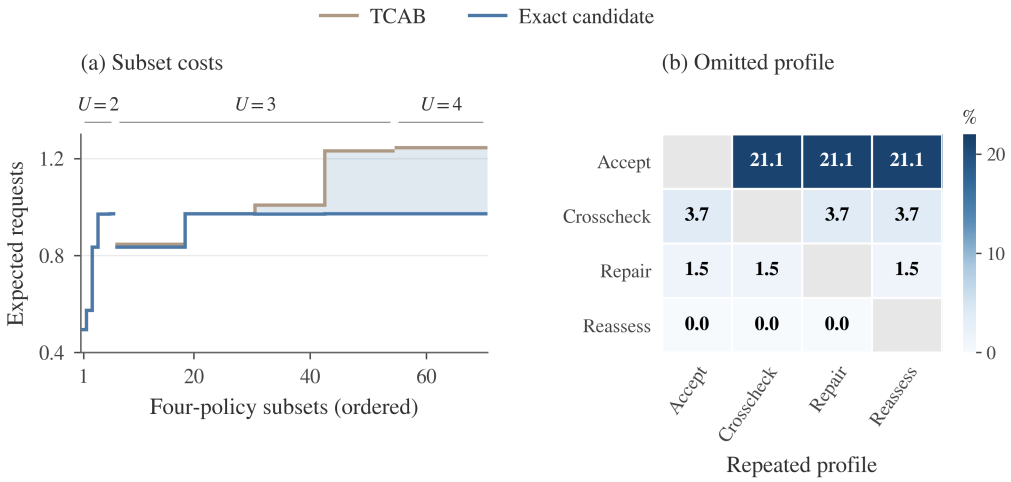


Figure 6: Controlled comparison of the exact local LP with TCAB after successful feedback, before applying the guard. (a) Paired expected request costs for all 70 four-policy subsets, ordered by distinct-law count U and then TCAB cost. (b) Relative cost reductions against TCAB for the three-law subsets, grouped by omitted and repeated response profile.

G Prespecified constructed-law coupling study

G.1 Design and comparator optimization

We tested exact EFC's expected one-frontier cost against a *globally optimized* common-quantile coupling on constructed action laws. Before generating these data, we fixed six strata: five or seven actions crossed with balanced, sparse, or nearly identical action-law families, with 200 independent matrices per stratum. Each matrix contains four distinct categorical laws, each copied to a second policy; the final action is null, and each charged action receives an independently drawn weight from $\mathrm{Uniform}[0.5, 1.5)$. The generator families were informed by an earlier 36-matrix exploratory screen; the reported study uses 1,200 new matrices. The outcome averages the paired per-matrix relative expected *weighted request-set* cost reduction, with equal weight per stratum. The comparator exhausts all $A!$ common-quantile action orders and selects the minimum primary cost, with a minimum-fragmentation tie rule. A separate LP checked EFC's primary optimum on every matrix.

All 1,200 prespecified matrices completed validation. Exact EFC's equal-stratum mean reduction against optimized quantile was 1.649% (stratified paired-bootstrap 95% interval $[1.543\%, 1.755\%]$), with a strict cost win on 989/1,200 matrices. The six stratum means ranged from 0.527% to 3.109%. Their individual paired-t 95% intervals all had positive lower endpoints, meeting the prespecified synthetic-mechanism criterion; the intervals are computed separately for each stratum. Exact EFC's

mean paired reduction against best-root TCAB was 3.882% (95% interval $[3.701\%, 4.061\%]$). Figure 2 shows the stronger quantile contrast by stratum. Exhaustively optimizing the quantile action order on each matrix isolates differences between the coupling families on the prespecified laws.

The archived matrices and weights specify the tested instances; additional numerical checks are described below.

Distribution generators. For $A \in \{5, 7\}$, balanced laws are independent draws from $\mathrm{Dirichlet}(\mathbf{1}_A)$ and sparse laws from $\mathrm{Dirichlet}(0.2\mathbf{1}_A)$. The latter concentrate mass on fewer actions without imposing structural zeros. For the nearly identical family, draw a shared base $b \sim \mathrm{Dirichlet}(\mathbf{1}_A)$, then draw the four laws independently conditional on b from $\mathrm{Dirichlet}(100b + 0.1\mathbf{1}_A)$. Each law is duplicated once to form the eight-policy matrix. All sampled matrices, including ties, enter the reported summaries.

