MEASURE FIRST, THEN ALLOCATE: CALIBRATED ROLLOUT BUDGETS FOR MCP AGENTS IN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training tool-using language models requires costly interaction, yet identical rewards within a rollout group provide no group-relative learning signal. We use reward disagreement to guide harness evaluation and the allocation of rollouts for policy training in Model Context Protocol (MCP) environments. We first introduce **Discord**, which tests harness edits using exact paired inference and decision-preserving early stopping: only disagreeing paired outcomes favor one harness over another. This insight motivates prospective allocation in **BRACE** (**B**ank-calibrated **R**ollout **A**llocation under **C**orrelated **E**pisodes). BRACE combines historical task evidence, uncertainty in task success rates, and a correlated reward model to forecast mixed-reward groups before allocating generation budget. Controlled harness experiments favor **Discord**, while live MCP experiments expose a retention–cost trade-off. At three Qwen3-VL model scales, **BRACE** exceeds the strongest plotted baseline peaks with fewer new training rollouts; historical-bank collection is counted separately. It also leads the compared methods on BFCL and NESTFUL without further training. Ablations probe the bank, estimator, and allocation score; prospective evaluation checks forecast calibration. These results support forecasting reward contrast before generation. [1]

1 INTRODUCTION

Language agents use tools over multiple turns before a task's outcome can be verified (Zhou et al., 2024; Wang et al., 2026). The Model Context Protocol (MCP) standardizes how agent applications discover and invoke external tools and access resources (Hou et al., 2026; Anjum et al., 2026). An *agent harness* is the model-external software that configures prompts and tools and mediates the action–observation loop (Lee et al., 2026; Yang et al., 2024). Together with the model policy, it forms the agent: changing prompts or tool access changes the harness, whereas training model weights changes the policy. Automated search can improve the harness (Hu et al., 2025; Zhang et al., 2025; Lee et al., 2026), but deciding whether to retain an edit requires evaluating its effect across costly agent trajectories. An apparent gain on a finite task set may reflect noise. When the candidate and incumbent run on the same tasks, agreements cannot favor either one; only disagreements reveal the direction of the change. This motivates DISCORD, which tests paired outcomes and stops evaluation when the remaining tasks cannot change the retention decision.

A similar allocation problem arises when training that policy. Reinforcement learning with verifiable rewards (RLVR) uses task outcomes from complete trajectories (Guo et al., 2025; Wang et al., 2026). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) compares several rollouts of the same task, but a group in which all rewards agree has zero centered reward advantage. Generating such groups is especially costly when each rollout may involve repeated model inference and MCP tool calls. DAPO discards them after generation (Yu et al., 2025); PLR prioritizes tasks by estimated learning potential (Jiang et al., 2021), and VIP, TRACE, and DUET allocate generation prospectively (Nguyen et al., 2026; Zou et al., 2026; Hu et al., 2026). For multi-turn tool-use tasks, we ask whether outcomes from earlier runs can predict which fixed-size groups will contain mixed rewards, despite uncertain task success rates and correlated rollouts. BRACE uses that forecast to allocate generation

[1] Code can be accessed at: `https://anonymous.4open.science/r/brace-38DC`

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.

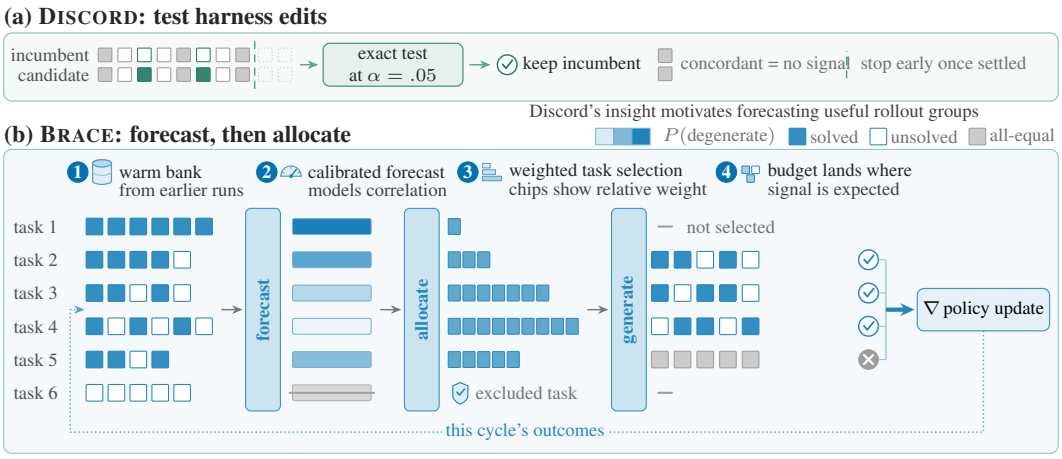


Figure 1: Observed disagreement in DISCORD motivates predictive allocation in BRACE. *(a)* Paired disagreements determine retention evidence. Even two additional candidate wins cannot pass the toy example's $\alpha = 0.05$ test, so evaluation stops. *(b)* BRACE uses ❶ historical outcomes to ❷ forecast reward unanimity and ❸ weight eligible tasks before ❹ generating one fixed-size group per selected task. Chips show relative sampling weights. The panels depict observed reward contrast for harness retention and forecast reward contrast for task allocation; all-equal rollout groups supply no group-relative reward signal.

before the next group is sampled, forming an adaptive curriculum based on the expected availability of a group-relative learning signal.

We study two complementary routes to improving tool-using agents. DISCORD uses evaluation compute to decide whether to retain a harness edit without changing model weights. Its focus on informative disagreement motivates BRACE to forecast which training tasks will produce mixed-reward groups and allocate rollouts accordingly. This directs training compute toward the policy operating within the harness (Figure 1). Our contributions are:

- **Reliable harness retention.** DISCORD uses exact paired inference and decision-preserving stopping; controlled and live experiments measure its recovery and evaluation cost.
- **Prospective task selection.** BRACE combines bounded historical evidence, uncertainty over task success, and within-group dependence to forecast informative groups. We characterize the resulting objective and test its calibration and components.
- **Training efficiency and transfer.** Across three model scales, BRACE improves held-out tool-use performance and reaches the strongest plotted baseline peaks with fewer new rollouts. It also transfers to BFCL (Patil et al., 2025) and NESTFUL (Basu et al., 2024) without further training.

2 PROBLEM FORMULATION

2.1 GROUP DEGENERACY IN RLVR

For task i, let $r_{i1}, \ldots, r_{ik} \in \{0, 1\}$ be verifier rewards for k rollouts. The centered reward component of the GRPO (Shao et al., 2024) advantage is

$$A_{ij} = r_{ij} - \bar{r}_i, \qquad \bar{r}_i = \frac{1}{k} \sum_{j'=1}^{k} r_{ij'}. \tag{1}$$

Any within-group normalization leaves the following degeneracy condition unchanged:

$$r_{i1} = \cdots = r_{ik} \quad \Longrightarrow \quad A_{ij} = 0 \quad \forall j. \tag{2}$$

This condition concerns the reward-based policy-gradient term; other regularizers may still contribute to an update. Non-degeneracy is necessary for that term to be informative, but does not guarantee a

Algorithm 1: BRACE allocation cycle: forecast and select tasks before generation.

Inputs: task pool $\mathcal{T}$, excluded set $\mathcal{D}$, counts (s_i, f_i), bank $\mathcal{B}$, group size k, task budget $N \leq |\mathcal{T} \setminus \mathcal{D}|$, discount γ, correlation $\hat{\rho}$, temperature τ, exploration rate ϵ.

1. **Update evidence:** on the first cycle, set (s_i, f_i) to capped counts from $\mathcal{B}$. On later cycles, apply $(s_i, f_i) \leftarrow \gamma(s_i, f_i) + (\Delta s_i, \Delta f_i)$ using the preceding cycle's outcomes exactly once.
2. **Forecast informative groups:** for each $i \in \mathcal{T} \setminus \mathcal{D}$, use $p_i \sim \mathrm{Beta}(1 + s_i, 1 + f_i)$ and $\nu = (1 - \hat{\rho})/\hat{\rho}$ to compute $w_i = 1 - \mathbb{E}[q(p_i)] - \mathbb{E}[q(1 - p_i)]$ via Equations 9 and 10.
3. **Allocate budget:** sample $N - n_\epsilon$ tasks without replacement with probability $\propto w_i^{1/\tau}$, where $n_\epsilon = \mathrm{round}(\epsilon N)$, and sample the remaining n_ϵ uniformly from unselected eligible tasks. Tasks in $\mathcal{D}$ receive no budget.
4. **Generate and update:** generate one group of k rollouts for each selected task, perform the scheduled policy updates, and record $(\Delta s_i, \Delta f_i)$ for the next cycle's evidence update.

