RESOLVABLE-EFFECT ACCOUNTING FOR TOOL-CALLING AGENTS: REWARD BIAS AND EVALUATION RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Small gains from training tool-calling language models are difficult to assess when the training signal is affected by filtering and evaluation outcomes vary across runs. We introduce Resolvable-Effect Accounting (REA), a framework for examining reward logging and evaluation resolution, with candidate coverage and gradient statistics as supporting diagnostics. We show that, under group filtering in Group Relative Policy Optimization (GRPO), the retained-group reward is a conditional mean that need not move with the underlying success rate; logging the unfiltered mean avoids this selection effect for a fixed prompt distribution. In twelve-run HotpotQA evaluation panels, an estimated 69% of single-run variance is associated with run-level variation under the tested seed protocol, and same-seed pairing reduces comparison uncertainty when outcomes covary across conditions. In a Qwen3.5-2B case study, the observed checkpoint differences after 240 GRPO updates and 20 self-distillation updates are $+0.098$ and $+0.032$ percentage points, respectively, with 95% intervals that include zero. These results motivate explicit reporting of reward-selection rules and evaluation uncertainty. REA quantities can inform evaluation planning before optimizer updates, while their prospective value for training decisions remains to be assessed across matched configurations and additional settings.

1 INTRODUCTION

Reinforcement learning and on-policy distillation have enabled language models to improve reasoning and tool use through interaction and feedback (Shao et al., 2024; Jin et al., 2025; Chen et al., 2025; Zhao et al., 2026). For researchers working with limited computational resources, however, assessing the value of a training run requires more than observing a favorable optimization curve. The reported progress signal must reflect the intended objective, and the evaluation protocol must distinguish plausible improvements from stochastic variation. These requirements become particularly important when training effects are small: an apparent increase in reward may arise from changes in how trajectories are selected, while a small difference in task accuracy may remain uncertain even on a large evaluation set.

Two measurement problems complicate this assessment. First, the interpretation of a logged reward depends on which samples contribute to it. In implementations of Group Relative Policy Optimization (GRPO) that discard groups with zero reward variance, averaging rewards only over retained groups produces a selection-conditioned statistic whose value depends both on the policy's success probabilities and on which groups survive filtering. The logged mean can therefore decrease even when the underlying success rate increases (Figure 2(b)), and filtered and unfiltered readings diverge during the same training run (Figure 2(c)). Second, evaluation uncertainty may include variation shared across questions within a run. In our panels, a substantial fraction of the observed variance is attributable to a run-level component (Figure 3(b)), so increasing the number of questions alone leaves a nonzero uncertainty floor for unpaired comparisons at a fixed number of repeated runs (Figure 3(a)). Pairing evaluations by generation seed can reduce comparison variance when fluctuations are shared across conditions (Figure 3(c)), although complete cancellation requires additional assumptions.

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.

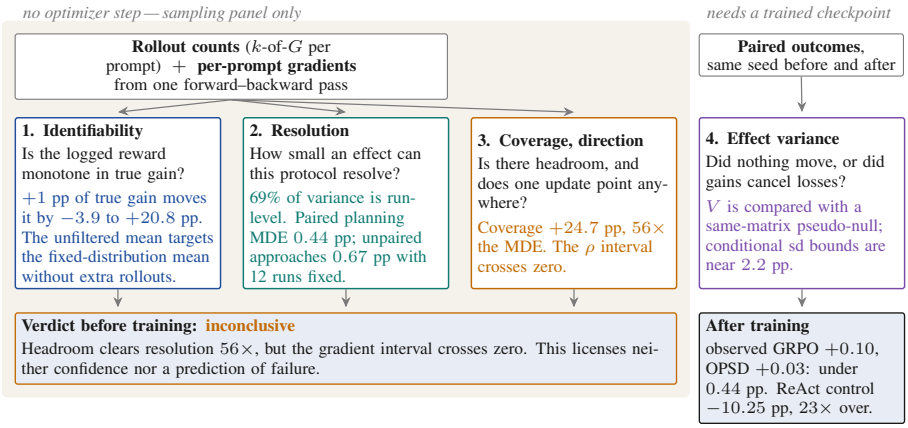

Figure 1: REA connects four questions to their required evidence. Reward logging and evaluation resolution are the core measurement analyses; candidate coverage and gradient signal provide supporting evidence; per-question effects require a fixed trained checkpoint. The displayed quantities come from the case study below.