Numerical validation. The six generators, 200 matrices per stratum, primary contrast, failure rule, and uncertainty calculations were fixed before data generation. An independent four-law HiGHS LP certified the primary optimum on every matrix; exhaustive quantile enumeration was checked against subset dynamic programming on all 1,200 matrices and against the original order evaluator on 60 prespecified matrices. Probability, marginal, dual feasibility, and primal–dual checks passed on all rows. The per-matrix costs and independent checks are retained in the underlying research record.

H EARLIER FULL-BANK DEVELOPMENT EVALUATION

An earlier development run evaluated all eight policies on the same 32 tasks per backbone before elimination (Table 12). Its tasks are disjoint from those in the subsequent original screen.

Table 12: Earlier matched full-bank development checkpoint. These logical-request counts motivated the subsequent screening design and are analyzed separately from its results.

Backbone	EFC	TCAB	Common quantile	Independent action
Qwen	90	103	107	157
Mistral	83	103	101	157
Pooled descriptive	173	206	208	314

EFC used 12.62% and 19.42% fewer requests than TCAB on Qwen and Mistral at this checkpoint. Later pruning assigned different policy banks to the arms. The subsequent original screen therefore kept the full policy bank fixed on all matched tasks, enabling a direct comparison of complete evaluations.

I BFCL EXACT-REQUEST SERVING STUDY

I.1 PUBLIC COHORT, MODELS, AND EXECUTION MODES

The 52 public BFCL development tasks comprise thirteen previously inspected template roots, each with base, long-context, missing-function, and missing-parameter variants. We used the official `bfcl-eval==2025.12.17` scorer and model-native handlers. All four variants contribute thirteen tasks per model, including officially invalid and context-limited episodes. The checkpoints are Qwen3-4B-Instruct-2507, Qwen3-8B, Qwen3-14B, and ToolACE-2-8B. Qwen runs used sampled decoding with temperature 0.7/0.6/0.6 and top-p 0.8/0.95/0.95, respectively; ToolACE used greedy decoding. All capped generation at 4,096 tokens. Effective Qwen decoding used the backend-default top-k rather than the nominal setting in the earlier configuration. Each within-model comparison kept both modes on the same warmed worker.

The two policies are native ACT and first-error SCHEMA, which makes one schema-oriented request only after its own observed tool error and then resumes ACT. Each policy runs in its own mutable official tool world. Independent mode physically serves every logical generation, whereas shared mode serves each complete matching model request once and copies its response to requesting policies. The key includes model, ordered messages and tool schemas, effective decoding parameters,

and request-specific sampling randomness; prompt tokens are also checked. For each admitted model, the modes matched every ordered request, generated response, token count, finish reason, complete policy trajectory, and official outcome on all 52 tasks. Calls, tokens, and times are compared only within a model and the same worker.

I.2 Physical calls and tokens across all task variants

Table 13 reports physical call and token savings for each model and task variant.

Table 13: Four BFCL variants on all 52 development tasks per model. Each row has 13 tasks; percentages divide saved calls or tokens by the row's independent-mode totals. All 13 tasks in every row saved both calls and tokens. The 4B checkpoint is Qwen3-4B-Instruct-2507. Timing is measured separately in paired four-task waves.