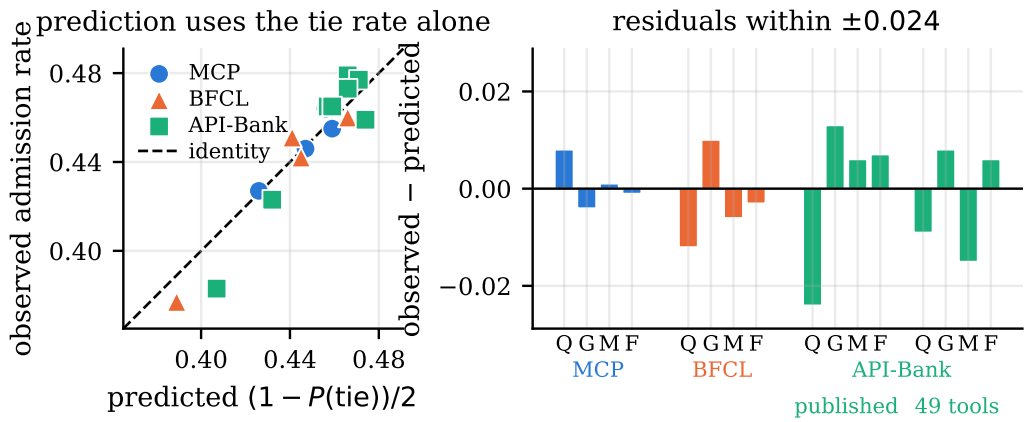


Figure 2: Inert-edit admission across AWM (Wang et al., 2026), BFCL (Patil et al., 2025), and two API-Bank (Li et al., 2023) tool surfaces. Left: predictions from independently measured tie rates versus observed admission. Right: residuals, all within ± 0.024 (Appendix C.1).

beneficial update. We quantify degeneracy in a bank snapshot of 482,510 rollouts on 1,493 verifier-scored tasks from Agent World Model (AWM) (Wang et al., 2026), a suite of SQLite-backed MCP environments scored against the final database state. It includes the 1,127-task training pool and additional tasks from the same substrate. Reconstructing original optimizer steps and retaining groups with at least two rollouts yields 92,706 groups, of which 85.2% are degenerate.

2.2 PAIRED OUTCOMES IN HARNESS EVOLUTION

Harness evolution compares an incumbent configuration A with a candidate B by evaluating both on the same tasks. Pairing controls for task difficulty and allows the decision to depend directly on how the two configurations differ on identical inputs. For binary outcomes $(A_i, B_i) \in \{0, 1\}^2$, define the two discordant counts

$$b = \#\{i : A_i = 0, B_i = 1\}, \qquad c = \#\{i : A_i = 1, B_i = 0\}. \tag{3}$$

The paired difference in mean reward can then be written as

$$\bar{B} - \bar{A} = \frac{b - c}{n}, \tag{4}$$

because pairs with $(A_i, B_i) = (0, 0)$ or $(1, 1)$ cancel exactly. Thus, only discordant pairs favor one harness over the other; $n_d = b + c$ counts informative pairs.

These identities distinguish two statistical objects. DISCORD uses paired disagreements to support a harness-retention decision; BRACE forecasts whether a same-task group will mix successes and failures, enabling a group-relative reward signal. The shared principle is to allocate interactions toward outcomes that can affect the next decision or update.

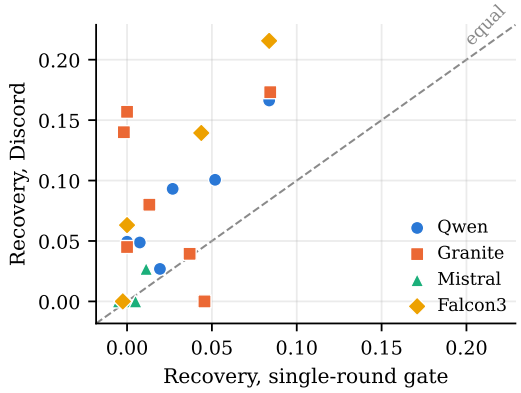


Figure 3: Harness recovery across four callers: $18/21$ paired loops favor DISCORD. The dashed line marks equal recovery.

Table 1: Live AWM (Wang et al., 2026) retention and cost. Accepted edits are classified as good/bad by measured effect; objective and episode counts report realized outcomes.

Regime	Rule	Accepted (good/bad)	Final objective	Episodes
Shallow	Deployed	3 (2/1)	0.2656	1,760
	DISCORD	0 (0/0)	0.2594	1,906
Deep	Deployed	3 (3/0)	0.2812	1,920
	DISCORD	2 (2/0)	0.2703	4,008

Shallow and deep regimes withhold 10% and 40% of tool names. The full-surface objective is 0.2813.

3 DISCORD: EXACT HARNESS RETENTION

DISCORD evaluates proposed harness edits, holding model weights fixed. It retains an edit only when paired outcomes provide sufficient evidence of improvement; it does not generate proposals.

3.1 WHY A POSITIVE MEASURED GAIN IS INSUFFICIENT

An *inert edit* leaves true performance unchanged. If incumbent and candidate scores are exchangeable, apparent improvements and regressions are equally likely:

$$P(\text{retain}) = P(\bar{B} > \bar{A}) = \tfrac{1}{2}\big(1 - P(\text{tie})\big). \tag{5}$$

When ties are rare, strict-improvement retention accepts nearly half of inert edits. Appendix C.1 gives the derivation and measurement protocol.

3.2 TESTING WHETHER THE CANDIDATE WINS RELIABLY

Under the paired null, a disagreement is equally likely to favor the candidate or the incumbent. Assuming independent evaluation pairs and conditioning on the observed number of disagreements $n_d = b + c$, the number of candidate wins follows McNemar's exact conditional distribution (McNemar, 1947):

$$b \mid n_d \sim \text{Binomial}(n_d, 1/2). \tag{6}$$

DISCORD retains the candidate when the upper-tail probability is at most a prespecified α. For example, $b = 12$ and $c = 4$ on 100 paired tasks give an 8 pp improvement and an exact one-sided probability of 0.0384, passing at $\alpha = 0.05$. With no disagreements, retention is rejected. Operationally, each paired task either increments b, increments c, or ties and changes neither count. Conditioning on n_d removes the unknown tie rate: the test asks whether the directions of observed disagreements favor the candidate, independently of how often the harnesses agree. Error control applies to this prespecified comparison, not an unrestricted adaptive search (Appendix C.2).

Stopping without changing the decision. The maximum number of paired tasks is fixed in advance. After each block, *exact curtailment* (Alling, 1963) checks whether all possible remaining pairs (candidate wins, incumbent wins, or ties) lead to the same terminal decision. If so, evaluation stops; otherwise it continues.

For example, with $b = 2$, $c = 0$ and two pairs remaining, the most favorable completion for the candidate is $b = 4$, $c = 0$. Its one-sided probability is $1/16 = 0.0625$, so it still fails an $\alpha = 0.05$ test; the procedure can immediately keep the incumbent. This is different from repeatedly stopping when a provisional p-value happens to cross the threshold: exact curtailment always returns the decision

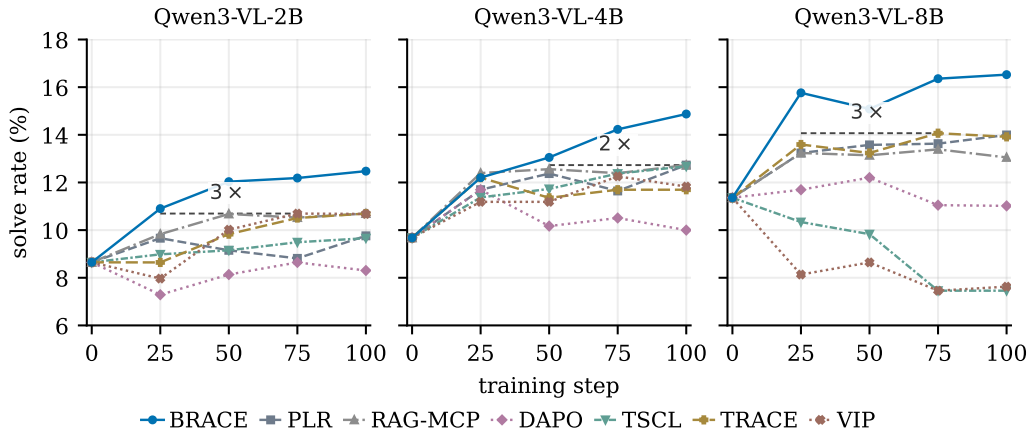


Figure 4: Observed-checkpoint comparison of new rollouts (160 attempts/step). For each scale, the horizontal dashed segment spans the earliest peak checkpoint of the strongest plotted baseline and BRACE's first measured checkpoint that meets or exceeds that solve rate. Labels give the baseline/BRACE rollout ratio at those checkpoints; bank collection is excluded. Table 2 reports final-checkpoint results.

that full evaluation would have returned. Ambiguous comparisons can still require the entire budget (Appendices C.2 and C.3).

From observed to predicted disagreement. DISCORD identifies when further paired outcomes cannot change a harness decision. This suggests a training question: which tasks are likely to yield mixed rewards before their rollouts are generated? BRACE forecasts that signal from historical task outcomes and allocates the next budget accordingly. We evaluate its forecasts and policy gains separately.