We introduce **Resolvable-Effect Accounting (REA)** to organize this audit. Its two central components are (i) the relationship between filtered and unfiltered reward summaries and (ii) the resolution of paired and unpaired evaluation designs under estimated question- and run-level variation. Candidate coverage and gradient signal provide supporting information about sampled outcomes and optimization gradients, while post-training checkpoint comparisons describe observed task differences. Several REA quantities can be computed from rollouts or gradient measurements without an optimizer update and can therefore inform evaluation planning; they do not determine whether an optimizer will improve a target metric.

Our study examines these quantities for Qwen3.5-2B on HotpotQA (Yang et al., 2018) with BM25 retrieval. For groups of $G = 16$ rollouts, the retained-only reward estimates $\mathbb{E}[k/G \mid 0 < k < G]$, whereas the unfiltered mean estimates $\mathbb{E}[k/G]$. Under four constructed changes to a fitted success-probability distribution, each matched to a $+1.00$ percentage-point change in its mean, the retained statistic changes by -3.93, $+0.52$, -2.61, or $+20.77$ pp (Figure 2). These examples illustrate dependence on the distributional shift; they are not a universal calibration.

In the reported checkpoints, the estimated EM differences after 240 GRPO updates and 20 On-Policy Self-Distillation (OPSD) updates are $+0.098$ pp (95% CI $[-0.123, +0.318]$) and $+0.032$ pp (95% CI $[-0.221, +0.284]$), respectively. Both intervals include zero. Because there are no independent training repetitions and the evaluation manifests are not fully configuration-identical, we report these as observed checkpoint-panel differences rather than effects attributable solely to training. This study characterizes how REA quantities behave in one model–task setting; whether using them prospectively improves training decisions requires further evaluation.

2 Resolvable-Effect Accounting

REA separates two measurement questions from two supporting analyses. Reward logging and evaluation resolution describe whether available signals track success and whether a specified design can distinguish a change. Candidate coverage and gradient signal add information about sampled outcomes and gradients but do not determine optimizer gains. Post-training checkpoint comparisons provide a separate description of observed task differences; they are not part of the training-before quantities. Several REA quantities are available before an optimizer update and provide training-before evidence about reward-log validity and evaluation resolution. We use fixed checkpoints to characterize these quantities in one setting; whether they improve allocation decisions across configurations remains open. Our agents interact with a retrieval tool over multiple turns, and a trajectory succeeds when its final answer matches the gold answer under exact match (EM). We use $F(q)$