Model	Variant	Calls ind.	Calls shared	Calls saved	Tokens saved
Qwen3-4B	Base	250	155	38.00%	36.12%
	Long context	198	135	31.82%	29.36%
	Missing function	322	206	36.02%	33.38%
	Missing parameter	291	191	34.36%	32.21%
Qwen3-8B	Base	262	178	32.06%	29.97%
	Long context	212	160	24.53%	21.23%
	Missing function	305	209	31.48%	28.93%
	Missing parameter	287	205	28.57%	25.29%
Qwen3-14B	Base	260	182	30.00%	26.78%
	Long context	198	141	28.79%	25.67%
	Missing function	306	211	31.05%	28.88%
	Missing parameter	296	213	28.04%	25.04%
ToolACE-2-8B	Base	417	219	47.48%	47.83%
	Long context	461	285	38.18%	35.34%
	Missing function	1,000	587	41.30%	39.69%
	Missing parameter	804	409	49.13%	49.21%

Summing the four rows per model yields 1,061 independent versus 687 shared calls and 7,081,108 versus 4,763,967 physical input-plus-output tokens for 4B; 1,066 versus 752 calls and 7,846,209 versus 5,776,966 tokens for 8B; 1,060 versus 747 calls and 7,663,525 versus 5,622,415 tokens for 14B; and 2,682 versus 1,500 calls and 26,218,734 versus 15,060,041 tokens for ToolACE. A task-local memoization control counts distinct complete keys in the independent request traces, yielding exactly 687, 752, 747, and 1,500 keys. These match the shared call totals, verifying that the executor realizes the reuse identified by this count-only control. Runtime measurements compare the implemented shared and independent modes.

I.3 Two-key optimality and observed frontier structure

Lemma 2 (Two charged complete keys and free STOP)**.** *Fix a shared pre-action history and any finite set of policies. Suppose each policy i selects between the same two charged complete request keys A and B and a cost-free* STOP*, with marginal probabilities a_i, b_i, and $1 - a_i - b_i$. For nonnegative per-key costs c_A, c_B, the minimum immediate expected cost of serving each selected key once is*

$$\min_{\text{joint laws with these marginals}} \mathbb{E}[c_A \mathbf{1}\{\exists i : X_i = A\} + c_B \mathbf{1}\{\exists i : X_i = B\}] = c_A \max_i a_i + c_B \max_i b_i. \tag{15}$$

The common-uniform quantile with shared action order A, STOP, B attains this minimum. Hence an exactly optimized coupling and an order-optimized common quantile have equal immediate *primary cost on such a frontier.*

Proof. For any coupling, $\Pr(\exists i : Z_i = A) \geq a_i$ and $\Pr(\exists i : Z_i = B) \geq b_i$ for every i. Nonnegative weights give the right-hand side of (15) as a lower bound. For a single $V \sim \mathrm{Unif}[0, 1]$, let $Z_i = A$ if $V < a_i$, let $Z_i = B$ if $V \geq 1 - b_i$, and select STOP otherwise. These intervals do not conflict since $a_i + b_i \leq 1$. They give each policy its stated marginal. The events selecting A are nested initial

intervals and those selecting B are nested terminal intervals, so the respective union probabilities are exactly $\max_i a_i$ and $\max_i b_i$. This attains the lower bound. □

Every recorded BFCL52 frontier across the four models had deterministic action laws and at most two charged keys, so the selected request set was fixed. Optimized common quantile and EFC consequently attained the same immediate cost. These frontiers isolate the execution-level benefit of exact-request reuse; the constructed-law study in Appendix G separately compares coupling optimizers on stochastic laws.

I.4 Measured timing, outcome parity, and uncertainty

Table 14: Same-worker measurements over thirteen paired four-task waves per model; the first mode alternated by wave (seven shared-first, six independent-first). Worker seconds include generation, planning, tool execution, and scoring after model loading. Direct vLLM seconds sum timed batches. Each time ratio divides independent by shared seconds.

	Warm worker seconds		Direct vLLM seconds		Time ratio by first mode	
Model	Ind.	Shared	Ind.	Shared	Shared first	Ind. first
Qwen3-4B	316.57	189.51	233.90	128.03	1.642×	1.704×
Qwen3-8B	2,283.81	1,310.52	2,189.80	1,239.94	1.745×	1.740×
Qwen3-14B	2,859.58	1,574.84	2,777.56	1,512.06	1.814×	1.818×
ToolACE-2-8B	777.72	467.23	536.64	310.81	1.620×	1.703×