4 BRACE: CORRELATION-AWARE ROLLOUT ALLOCATION

BRACE selects tasks before generating their next rollout groups (Algorithm 1). It favors tasks predicted to produce both successes and failures while retaining uniform exploration. Main experiments select $N = 256$ distinct tasks per allocation cycle, then generate one $k = 5$ group per task over eight optimizer steps of 32 tasks each. Allocation changes task selection, not group size. Thus, the score targets the chance of receiving a group-relative reward signal, rather than the highest expected solve rate; the exploration share continues to sample eligible tasks regardless of their scores. Algorithm 1 records each selected batch and its predicted degeneracy before generation, enabling prospective calibration checks.

4.1 ALLOCATION OBJECTIVE AND BANKED EVIDENCE

For task solve probability p_i, an independent group is degenerate if all k rollouts succeed or all fail. The complementary probability of mixed rewards is

$$g(p_i, k) = 1 - p_i^k - (1 - p_i)^k. \tag{7}$$

Under independence, we account for uncertainty in p_i by integrating over its posterior:

$$w_i^{\text{ind}} = \mathbb{E}[g(p_i, k)], \qquad p_i \sim \text{Beta}(a_i, b_i), \quad \mathbb{E}[p_i^k] = \prod_{j=0}^{k-1} \frac{a_i + j}{a_i + b_i + j}. \tag{8}$$

Tasks with the same posterior mean can receive different scores when their evidence counts differ.

Banked evidence. We approximate uncertainty in the task mean by $p_i \sim \text{Beta}(1 + s_i, 1 + f_i)$ using success and failure counts (s_i, f_i); this is not the exact posterior under the correlated group

likelihood. A *warm bank* provides success and failure counts from previous runs across model scales, capped so a large historical log cannot dominate later evidence. It is read once to initialize task-level evidence for allocation. Subsequently, the counts are discounted and incremented with current-run outcomes; the bank is not reread. Every rollout used for a policy update is generated by the current policy. Without bank observations, unseen tasks all start at $\mathrm{Beta}(1,1)$, so allocation cannot initially distinguish tasks that rarely solve from those that produce mixed outcomes. Bank counts make that distinction available before the first current-run rollout; capping limits their influence as new evidence arrives. Appendix D.2 gives the exact initialization and cost accounting.

4.2 Modeling Correlated Rollout Groups

Rollout rewards within the same task group need not be independent: shared task difficulty and sampling conditions can make the outcomes move together. We model this dependence with an exchangeable Beta–Binomial distribution with mean p_i and concentration $\nu = (1-\rho)/\rho$, where ρ is the within-group correlation. Equivalently, for each group we draw a latent success propensity $\theta_i \sim \mathrm{Beta}(\nu p_i, \nu(1-p_i))$, then draw the k rewards conditionally independently given θ_i. This group-level draw is distinct from the posterior uncertainty over the task mean p_i. Under this model,

$$q(p_i) = P(\text{all } k \text{ succeed} \mid p_i) = \prod_{j=0}^{k-1} \frac{\nu p_i + j}{\nu + j}, \tag{9}$$

and $q(1-p_i)$ is the probability of unanimous failure. The allocation score is

$$w_i = 1 - \mathbb{E}[q(p_i)] - \mathbb{E}[q(1-p_i)]. \tag{10}$$

Under the Beta evidence approximation and exchangeable group model, w_i is the posterior predictive probability that task i's next group contains both successes and failures. For one group per selected task, $\sum_{i \in S} w_i$ is the model's expected informative-group count (Appendix D.4). This is a signal-availability objective, not a policy-improvement guarantee. The independent case is recovered as $\nu \to \infty$. Positive correlation raises the probability of unanimous groups; for example, at $p_i = 0.5$, $k = 5$, and $\nu = 0.285$, the mixed-reward probability is 0.23, compared with 0.94 under independence. The expectations in Equation 10 are evaluated exactly using Beta moments, without sampling (Appendix D.3).

4.3 Rank Preservation under Correlation Correction

Proposition 1. *For $k \geq 2$ and $\nu > 0$, let $f(p) = \prod_{j=0}^{k-1}(\nu p + j)/(\nu + j)$. Then $g_c(p) = 1 - f(p) - f(1-p)$ is strictly increasing in $p(1-p)$. Consequently, at fixed k and shared ν, known-probability scores $g_c(p)$, $1 - p^k - (1-p)^k$, and the Bernoulli variance $p(1-p)$ induce the same task ordering.*

For the weighted sampling step, the relative weight of eligible tasks i and j is $(w_i/w_j)^{1/\tau}$. A common rescaling of all scores cancels, while a nonlinear transformation can change these ratios even when ranks agree. Posterior integration can also change rankings because evidence counts differ across tasks. The mechanism therefore depends on the estimated evidence and relative weights, alongside forecast calibration. Appendix E.2 proves the proposition; Section 5.5 connects it to sampling controls and training ablations. The theorem concerns fixed group sizes.

5 Experiments

5.1 Experimental setup

Models and baselines. We train Qwen3-VL-Instruct (Bai et al., 2025) at 2B, 4B, and 8B using GRPO (Shao et al., 2024) in `verl` (Sheng et al., 2025), with rank-32 LoRA. Every method generates 160 attempts per training step (32 prompts, $k = 5$), counting discarded and incomplete attempts. The AWM (Wang et al., 2026) training pool contains 1,127 verifier-scored tasks over 305 live MCP servers under a fixed tool surface. Baselines include uniform GRPO (Shao et al., 2024), PLR (Jiang et al., 2021), RAG-MCP (Gan & Sun, 2025), DAPO (Yu et al., 2025), TSCL (Matiisen et al., 2020), TRACE (Zou et al., 2026), and VIP (Nguyen et al., 2026) (Table 2). Methods share the model,

Table 2: Held-out AWM (Wang et al., 2026) results after 100 training steps. Higher solve rate, lower invalid-call rate, and fewer turns are better. Bold marks the best trained method per metric and scale; evaluation uses four seeds.

Method	Qwen3-VL-2B			Qwen3-VL-4B			Qwen3-VL-8B		
	solve (%)↑	invalid (%)↓	turns ↓	solve (%)↑	invalid (%)↓	turns ↓	solve (%)↑	invalid (%)↓	turns ↓
base model	8.6	16.6	14.7	9.7	6.5	8.4	11.4	6.8	9.6
BRACE	**12.5**	19.3	12.1	**14.9**	4.6	10.0	**16.5**	9.0	9.0
PLR (Jiang et al., 2021)	9.8	20.0	**7.6**	12.7	6.9	11.3	14.0	9.1	10.2
RAG-MCP (Gan & Sun, 2025)	10.7	29.4	11.6	12.7	**3.0**	10.9	13.1	**3.6**	11.6
DAPO (Yu et al., 2025)	8.3	18.9	14.2	10.0	6.3	11.6	11.0	8.0	12.3
TSCL (Matiisen et al., 2020)	9.7	**10.4**	10.3	12.7	5.5	8.4	7.5	9.3	**3.7**
TRACE (Zou et al., 2026)	10.7	30.2	9.1	11.7	6.9	13.0	13.9	6.6	10.3
VIP (Nguyen et al., 2026)	10.7	14.3	13.6	11.8	5.0	**6.7**	7.6	6.4	11.0
uniform GRPO (Shao et al., 2024)	8.9	29.9	11.2	11.4	9.8	10.1	13.2	13.0	8.7

optimizer, current-run rollout budget, and evaluation protocol; BRACE additionally uses historical bank evidence.

Eligibility checks. We exclude 104 of 1,127 training tasks: 66 whose verifier succeeds on the unchanged initial state, 33 proved unreachable by an initial-state guard, and 5 flagged by a string-matching heuristic, reported separately from certificates. The guard never fires on the bank's 826 tasks solved at least once; excluded tasks produce no usable learning groups in historical data. These checks determine eligibility before posterior weighting: failures alone do not prove impossibility.

BRACE configuration. Historical counts are capped at 20 successes and 40 failures per task, with $\gamma = 1$. We use $\nu = 0.285$ ($\hat{\rho} \approx 0.78$) and $\tau = 0.5$ at all scales; uniform exploration is $\epsilon = 0.05$ at 2B/4B and 0.10 at 8B. The correlation parameter is fitted once on the first two cycles of the 8B calibration run and then held fixed. Those cycles are excluded from forecast evaluation. Appendix D.2 documents bank provenance and costs.

Evaluation. We evaluate on 295 held-out AWM (Wang et al., 2026) tasks, excluded from training and bank initialization, using four evaluation seeds. BFCL v4 (Patil et al., 2025) and NESTFUL (Basu et al., 2024) supply no AWM (Wang et al., 2026) training data. Main AWM (Wang et al., 2026) results use step 100; paired contrasts use matched task indices.

5.2 DISCORD: HARNESS RETENTION

Inert-edit admission. Independently measured tie rates predict inert-edit admission within ± 0.024 across AWM (Wang et al., 2026), BFCL (Patil et al., 2025), and two API-Bank (Li et al., 2023) tool surfaces (Figure 2). On AWM (Wang et al., 2026), strict-improvement retention accepts 43.3% of 1,440 inert edits, and 43.6% of the accepted edits fail to replicate (Appendices C.2 and C.1).