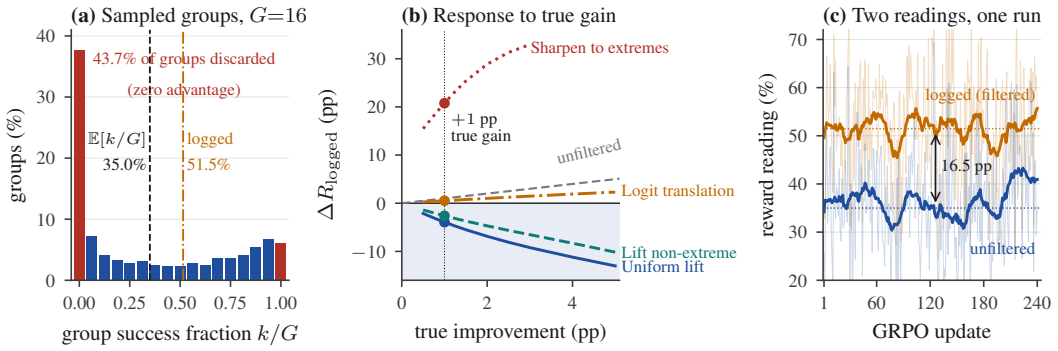

Figure 2: Filtered reward logging fails as a progress signal. **(a)** GRPO discards 43.74% of groups (all-correct or all-wrong); the logged mean over retained groups sits 16.49 pp above the unfiltered mean. **(b)** Under four policy-shift families calibrated to deliver +1.00 pp true improvement, the filtered log responds with -3.93 to $+20.77$ pp—a forty-fold range that changes sign. Only the unfiltered mean tracks the diagonal. **(c)** The divergence in the actual run: faint lines are per-update values, heavy lines a 21-update moving average.

for the distribution of per-question success probabilities; finite rollout counts identify only certain functionals of F, while deconvolution-based extrapolations remain model-conditional.

2.1 Core Diagnostic 1: Filtered Reward Logging Introduces Selection Bias

For $K \mid q \sim \text{Binomial}(G, q)$, the filtered statistic has the explicit form

$$R_{\text{logged}}(F) = \frac{\mathbb{E}_F[q - q^G]}{\mathbb{E}_F[1 - q^G - (1-q)^G]}, \qquad R_{\text{all}}(F) = \mathbb{E}_F[q], \tag{1}$$

provided the denominator of the first expression is positive. Both terms in the ratio depend on F. When the policy improves and F shifts, conditioning on survival introduces selection bias (Elwert & Winship, 2014); whether the conditional mean rises or falls depends on the geometry of the shift, not only on the sign of the change in R_{all}. To make this concrete we construct four families of probability-level shifts, solve within each for the scale that produces a given amount of true improvement, and recompute R_{logged}. For $q \in [0,1]$, additive lift is $T_\epsilon^{\text{add}}(q) = \min\{1, q+\epsilon\}$; logit translation is $T_\lambda^{\text{logit}}(q) = \text{sigmoid}(\text{logit}(q) + \lambda)$ for $q \in (0,1)$ and leaves the endpoints fixed; interior-only lift applies the same clipped additive map only to $q \in (0,1)$; and sharpening is $T_\eta^{\text{sharp}}(q) = \text{clip}(q + 2\eta(2q-1), 0, 1)$. These are shifts of success probabilities, not token logits. Figure 2(b) plots the responses at $G = 16$. At $+1.00$ pp of true improvement the same gain yields -3.93, $+0.52$, -2.61 and $+20.77$ pp, a forty-fold range that changes sign. These values are plug-in constructed responses under the fitted mixture and shift families; we do not propagate fitting uncertainty or present them as a universal calibration, and their ordering holds across the range we can solve.

The correction costs nothing: log $\mathbb{E}[k/G]$ over all sampled groups before the filter. This is an unbiased estimate of the mean success rate under the fixed target prompt distribution. It does not substitute for a held-out test set when the prompt distribution changes. In our GRPO run the retained-only reading is 51.48% and the unfiltered reading is 35.00%, a 16.49 pp selection gap.

2.2 Core Diagnostic 2: Run-Level Variance Imposes a Resolution Floor

For condition $a \in \{0, 1\}$ (for example, base and trained), write the binary outcome for question i in run j as $Y_{ij}^{(a)} = q_i^{(a)} + \delta_j^{(a)} + \epsilon_{ij}^{(a)}$. Here $\delta_j^{(a)}$ is a run-level offset and $\epsilon_{ij}^{(a)}$ is question-level sampling noise. A single run's mean has variance approximately $\mathbb{E}[q_i^{(a)}(1 - q_i^{(a)})]/N + \text{Var}(\delta_j^{(a)})$. For a same-seed paired difference, the run-level contribution is instead

$$\text{Var}(\delta_j^{(1)} - \delta_j^{(0)}) = \sigma_1^2 + \sigma_0^2 - 2\,\text{Cov}(\delta_j^{(1)}, \delta_j^{(0)}). \tag{2}$$