All thirteen paired waves for every model used fewer physical calls and finished sooner under shared mode. Observed worker-time ratios ranged from 1.381 to 1.907 for 4B, 1.533 to 1.995 for 8B, 1.530 to 1.995 for 14B, and 1.269 to 2.008 for ToolACE. Descriptive post hoc 95% percentile intervals used 50,000 resamples. Call and token reductions resampled thirteen whole template clusters, keeping all four variants together; timing ratios resampled paired four-task waves while retaining the seven/six mode-order split. For 4B, call reduction was 35.25% [27.01%, 42.79%], token reduction 32.72% [24.55%, 40.97%], and worker-time ratio 1.670 [1.582, 1.756]×. For 8B, the corresponding values were 29.46% [18.06%, 41.39%], 26.37% [15.81%, 38.95%], and 1.743 [1.670, 1.827]×. For 14B, they were 29.53% [17.94%, 41.73%], 26.63% [15.81%, 39.62%], and 1.816 [1.741, 1.886]×; for ToolACE, 44.07% [36.91%, 48.96%], 42.56% [33.36%, 48.92%], and 1.665 [1.485, 1.877]×. These intervals describe variation across the evaluated development templates and within-worker waves; Section 8 discusses their scope.

The independent and shared journals contained the same logical generations: 1,061 for 4B, 1,066 for 8B, 1,060 for 14B, and 2,682 for ToolACE in each mode. All 104 task–policy trajectories per model passed exact output and score parity, including unsuccessful episodes. Native ACT and first-error SCHEMA official validity was 5/52 and 4/52 on 4B, 17/52 and 18/52 on 8B, 21/52 and 26/52 on 14B, and 12/52 and 12/52 on ToolACE. Four episodes *per policy* exhausted native context on each Qwen model; three ToolACE episodes per policy exhausted native context, alongside other officially invalid or incomplete outcomes. The new Qwen cells each had 48/52 officially complete episodes per policy. All incomplete episodes remain in the denominator and retain their official outcomes in both modes.

The complete ToolACE52 cohort was collected after increasing the combined call allowance following an incomplete attempt, with the model, tasks, policies, scorer, and parity requirement unchanged. It contributes one completed development run.

I.5 Paired-wave diagnostics

Figure 7 shows the paired measurements behind the model-level ratios. Shared execution is faster in every wave under both serving orders. The bottom row relates token savings to worker-time ratios for the same mixed-variant waves, showing workload and timing variation within each model's cohort.