Controlled recovery and harmful proposals. We compare retention rules on 200 BFCL v3 (Patil et al., 2025) single-turn tasks, using four callers and eight rounds of three edits to advertised function names. Withheld names create a recoverable handicap; a full surface makes proposals predominantly harmful. Shared proposals and matched episode budgets isolate retention from edit generation (Appendix C.4). The single-round gate uses a label-permutation test on disjoint task halves; DISCORD uses discordant outcomes from the same tasks. DISCORD recovers more of the handicap than the single-round gate in 18 of 21 paired loops (Figure 3; one-sided paired t-test, $p = 0.00015$). Under harmful proposals, degradation occurs in $0/17$ seeds for DISCORD and $9/17$ for strict-improvement retention. The seeds are matched; a two-sided exact McNemar test gives $p = 0.0039$.

Live MCP trade-off. On a shared proposal stream across 320 tasks and 158 AWM (Wang et al., 2026) servers, shallow and deep handicaps withhold 10% and 40% of tool names. DISCORD accepts no harmful edits in either regime, but uses more evaluation episodes and reaches a lower final objective than the deployed rule (Table 1). The deployed rule overstates its final objective by 32% and 47%,

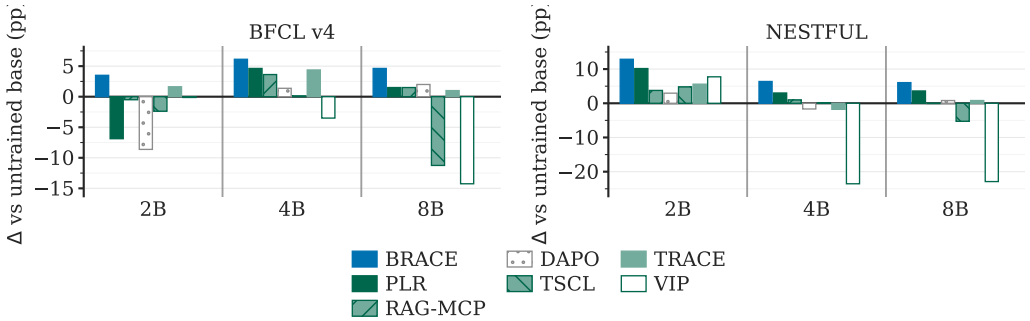


Figure 5: Transfer to BFCL (Patil et al., 2025) and NESTFUL (Basu et al., 2024) without further training. Bars show percentage-point changes from the corresponding untrained base model, not absolute rates. BRACE has the largest gain among the plotted methods at each scale.

respectively. This trade-off favors DISCORD when avoiding harmful edits warrants more evaluation; conservative retention can miss useful edits under limited evidence.

5.3 HELD-OUT MCP PERFORMANCE

BRACE leads all baselines at 2B, 4B, and 8B (Table 2), solving 12.5%, 14.9%, and 16.5% of tasks. Uniform GRPO (Shao et al., 2024) reaches 8.9%, 11.4%, and 13.2%; BRACE gains 3.6, 3.5, and 3.3 percentage points (pp). The strongest baselines reach 10.7%, 12.7%, and 14.0%, giving margins of 1.8, 2.2, and 2.5 pp. Section 5.5 separates bank and allocation effects. At 2B, 4B, and 8B, BRACE exceeds the strongest plotted baseline peaks using 4,000, 8,000, and 4,000 new rollouts, versus 12,000, 16,000, and 12,000 at the baselines' earliest peak checkpoints. These observed comparisons require no interpolation. Figure 4 plots solve rate against training steps; dashed segments span the measured checkpoints used for the rollout ratios. RAG-MCP (Gan & Sun, 2025) has the lowest invalid-call rates at 4B and 8B; PLR (Jiang et al., 2021), VIP (Nguyen et al., 2026), and TSCL (Matiisen et al., 2020) use the fewest turns at 2B, 4B, and 8B. Table 2 thus shows better task completion without consistently shorter trajectories for BRACE.

5.4 GENERALIZATION

Transfer tests multi-turn function calling on BFCL (Patil et al., 2025) and nested function composition on NESTFUL (Basu et al., 2024), without further adaptation. Figure 5 shows percentage-point gains over untrained bases; BRACE leads at all three scales. NESTFUL (Basu et al., 2024) win rate evaluates the executed answer, not exact call-sequence matching. Our modified prompt and parsing conventions support within-study comparisons, not direct comparison with published NESTFUL (Basu et al., 2024) scores.

5.5 ABLATION STUDY

Table 3 shows lower solve rates in every component ablation at all three scales. Removing the bank costs up to 4.3 pp; removing correlation correction costs up to 4.6 pp. Plug-in estimation, per-scale correlation fitting, and the VIP-style (Nguyen et al., 2026) variance score also underperform. The score comparison holds bank counts, estimator, temperature, exploration, and eligibility fixed. **Mechanism and cost controls.** The bank reuses episodes from earlier 2B, 4B, and 8B runs. It is read once at initialization; no dedicated collection run was launched, although the historical generation cost remains separate from current-run rollouts. The VIP-score (Nguyen et al., 2026) and correlation-removal ablations start from the same capped per-task bank counts as full BRACE (Appendix D.2). On 83 prospective arm-cycles that exclude the cycles used to fit ρ, correlation-aware forecasts reduce mean absolute degeneracy-prediction error from 0.591 to 0.056 (Appendix E.1). This tests forecast calibration, while held-out solve rates assess learning; Proposition 1 also limits what calibration alone can imply about task ranking.

Table 3: Component comparisons and a uniform-GRPO (Shao et al., 2024) reference on held-out AWM (Wang et al., 2026) tasks. "Δ vs full" and "effect vs base" are percentage-point differences computed from the displayed solve rates. The full method has the highest solve rate at all three model scales.

ablation	Qwen3-VL-2B			Qwen3-VL-4B			Qwen3-VL-8B		
	solve (%)	Δ vs full	effect vs base	solve (%)	Δ vs full	effect vs base	solve (%)	Δ vs full	effect vs base
BRACE (full)	**12.5**	–	+3.9	**14.9**	–	+5.2	**16.5**	–	+5.1
w/o warm bank	9.2	−3.3	+0.6	12.8	−2.1	+3.1	12.2	−4.3	+0.8
w/o correlation correction	10.0	−2.5	+1.4	10.3	−4.6	+0.6	14.4	−2.1	+3.0
estimator: posterior → plug-in	10.7	−1.8	+2.1	13.1	−1.8	+3.4	15.2	−1.3	+3.8
correlation ρ: one → per-scale	11.3	−1.2	+2.7	13.9	−1.0	+4.2	15.6	−0.9	+4.2
score → VIP-style variance	10.6	−1.9	+2.0	13.2	−1.7	+3.5	14.1	−2.4	+2.7
reference: uniform GRPO	8.9	−3.6	+0.3	11.4	−3.5	+1.7	13.2	−3.3	+1.8

6 RELATED WORK

Task selection and rollout allocation. Adaptive curricula change which tasks an agent encounters. PLR prioritizes levels using TD error (Jiang et al., 2021), while problem-level RL replay uses empirical intermediate success rates (Fatemi, 2026). Other selectors use reward dynamics (Zheng et al., 2025b), task difficulty (Qu et al., 2025; 2026; Gao et al., 2025), estimated learning impact (Li et al., 2025), or replay priorities (Sun et al., 2025). For group-relative training, DAPO filters all-equal groups after generation (Yu et al., 2025). VIP allocates rollout counts by predicted gradient variance (Nguyen et al., 2026); DUET also allocates token budgets (Hu et al., 2026), and TRACE predicts mixed outcomes at prompts and interaction prefixes using an independent rollout model (Zou et al., 2026). BRACE instead selects tasks for fixed-size groups using capped historical evidence, posterior uncertainty, and a correlated reward model. The correlation correction improves forecast calibration while preserving known-probability task ranks at fixed group size (Proposition 1); historical evidence and relative weights therefore matter to the resulting allocation. Policy-gradient estimators (Liu et al., 2025; Ahmadian et al., 2024; Zheng et al., 2025a) address a separate choice.

Tool-using agents and harness search. Multi-turn training studies delayed rewards and tool interaction (Zhou et al., 2024; Wang et al., 2025b; Jin et al., 2025; Qian et al., 2025). MCP work studies protocol architecture and overhead, tool discovery, and agent evaluation (Hou et al., 2026; Anjum et al., 2026; Fei et al., 2025; Guo et al., 2026; Wang et al., 2025a), while benchmarks measure tool execution and function composition (Patil et al., 2025; Basu et al., 2024; Qin et al., 2023; Yao et al., 2024). Harness-search systems propose edits to prompts, workflows, or agent code (Hu et al., 2025; Zhang et al., 2025; Agrawal et al., 2026; Lee et al., 2026; Zelikman et al., 2024; Zhang et al., 2026). DISCORD addresses the subsequent retention decision with exact conditional paired inference (McNemar, 1947) and decision-preserving curtailment (Alling, 1963); alternative matched-pair tests include exact unconditional procedures (Suissa & Shuster, 1991).