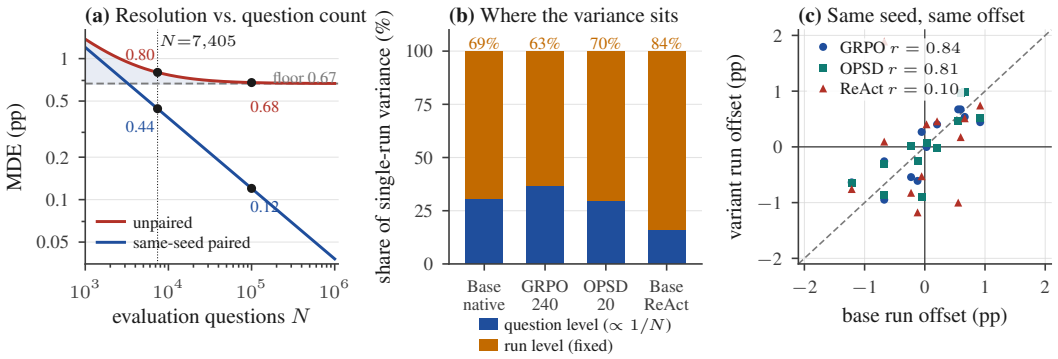

Figure 3: Run-level variance imposes a resolution floor that more questions cannot overcome. **(a)** Unpaired MDE flattens against 0.67 pp when the run count stays fixed at twelve; paired comparison under same-seed coupling lowers the run-level component when offsets correlate. **(b)** The run-level share of single-run variance across four evaluation panels. **(c)** Per-seed run means as deviations from panel averages: variants sharing weight ancestry correlate strongly ($r = 0.84$ base-to-GRPO), protocol changes do not ($r = 0.10$ base-to-ReAct).

Thus pairing reduces this component when the offsets are positively correlated; using the same seed alone does not prove that the residual is zero.

For Qwen3.5-2B on HotpotQA with native tool_calls and generation seeds 42 through 53, the estimated single-run sd is 0.6355 pp, of which the estimated run-level component contributes 0.5295 pp—69.4% of the variance, leaving 30.6% at the question level. Figure 3(b) shows the split holds across all four panels, between 63% and 84%. That component correlates strongly across variants sharing weight ancestry ($r = 0.84$ base-to-GRPO, $r = 0.81$ base-to-OPSD) and weakly across protocols ($r = 0.10$ base-to-ReAct), which Figure 3(c) shows run by run. These correlations are consistent with shared run-level variation, but they do not identify its cause or establish a causal seed effect.

The design consequences diverge sharply (Figure 3(a)). Unpaired comparison carries the run-level component and therefore approaches a floor when the number of independent runs is held at twelve: at 80% power and $\alpha = 0.05$, unpaired MDE is 0.80 pp at our 7,405 questions and improves only to 0.68 pp at 100,000, against an asymptotic floor of 0.67 pp. A paired projection under the common-offset approximation falls as $1/\sqrt{N}$: 0.44 pp at 7,405 questions and 0.12 pp at 100,000. In the observed panels, pairing lowers rather than proves elimination of run-level variation. These are planning projections conditional on the estimated components, fixed run count, and stable protocol; the MDE is a power-planning quantity, not an equivalence threshold.

The component's behaviour is consistent with our generation-seed mechanism, in which every request in a run receives the same seed value. Our twelve runs confound seed value with process identity, so we can state the component lives at the run level and is compatible with a shared-seed mechanism, but not that seeds cause it. A seed ablation varying seed and process independently would be needed for causal identification.