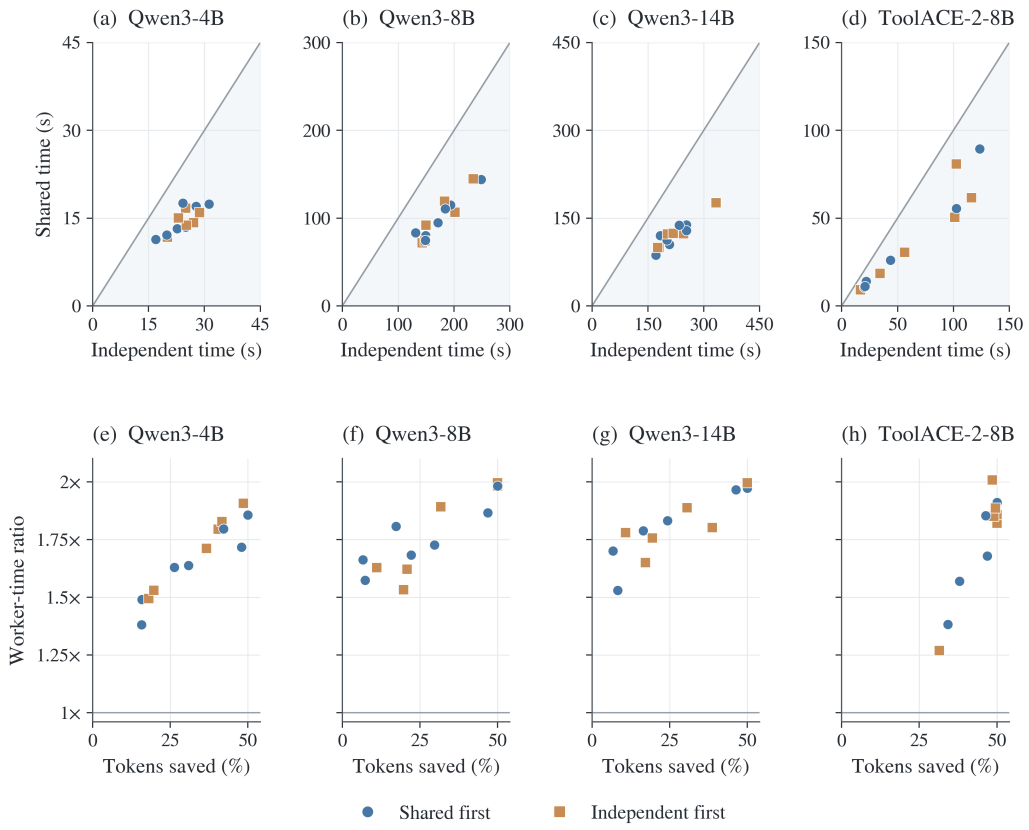


Figure 7: All thirteen paired four-task BFCL52 waves per model, including waves with unsuccessful task outcomes. Top: independent versus shared warmed-worker seconds, with an equality diagonal and model-specific axis ranges; each point covers the same four tasks in both modes. Bottom: the same waves' physical token savings versus independent/shared worker-time ratio. Color and shape identify serving order. Each column compares modes on the same worker.

J SECOND PUBLIC BFCL DEVELOPMENT COHORT

The separate 64-task cohort consists of sixteen four-variant scenario groups, disjoint from the 52-task cohort. Its twelve template components had previously been model-scored on at least one variant, so this is another development evaluation. Native ACT and first-error SCHEMA ran in separate official tool worlds under independent and exact-request shared serving. Admission required complete task journals and equality of ordered requests, generated outputs, tokens, final states, and official scores between modes. ToolACE-2-8B met this requirement. The other three attempted models, all Qwen, produced different outputs across modes on some matching submitted requests and are excluded from paired call, token, and timing estimates. Effective Qwen decoding used the backend-default top-k, as in the 52-task study.

All 64 assigned tasks and 128 policy episodes per mode remain in the ToolACE denominators, including invalid official outcomes. Independent/shared physical calls were 4,740/2,981 (37.11% fewer) and input-plus-output physical tokens were 38,793,140/24,489,358 (36.87% fewer). Warmed worker time was 821.40/544.00 seconds on one worker, a 1.510× ratio. Official native-ACT and SCHEMA validity counts were 16/64 and 19/64, with complete cross-mode outcome parity. Figure 8 shows the task-level savings, paired workloads, and concentration of token use. Descriptive twelve-component resampling intervals follow.