Statistical foundations. Beta–Binomial models describe clustered binary outcomes (Kleinman, 1973; Crowder, 1978; Williams, 1975; Ridout et al., 1999) and motivate BRACE's within-group forecast. Its task selection is related to active learning (Settles, 2009; Houlsby et al., 2011) and adaptive replay (Schaul et al., 2016), with non-degenerate group probability as the allocation objective.

7 CONCLUSION

Reward disagreement provides a common lens for allocating evaluation and training computation. DISCORD uses exact paired inference and decision-preserving stopping to retain harness edits when observed disagreements support improvement. BRACE uses capped historical evidence, posterior uncertainty, and reward dependence to forecast mixed-reward groups before generation. Across three model scales, BRACE achieves higher observed held-out MCP solve rates than the compared methods, reaches their strongest plotted peaks with fewer new rollouts, and transfers to BFCL and NESTFUL without further training. Together, the methods direct interaction budget toward harness comparisons that can change retention and rollout groups that can supply a group-relative training signal.

AI USE STATEMENT

We used generative AI tools to assist with coding, including implementation, debugging, and analysis scripts, and with reviewing relevant literature. The authors directed these uses, checked the AI-assisted code and outputs, and verified the resulting text and citations. All AI-assisted work was reviewed by the authors, who take responsibility for the final content.

ETHICS STATEMENT

Improved tool-use efficiency can support beneficial applications but can also increase the scale of unintended or unauthorized actions. Benchmark success and lower harmful-edit retention do not establish deployment safety: "harmful" here means reduced benchmark performance, not a comprehensive safety assessment. Real-world use requires appropriate permissions, execution isolation, monitoring, and safeguards beyond the reward verifier. Task exclusion and historical-data reuse may also favor tasks already represented in the bank; our results do not establish equitable performance across users or application domains.

REPRODUCIBILITY STATEMENT

Appendix B specifies model and training settings, task splits, evaluation procedures, and checkpoint selection. Appendix D.2 describes the source, caps, and initialization of historical evidence. Allocation forecasts are recorded before the corresponding rollouts are generated; paired tests use matched task indices. The code repository is available at `https://anonymous.4open.science/r/brace-38DC`.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024.

David W. Alling. Early decision in the wilcoxon two-sample test. *Journal of the American Statistical Association*, 58(303):713–720, 1963.

Sumera Anjum, Weijian Zheng, Rajkumar Kettimuthu, Heng Fan, and Yunhe Feng. ProMCP: Profiling token flows and latency costs in model context protocol–based LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 39476–39487, 2026. doi: 10.18653/v1/2026.findings-acl.1967. URL `https://aclanthology.org/2026.findings-acl.1967/`.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL `https://arxiv.org/abs/2511.21631`.

Kinjal Basu, Ibrahim Abdelaziz, Kiran Kate, Mayank Agarwal, Maxwell Crouse, Yara Rizk, Kelsey Bradford, Asim Munawar, Sadhana Kumaravel, Saurabh Goyal, Xin Wang, Luis A. Lastras, and

Pavan Kapanipathi. NESTFUL: A benchmark for evaluating LLMs on nested sequences of API calls. *arXiv preprint arXiv:2409.03797*, 2024.

M. J. Crowder. Beta-binomial ANOVA for proportions. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 27(1):34–37, 1978.

Mehdi Fatemi. Prioritized replay for RL post-training. *arXiv preprint arXiv:2601.02648*, 2026.

Xiang Fei, Xiawu Zheng, and Hao Feng. Mcp-zero: Active tool discovery for autonomous llm agents. *arXiv preprint arXiv:2506.01056*, 2025.

Tiantian Gan and Qiyao Sun. RAG-MCP: Mitigating prompt bloat in llm tool selection via retrieval-augmented generation. *arXiv preprint arXiv:2505.03275*, 2025.

Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. Prompt curriculum learning for efficient LLM post-training. *arXiv preprint arXiv:2510.01135*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Zikang Guo, Benfeng Xu, Chiwei Zhu, Wentao Hong, Xiaorui Wang, and Zhendong Mao. MCP-AgentBench: Evaluating real-world language agent performance with MCP-mediated tools. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30888–30896, 2026. doi: 10.1609/aaai.v40i37.40347.

Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. Model context protocol (MCP): Landscape, security threats, and future research directions. *ACM Transactions on Software Engineering and Methodology*, 35(10):1–37, 2026. doi: 10.1145/3796519.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Haoyu Hu, Xuandong Zhao, Xuhai Xu, and Nori Jacoby. DUET: Optimize token-budget allocation for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2605.08441*, 2026.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations (ICLR)*, 2025.

IBM Granite Team. Granite-3.3-8B-Instruct. Model card, 2025. URL `https://huggingface.co/ibm-granite/granite-3.3-8b-instruct`. Accessed September 10, 2026.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning (ICML)*, volume 139, pp. 4940–4950, 2021.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Joel C. Kleinman. Proportions with extraneous variance: Single and independent samples. *Journal of the American Statistical Association*, 68(341):46–54, 1973.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-Harness: End-to-end optimization of model harnesses. In *Conference on Language Modeling (COLM)*, 2026. URL `https://arxiv.org/abs/2603.28052`.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. arXiv:2304.08244.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. LIMR: Less is more for RL scaling. *arXiv preprint arXiv:2502.11886*, 2025.

Zichen Liu, Changyu Chen, Wenjun Li, et al. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, 2020.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Mistral AI. Mistral-7B-Instruct-v0.3. Model card, 2024. URL `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3`. Accessed September 10, 2026.

Hieu Trung Nguyen, Bao Nguyen, Wenao Ma, Yuzhi Zhao, Ruifeng She, and Viet Anh Nguyen. Adaptive rollout allocation for online reinforcement learning with verifiable rewards. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2602.01601.

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 *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392, 2025.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2504.13958.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. *arXiv preprint arXiv:2307.16789*, 2023.

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating RL finetuning of reasoning models? *arXiv preprint arXiv:2507.04632*, 2025.

Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Weijie Liu, Yu Bai, Kai Yang, Wei Chen, Saiyong Yang, and Xiangyang Ji. Small generalizable prompt predictive models can steer efficient RL post-training of large reasoning models. *arXiv preprint arXiv:2602.01970*, 2026.

Martin S. Ridout, Clarice G. B. Demétrio, and David Firth. Estimating intraclass correlation for binary data. *Biometrics*, 55(1):137–148, 1999.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations (ICLR)*, 2016.

Burr Settles. Active learning literature survey. Technical Report Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. Revised January 2010.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, 2025. arXiv:2409.19256; the VERL framework.

Samy Suissa and Jonathan J. Shuster. The 2×2 matched-pairs trial: Exact unconditional design and analysis. *Biometrics*, 47(2):361–372, 1991.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for LLM reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*, 2025.

Technology Innovation Institute. Falcon3-7B-Instruct. Model card, 2024. URL https://huggingface.co/tiiuae/Falcon3-7B-Instruct. Accessed September 10, 2026.

Zhaoyang Wang, Canwen Xu, Boyi Liu, Yite Wang, Siwei Han, Zhewei Yao, Huaxiu Yao, and Yuxiong He. Agent World Model: Infinity synthetic environments for agentic reinforcement learning. *arXiv preprint arXiv:2602.10090*, 2026.

Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, et al. MCP-Bench: Benchmarking tool-using llm agents with complex real-world tasks via MCP servers. *arXiv preprint arXiv:2508.20453*, 2025a.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025b.

D. A. Williams. The analysis of binary responses from toxicological experiments involving reproduction and teratogenicity. *Biometrics*, 31(4):949–952, 1975.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1601. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, et al. DAPO: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (STOP): Recursively self-improving code generation. In *Conference on Language Modeling (COLM)*, 2024.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations (ICLR)*, 2026.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*, 2025b.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62178–62209, 2024.

Heming Zou, Qi Wang, Yun Qu, Yuhang Jiang, Lizhou Cai, Yixiu Mao, Ru Peng, Xin Xu, Weijie Liu, Kai Yang, Saiyong Yang, and Xiangyang Ji. TRACE: A unified rollout budget allocation framework for efficient agentic reinforcement learning. *arXiv preprint arXiv:2606.11119*, 2026.

This appendix discusses limitations and provides experimental protocols, retention analyses, implementation details, and calibration controls supporting the main results.

A LIMITATIONS

BRACE predicts whether a rollout group has nonzero centered reward, not whether its gradient improves the policy. The expected-yield interpretation in Section 4.2 therefore explains the allocation target but does not establish that informative-group yield causes the observed solve-rate gains. The bank uses rollouts generated by earlier policies and model scales. Although no dedicated bank-collection run was launched here, those historical rollouts have a cost separate from the current-run budget; cold-start or shifted task pools may provide less useful prior evidence. The experiments use binary verifier rewards, fixed-size groups, and the tested MCP and transfer benchmarks; other rewards, group sizes, and tool environments remain untested.

DISCORD's exact test controls a prespecified paired comparison, not an unrestricted sequence of adaptively proposed edits. Its conservative retention can increase evaluation cost and miss useful edits under limited evidence, as the live MCP comparison illustrates. It is therefore a safeguard for individual retention decisions rather than an optimal harness-search policy.

B EXPERIMENTAL SETUP AND EVALUATION

B.1 MODELS, TRAINING DATA, AND OPTIMIZATION

We train Qwen3-VL-Instruct models at 2B, 4B, and 8B (Bai et al., 2025), using GRPO (Shao et al., 2024) in `verl` (Sheng et al., 2025). The training substrate is Agent World Model (AWM) (Wang et al., 2026), whose MCP tools operate on SQLite-backed state. A binary verifier evaluates the final database state. Training uses 1,127 tasks across 305 live servers with a fixed tool surface. The held-out AWM (Wang et al., 2026) set contains 295 tasks disjoint from the training pool.

The bank snapshot used for the degeneracy diagnostic contains 482,510 episodes on 1,493 tasks from the same substrate, including the training pool. Reconstructing the original optimizer steps and retaining groups with at least two rollouts produces 92,706 groups, of which 85.2% are degenerate. These bank-wide counts are distinct from the task count and group composition of an individual run. Only training-task outcomes initialize the allocator. All 295 held-out AWM (Wang et al., 2026) tasks are excluded from the initialization bank as well as from training.

Table 4: Core training configuration. Allocation changes task selection before rollout generation; evaluation uses a separate held-out task set.

Setting	Value
Model family	Qwen3-VL-Instruct (2B, 4B, 8B)
Training framework and objective	`verl`, GRPO
LoRA rank	32
Prompts per optimizer step	32
Rollouts per prompt	5
Episodes per optimizer step	160
Training pool	1,127 tasks, 305 MCP servers
Held-out MCP evaluation	295 tasks, four evaluation seeds
Training replication	One training seed per configuration

B.2 BASELINES AND COMPARISON SCOPE

The main comparison includes uniform GRPO (Shao et al., 2024), Prioritized Level Replay (PLR) (Jiang et al., 2021), RAG-MCP (Gan & Sun, 2025), DAPO (Yu et al., 2025), Teacher–Student Curriculum Learning (TSCL) (Matiisen et al., 2020), TRACE (Zou et al., 2026), and VIP (Nguyen et al., 2026). These methods provide comparisons to uniform sampling, replay and curriculum selection, tool retrieval, post-generation filtering, and pre-generation allocation. The comparisons use the same model, optimizer, current-run rollout budget, and evaluation protocol as specified

in Section 5.1. Table 2 compares final-checkpoint performance, and Figure 5 evaluates transfer. BRACE's historical warm-bank initialization is described in Appendix D.1; Table 3 evaluates its removal alongside the other component comparisons.

The comparison unit matters for interpreting efficiency. Removing a degenerate group after generation does not recover the episodes already executed. Likewise, a warm bank avoids new probing episodes but requires previously collected data. We distinguish the generation budget of the current run from historical data acquisition and from the number of groups ultimately used for optimization.

B.3 Statistical reporting

For two policies evaluated on the same task–seed pairs, let b count pairs won only by the candidate and c count pairs won only by the reference. The paired solve-rate difference is $(b-c)/n$, where n includes both concordant and discordant pairs. Effects expressed in percentage points are $100(b-c)/n$; a change from 11.4% to 14.9%, for example, is 3.5 pp. The exact conditional McNemar test (McNemar, 1947) evaluates the discordant count under $b \mid b+c \sim \mathrm{Binomial}(b+c, 1/2)$, assuming independent pairs and an exchangeable paired null. Repeated seeds on the same task need not form independent pairs; the difference identity remains valid without this assumption, but exact binomial error control does not.

Holm correction is applied within each registered comparison family. For m ordered unadjusted values $p_{(1)} \le \cdots \le p_{(m)}$, the adjusted value at rank j is

$$\widetilde{p}_{(j)} = \min\left\{1, \max_{\ell \le j}(m-\ell+1)p_{(\ell)}\right\}.$$

Family membership is determined by the registered analysis, rather than by which contrasts are favorable. Paired significance and absolute performance are reported separately; the aggregation conventions are specified below.

B.4 Evaluation and aggregation across runs

The training and evaluation seed counts are specified in Table 4. Repeated evaluations characterize checkpoint-level stochasticity; they do not estimate variability across independently trained policies. Main results use the final plotted checkpoint, while learning curves retain the intermediate evaluations.

B.5 Transfer benchmarks and processing

BFCL v4 (Patil et al., 2025) evaluates multi-turn function calling on 800 tasks from four categories: `base`, `long_context`, `miss_func`, and `miss_param`, with 200 tasks each. The official checker evaluates the executed result against the reference trajectory. The evaluation uses a common task index across arms and one response per task at temperature 0.001. Transfer evaluation uses checkpoints from AWM (Wang et al., 2026) training, without further training on BFCL (Patil et al., 2025). The v4 multi-turn categories are separate from the v3 single-turn split used for harness-retention experiments in Appendix C.4.

NESTFUL (Basu et al., 2024) evaluates nested function composition. A completion specifies API calls with dependencies, which are executed against the benchmark implementations. The win-rate metric checks whether execution yields the recorded answer; it differs from the benchmark's full-sequence matching metric. Evaluation uses $1{,}861$ tasks, one completion per task, and the same task index across arms. Two processing changes relative to the released Llama branch are applied uniformly: function specifications are embedded as JSON rather than as double-encoded strings, and completions are parsed from the first balanced top-level `[...]` span. These changes preserve the within-study comparison but prevent direct numerical comparison with results obtained under the benchmark's original processing convention.

Figure 5 reports percentage-point changes in BFCL (Patil et al., 2025) pass rate and NESTFUL (Basu et al., 2024) win rate relative to each scale's untrained base model, rather than absolute rates. Neither benchmark supplies training data or task-selection feedback to the reported AWM (Wang et al., 2026) training runs. Because the benchmarks use different task structures and success criteria, their rates should be interpreted within each benchmark.

C DISCORD: DERIVATIONS AND RETENTION EXPERIMENTS

C.1 INERT-EDIT ADMISSION IDENTITY

Let $D = \bar{B} - \bar{A}$ under an inert edit. Exchangeability implies that D and $-D$ have the same distribution, so $P(D > 0) = P(D < 0)$. Partitioning the outcome space into positive, negative, and zero differences yields

$$1 = P(D > 0) + P(D < 0) + P(D = 0), \qquad P(D > 0) = \tfrac{1}{2}[1 - P(D = 0)].$$

This proves Equation 5. The result concerns strict-improvement retention; a different treatment of ties changes the admission probability.

We measure tie rates and inert-edit admission separately on AWM (Wang et al., 2026), BFCL (Patil et al., 2025), and API-Bank (Li et al., 2023). The AWM (Wang et al., 2026) diagnostic uses $6{,}000$ draws per caller (12 ordered interface pairs with 500 draws each). BFCL (Patil et al., 2025) and API-Bank (Li et al., 2023) use $1{,}500$ group-of-four resampling trials per caller. These estimators test the same identity at different sampling units; their rates are not pooled. API-Bank's (Li et al., 2023) published and 49-tool surfaces are two settings of one substrate, not independent benchmarks. Table 5 provides the values underlying Figure 2.

Table 5: Independent predictions and measurements of inert-edit admission. Difference is observed minus predicted admission probability.

Substrate	Caller	Tie rate	Predicted	Observed	Difference
AWM (Wang et al., 2026)	qwen	0.088	0.456	0.464	+0.008
	granite	0.082	0.459	0.455	−0.004
	mistral	0.149	0.426	0.427	+0.001
	falcon3	0.107	0.447	0.446	−0.001
	mean	0.106	**0.447**	**0.448**	+0.001
BFCL (Patil et al., 2025)	qwen	0.223	0.389	0.377	−0.012
	granite	0.118	0.441	0.451	+0.010
	mistral	0.067	0.466	0.460	−0.006
	falcon3	0.110	0.445	0.442	−0.003
API-Bank (Li et al., 2023) (published)	qwen	0.187	0.407	0.383	−0.024
	granite	0.069	0.466	0.479	+0.013
	mistral	0.058	0.471	0.477	+0.006
	falcon3	0.068	0.466	0.473	+0.007
API-Bank (Li et al., 2023) (49 tools)	qwen	0.137	0.432	0.423	−0.009
	granite	0.085	0.457	0.465	+0.008
	mistral	0.051	0.474	0.459	−0.015
	falcon3	0.083	0.459	0.465	+0.006

C.2 CALIBRATION AND EVALUATION-BUDGET PLANNING

The admission identity is a property of exchangeability and the decision rule; it does not establish false-retention control. In the inert-edit AWM (Wang et al., 2026) experiment, strict-improvement retention accepts 43.3% of $1{,}440$ edits, and 43.6% of retained edits fail to replicate. For BFCL (Patil et al., 2025) and API-Bank (Li et al., 2023), holding the tool surface fixed and changing only random seeds makes the comparison inert by construction.