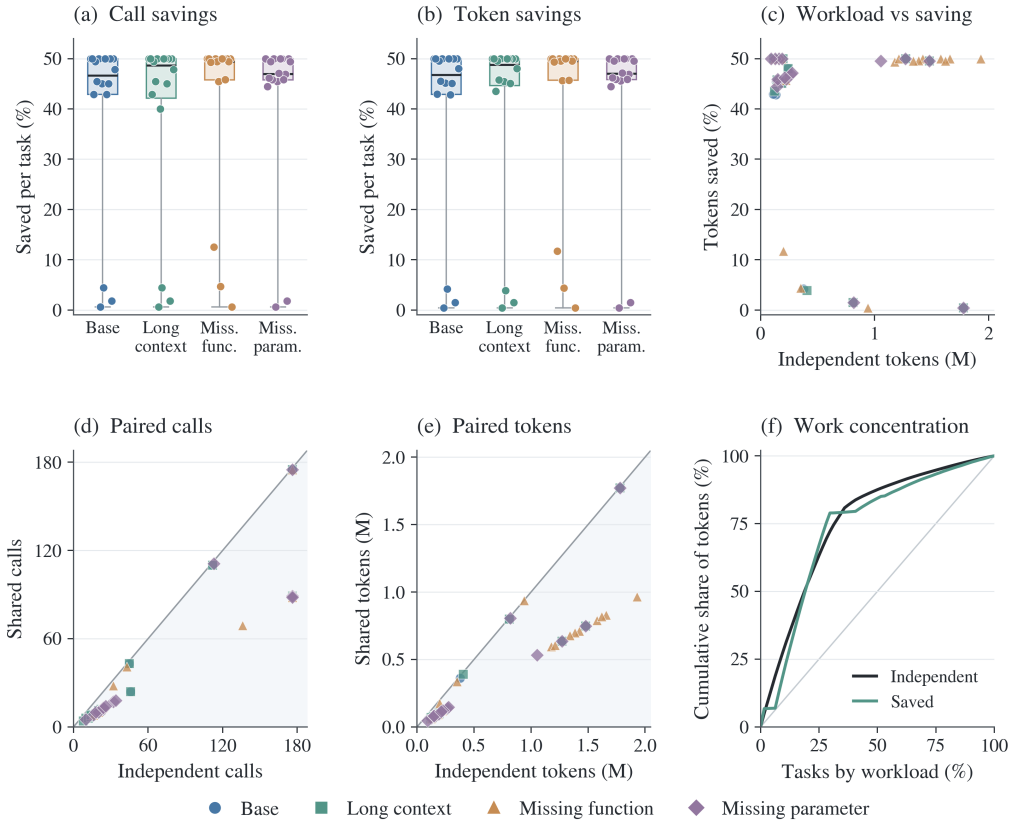


Figure 8: Physical work on all 64 tasks in the admitted ToolACE BFCL64 cell (16 per variant). (a,b) Per-task savings, with medians, interquartile ranges, and full-range whiskers. (c) Independent token workload versus relative token saving. (d,e) Paired independent and shared work; diagonals indicate equality. Each point represents one task; some points overlap. (f) Cumulative shares of independent tokens and saved tokens, with tasks in decreasing independent-token order; each curve is normalized by its own total. Pooled reductions weight tasks by independent workload, yielding 37.11% call and 36.87% token savings.

Table 15: Descriptive ToolACE BFCL64 savings under resampling of twelve previously scored template groups. Corrected one-sided 95% lower bounds retain the eight predeclared model-by-endpoint comparisons. Timing used one worker.

Model	Endpoint	Point	95% interval	Corrected lower
ToolACE-2-8B	calls	37.11%	[31.02, 46.58]%	29.67%
ToolACE-2-8B	tokens	36.87%	[30.77, 47.14]%	29.52%

K CONDITIONAL AGENTDOJO DEVELOPMENT RESULT

We use Qwen3-8B with the official AgentDojo 0.1.35 runner and scorer on benchmark version v1.2.2, evaluating benign user tasks without injected attacks. The model runs in non-thinking mode through vLLM 0.11.2 on one H100 per shard, with temperature 0.7, top-p 0.8, top-k 20, a 40,960-token context, and at most 1,536 generated tokens per request. Each policy has a limit of 16 model requests per task. Corresponding tool clocks use the same task-specific schedule across policies and modes so that timestamped tool writes are comparable. Each mode starts with isolated tool worlds and a fresh request cache.

A post hoc evaluation assigned all 47 public development tasks to ten nonoverlapping shards after a selected five-task engineering check. ACT and first-error SCHEMA ran in isolated official tool worlds under both serving modes. Shard admission required complete episodes and matched ordered requests, generated outputs, final tool states, and official user-task outcomes. Nine shards (44 tasks; 88 policy episodes per mode) passed. The three-task workspace shard was excluded in full: one task reached the configured context/prompt bound in both modes, while the other two completed.

Admitted tasks span banking, Slack, travel, and workspace. SCHEMA intervened after observed tool errors on four tasks; both policies succeeded on 21/44. The earlier five-task check also passed trajectory, state, and outcome parity, using 26 independent versus 13 shared calls.

For the 44 admitted tasks, independent/shared totals were 373/204 physical calls (45.31% saved) and 1,725,721/936,871 tokens (45.71% saved), with matched trajectories and official outcomes. Summed warmed intervals across nine distinct workers were 193.79/112.38 seconds (1.724×). Section 8 gives the coverage and timing scope of this conditional execution comparison.