Figure 6 examines a separate issue: the calibration and power of candidate evaluation procedures. Both displayed t-test variants exceed the nominal null false-positive rate for three of four callers, whereas the permutation procedures remain calibrated. Pairing increases power under injected effects. These observations motivate matched evaluation; the exact guarantee for DISCORD follows from its conditional binomial test under the assumptions in Section 3.2.

For budget planning, let $p > 1/2$ be the probability that a discordant pair favors the candidate. Under the null, the standard deviation of the discordant win fraction is $1/(2\sqrt{n_d})$; under the alternative, it is

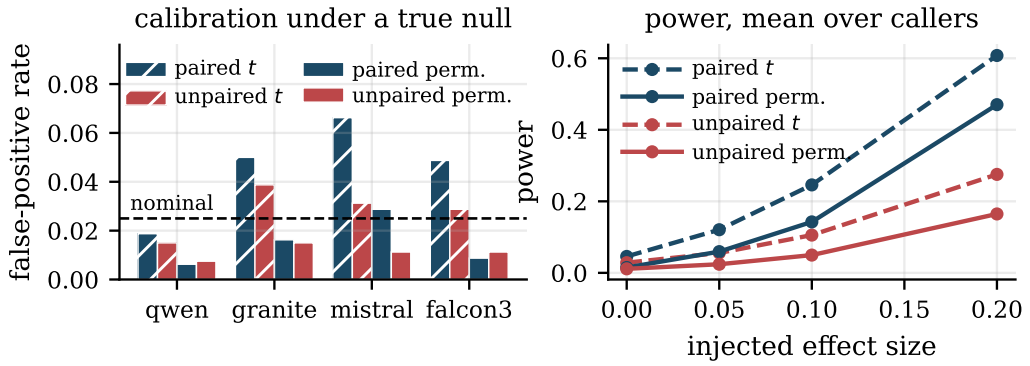


Figure 6: Null calibration and injected-effect power across four callers. The permutation procedures remain calibrated in the displayed diagnostic, and pairing increases power. These comparisons motivate paired evaluation; DISCORD's error control follows from its specified exact conditional test.

$\sqrt{p(1-p)/n_d}$. Equating the approximate rejection threshold and the desired power quantile gives

$$(p - 1/2)\sqrt{n_d} \approx z_{1-\alpha}/2 + z_{1-\beta}\sqrt{p(1-p)},$$

and hence the planning rule

$$n_d \approx \frac{\left[z_{1-\alpha}/2 + z_{1-\beta}\sqrt{p(1-p)}\right]^2}{(p-1/2)^2}, \qquad n_{\text{tasks}} \approx n_d/\pi_d, \tag{11}$$

where α is the one-sided test level, $1-\beta$ is target power, and z_u is the standard-normal quantile at probability u. The expected number of tasks needed to collect n_d discordances is approximately n_d/π_d, where π_d is the discordance probability. Rare discordance therefore increases evaluation cost even if the conditional effect is substantial. This normal approximation is used for planning; retention uses the exact tail probability.

C.3 EXACT CURTAILMENT

Fix the maximum number of evaluation pairs and the terminal significance threshold before inspecting outcomes. After each evaluation block, consider every feasible completion of the remaining pairs: a pair can add a candidate win, an incumbent win, or a concordance. Stop and accept only if every completion would be accepted by the terminal test; stop and reject only if every completion would be rejected. Otherwise, evaluate the next block.

For R remaining pairs, this condition can be expressed using

$$\mathcal{C}_R = \{(u, v) : u, v \in \mathbb{Z}_{\geq 0},\ u + v \leq R\},$$

where a completion yields counts $(b + u, c + v)$. Applying the terminal rule to all members of $\mathcal{C}_R$ defines the two stopping decisions exactly. When there are no discordances, the terminal rule rejects retention. For $n_d = b + c > 0$, the terminal one-sided tail is

$$p_{\text{tail}}(b, c) = 2^{-n_d} \sum_{j=b}^{n_d} \binom{n_d}{j}.$$

Retention requires $p_{\text{tail}} \leq \alpha$ for the prespecified level α; when $n_d = 0$, set $p_{\text{tail}} = 1$ and reject retention. The curtailed procedure (Alling, 1963) returns the same decision as full evaluation on every completed sequence and thus inherits its error control. It does not introduce an additional unadjusted test at each block. Savings depend on effect size and the remaining budget; ambiguous comparisons can require the full evaluation set.

C.4 CONTROLLED EVOLUTION LOOPS

The controlled loops use the BFCL v3 (Patil et al., 2025) `live_multiple` split, restricted to 200 single-turn tasks with binary syntax-tree matching. Callers are Qwen3-VL-8B-Instruct (Bai et al., 2025), Granite-3.3-8B-Instruct (IBM Granite Team, 2025), Mistral-7B-Instruct-v0.3 (Mistral AI, 2024), and Falcon3-7B-Instruct (Technology Innovation Institute, 2024), sampled at temperature 0.7 with eight seeds per task unless otherwise specified in the run records.

Each loop evaluates eight rounds of three proposed edits. Edits add or remove blocks of advertised function names. Withholding names at initialization creates a recoverable handicap; starting from the full surface instead produces a regime in which proposals are predominantly harmful. The three retention rules consume matched episode budgets and identical proposals. The untested rule retains a strict empirical improvement; the single-round rule compares disjoint task halves with a label-permutation test; DISCORD uses paired outcomes and its exact conditional test.

DISCORD recovers more of the injected handicap in 18 of 21 paired comparisons (Figure 3). The one-sided paired t-test gives $p = 0.00015$ ($t = 4.37$, 20 degrees of freedom); enumerating all 2^{21} sign assignments gives an exact sign-flip value of 8.5×10^{-5} under the corresponding symmetry null. The single-round gate overstates realized effects by 1.8–17×. In the harmful-proposal regime, DISCORD preserves performance in all 17 matched seeds, whereas strict-improvement retention degrades in 9 of 17. The paired contingency table has nine seeds with degradation only under strict-improvement retention, none with degradation only under DISCORD, and eight with no degradation under either method. The two-sided exact McNemar value is $2(1/2)^9 = 0.00390625$. Mean changes are small, so this result concerns the frequency of degradation rather than a large difference in mean improvement.

C.5 LIVE MCP COMPARISON

The live comparison uses 320 AWM (Wang et al., 2026) tasks across 158 servers, with a shared proposal stream and a full-surface reference objective of 0.2813. The shallow regime withholds 10% of tool names, producing a recoverable effect of 2.2 pp; the deep regime withholds 40%, producing 12.7 pp. Table 6 reports final objective values, acceptance counts, and episode cost.

The deployed rule reaches a higher final objective with fewer evaluation episodes but overstates it by 32% and 47% in the two regimes. DISCORD provides a test-based retention decision at greater evaluation cost. This comparison separates the reliability of reported improvement from the terminal objective achieved by a search; the method does not dominate both criteria.

Table 6: Live AWM (Wang et al., 2026) retention comparison. Parenthesized final-objective percentages are relative to the full-surface reference. Self-reported overstatement is relative to the measured final objective; a dash indicates that no numerical self-report is tabulated. Episode counts include evaluation cost.

regime	rule	accepts (good/bad)	true final	self-reported	episodes
shallow	deployed	3 (2/1)	0.2656 (94%)	0.3500 (+32%)	1,760
shallow	DISCORD	0 (0/0)	0.2594 (92%)	–	1,906
deep	deployed	3 (3/0)	0.2812 (100%)	0.4125 (+47%)	1,920
deep	DISCORD	2 (2/0)	0.2703 (96%)	–	4,008

D BRACE: POSTERIOR COMPUTATION AND IMPLEMENTATION

D.1 POSTERIOR EVIDENCE AND SAMPLING

A task's success and failure evidence defines the working Beta approximation $p_i \sim \mathrm{Beta}(a_i, b_i)$ with $a_i = 1 + s_i$ and $b_i = 1 + f_i$. At initialization, the implementation caps historical evidence separately at 20 successful and 40 failed episodes per task. The bank is incorporated once; subsequent cycles add local outcomes and apply the same evidence-discount rule. The registered configuration uses $\gamma = 1$ and retains evidence without decay. Setting $\gamma < 1$ would reduce the influence of outcomes from older policies.

A cycle selects 256 tasks, corresponding to eight optimizer steps of 32 prompts each, with $k = 5$ rollouts per prompt. Weighted sampling uses $w_i^{1/\tau}$ with $\tau = 0.5$ at all three scales. The exploration rate is $\epsilon = 0.05$ at 2B and 4B, and 0.10 at 8B. The implementation reserves $n_\epsilon = \mathrm{round}(\epsilon N)$ of the $N = 256$ selected rows for uniform exploration and allocates the rest by weight. The uniform exploration component samples from eligible tasks not already selected, as in Algorithm 1. Thus each selected task contributes one group in the cycle. Task indices, selected groups, and predicted degeneracy are recorded before the groups are generated.

A plug-in estimator substitutes a single estimated solve probability into the availability score. Posterior integration averages over uncertainty in that probability. Two tasks with the same posterior mean can therefore receive different scores when their evidence counts differ. This is especially relevant early in training, before the current run has collected substantial task-specific evidence.

D.2 Bank provenance and cost accounting

The default bank reads other runs' episode logs once at initialization and excludes the current arm's own log, which supplies its live evidence. Historical outcomes come from policies at 2B, 4B, and 8B. Thus initialization can transfer evidence about task difficulty across model scales, including from larger to smaller models. The cumulative audit inventory contains 195 episode files: 107 from 8B, 47 from 2B, and 41 from 4B policies. This inventory differs from an individual run's initialization snapshot, which depends on the files available at launch. Only eligible training-task outcomes contribute to initialization.

The VIP-score (Nguyen et al., 2026) and correlation-removal ablations use the same capped per-task starting counts as full BRACE. Their bank-read totals can differ because the available files and redundant observations differ, while capping produces the same retained evidence. Consequently, more parsed episodes do not imply a larger effective prior in these comparisons.

The historical corpus and the capped evidence have different sizes. For example, the recorded bank read for the 4B arm `q4bT2` parses 651,804 episodes from 115 files and primes 1,023 tasks. Separate caps of 20 successes and 40 failures bound its historical contribution by 61,380 counts, or 9.4% of the parsed episode count. This is an upper bound on retained historical counts; it is neither the fraction of eligible episodes retained nor a compute-saving estimate.

The bank reuses episodes generated by other experimental runs; no dedicated collection run was launched to build it. This avoids additional probing for the receiving run while retaining the historical collection cost and the benefit of additional data. A current allocation cycle requests $256 \times 5 = 1{,}280$ episodes. Historical generation, current-run generation, and the one-time bank read are separate costs.

D.3 Correlation correction and exact Beta moments

For a group on task i, draw a latent probability $\theta_i \sim \mathrm{Beta}(\nu p_i, \nu(1 - p_i))$ and draw its k binary rewards independently conditional on θ_i. Marginalizing θ_i induces within-group correlation $\rho = 1/(\nu + 1)$. Uncertainty about this group-level probability is distinct from posterior uncertainty about the task mean p_i.

Write the unanimous-success probability as the degree-k polynomial

$$q(p) = \prod_{j=0}^{k-1} \frac{\nu p + j}{\nu + j} = \sum_{m=0}^{k} c_m p^m.$$

Using rising factorials $(x)^{\overline{m}} = \prod_{j=0}^{m-1}(x + j)$, with $(x)^{\overline{0}} = 1$, the score can be computed without Monte Carlo integration:

$$w_i = 1 - \sum_{m=0}^{k} c_m \frac{(a_i)^{\overline{m}} + (b_i)^{\overline{m}}}{(a_i + b_i)^{\overline{m}}}.$$

D.4 Expected informative-group yield

Fix the policy and the evidence available at allocation time. Let I_i indicate that a new group on task i contains both reward values. By definition, $w_i = \mathbb{E}[I_i]$ under the predictive model. If one equal-cost group is generated for each task in a selected set S, linearity gives

$$\mathbb{E}\left[\sum_{i \in S} I_i \mid S\right] = \sum_{i \in S} w_i.$$

This holds without assuming independence across selected tasks. Under known scores, selecting the N largest w_i maximizes the model's expected count of informative groups among fixed-size sets. BRACE uses estimated scores, weighted sampling, and uniform exploration instead of this deterministic oracle. The identity interprets its allocation target; it does not establish that the resulting gradient steps improve held-out solve rate.

D.5 Eligibility and verifier checks

Exclusion affects training allocation only. The 104 excluded tasks comprise 66 initially satisfied verifiers, 33 objectives ruled out by an initial-state guard proof, and 5 tasks identified by a string-matching heuristic. The final category is not a proof-based certificate. The guard does not fire on any of the 826 bank tasks solved at least once. Evaluation and validation pools remain unchanged.

A verifier that already accepts the initial state does not establish meaningful task completion by a new trajectory. Conversely, repeated observed failures do not prove that success is impossible. Structural checks and posterior evidence therefore serve different roles: the checks determine eligibility, and the posterior ranks and weights eligible tasks. The non-degeneracy objective in Section 4.1 motivates weighting both consistently successful and consistently unsuccessful tasks below tasks likely to produce mixed rewards.

E Calibration, rank preservation, and allocation controls

E.1 Prospective forecast evaluation

For each selected batch, the predicted degenerate fraction is the average of $1 - w_i$ over selected tasks. The observed fraction counts groups whose verifier rewards are unanimous. Predictions are recorded before generation, using the shared correlation estimate $\hat{\rho} = 0.78$ specified in Section 4.2. The 83 evaluated arm-cycles exclude both cycles used to fit ρ; the parameter is held fixed during this forecast evaluation. This separation is distinct from excluding held-out AWM (Wang et al., 2026) tasks from bank initialization. Mean absolute forecast error is 0.591 under independence versus 0.056 with the correlated model.

Figure 7 retains the diagnostic underlying this comparison. It compares the two likelihood assumptions on the same cycles and the same per-task estimates. Its interpretation is prospective batch calibration; it does not by itself establish improvement in task ordering or downstream policy quality. Those questions require the rank analysis and the held-out comparisons, respectively.

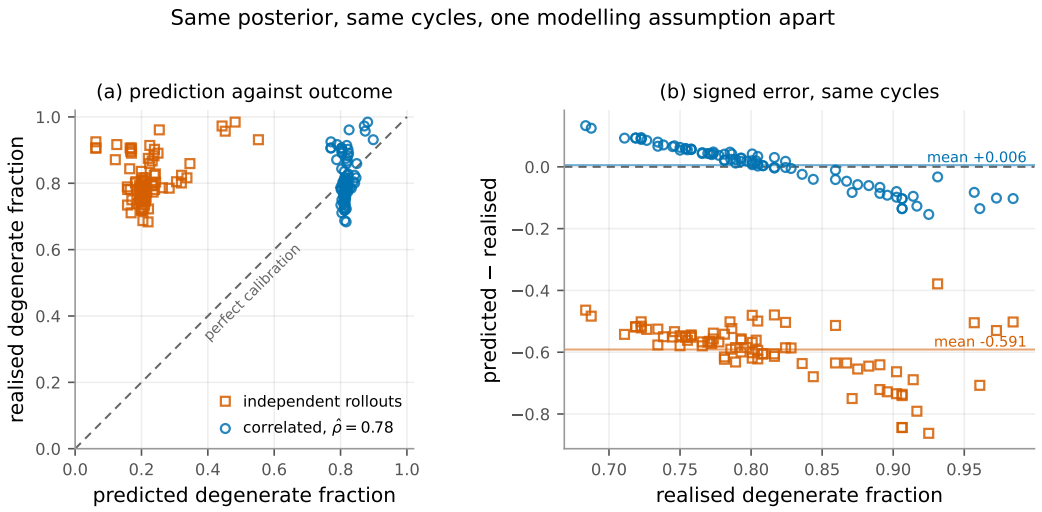


Figure 7: Predicted and observed group degeneracy in the recorded 8B calibration diagnostic. Left: predicted versus observed fractions. Right: signed forecast errors. Mean absolute errors are 0.591 and 0.056 for the independent and correlated models, respectively; the signed mean errors annotated in the plot are -0.591 and $+0.006$. The diagnostic contains 83 arm-cycles from 21 arms.

E.2 Proof of rank preservation

For the independent score, differentiation gives

$$\frac{\partial}{\partial p}[p^k + (1-p)^k] = k[p^{k-1} - (1-p)^{k-1}] < 0, \qquad 0 < p < 1/2.$$

Both availability and $u = p(1-p)$ are symmetric about $p = 1/2$, and both increase strictly on $[0, 1/2]$. Thus independent availability is strictly increasing in u. For the correlated case, use the function f defined in Proposition 1.

Proof of Proposition 1. f is a product of positive increasing linear functions of p, so as a polynomial $f(p) = \sum_m c_m p^m$ it has $c_m \geq 0$ with $c_0 = 0$ and $c_k = \nu^k / \prod_{j<k}(\nu + j) > 0$. Then $f''(p) = \sum_{m \geq 2} m(m-1)c_m p^{m-2} > 0$ on $(0, 1]$ because $c_k > 0$ and $k \geq 2$, so f' is strictly increasing and $g_c'(p) = f'(1-p) - f'(p) > 0$ for $p < \frac{1}{2}$. As $g_c(p) = g_c(1-p)$ by construction and $u \mapsto p = (1 - \sqrt{1-4u})/2$ is a strictly increasing bijection $[0, \frac{1}{4}] \to [0, \frac{1}{2}]$, the composition is strictly increasing. At the endpoint $p = \frac{1}{2}$ the derivative in u is $0/0$ with limit $f''(\frac{1}{2})$, which is positive for all $\nu > 0$, $k \geq 2$ by the same coefficient argument. □