2.3 Supporting Evidence: Candidate Coverage and Gradient Signal

Beyond the two core diagnostics, two quantities provide supporting but less definitive evidence. **Candidate coverage** upper-bounds reranking gains within observed rollouts: with $n = 12$ rollouts per question, pass@1 is 35.26% and pass@12 is 59.97%, a 24.71 pp increment that bounds perfect reranking in this panel. This is a descriptive upper bound, not a forecast of training gains. **Gradient signal-to-noise** quantifies directional consistency: following McCandlish et al. (2018), we compute $\rho = \|\mu\|^2 / \operatorname{tr}(\Sigma)$ from per-prompt gradients, where μ and Σ are the mean and covariance. An exploratory 64-prompt panel gives $\hat{\rho} = 0.001298$ with jackknife interval $[-0.000655, 0.003250]$, which includes zero. The reciprocal $1/\hat{\rho} \approx 770$ is an illustrative scale, not a reliable critical-batch

recommendation, and this panel used a different configuration ($\beta_{\text{KL}} = 0$, strict tool definitions) than the trained run.

Beyond the observed depth, k-of-n counts identify only the first n moments of F (Teicher, 1963; Singh & Singh, 2026). For $G = 16$, moment bounds give pass@16 $\in [59.97\%, 62.09\%]$; binomial deconvolution by EM (Laird, 1978) yields 61.63% as a model-conditional extrapolation. The coverage increment is 56 times the base-panel paired MDE proxy—a descriptive comparison, not a statistical guarantee. These summaries do not model the run-shared offset visible in our panels; detailed bounds and identifiability conditions appear in Appendix A.

Taken together, the supporting evidence is *inconclusive*: coverage clears the resolution proxy while gradient signal stays unresolved. This licenses neither confidence nor a prediction of failure; the two core diagnostics carry the main measurement interpretation.

2.4 Post-training Checkpoint and Per-question Analysis

The fourth analysis is deliberately separate from quantities available before training. For a trained checkpoint and a base checkpoint, let $\Delta_i = q_i^{\text{trained}} - q_i^{\text{base}}$ denote the latent change for question i. We report the paired mean difference over evaluation runs and a split-sample estimate of $\text{Var}_i(\Delta_i)$ after subtracting the condition-specific sampling variances (Appendix A). This decomposition asks whether a small aggregate difference could be accompanied by opposing question-level changes. It is conditional on the observed twelve runs, uses a base-native pseudo-null for estimator behavior, and does not identify a population effect variance or prove equivalence.

3 Experimental Design and Setup

We evaluate REA through four research questions:

- **RQ1 (reward validity):** How does filtering zero-variance groups change the relationship between logged reward and underlying success probability?
- **RQ2 (evaluation resolution):** How do question-level and run-level variation affect comparison uncertainty, and what does same-seed pairing change?
- **RQ3 (diagnostic information):** What do candidate coverage and gradient measurements reveal about available outcomes and optimization-signal uncertainty?
- **RQ4 (training outcomes):** How should average and per-question checkpoint differences be interpreted alongside optimization traces and controls?

Each training procedure was run once, and repeated evaluation panels were collected after the checkpoints were produced. The RQs therefore assess what the quantities reveal in this case; broader decision utility across settings remains open. Training and evaluation runs are distinct: twelve generation runs repeat a fixed checkpoint, whereas no independent training seeds are available.

3.1 Task, Model, and Retrieval Environment

We use Qwen3.5-2B (Qwen Team, 2026), a 2B-parameter dense model adapted with LoRA (Hu et al., 2022) at rank 16, $\alpha = 32$, and zero dropout. Adapters target the attention and feed-forward linear projections, leaving 16.82M trainable parameters while freezing the base checkpoint. Training uses 15,661 records from the hard HotpotQA training split; evaluation uses all 7,405 questions in the distractor validation split. The primary metric is exact match (EM), with token-level Answer F1 as a secondary metric.

Retrieval uses BM25 over a frozen Wiki-18 Lucene index through Pyserini. Each retrieval call returns the top three passages, truncated to 1,200 characters. The audited `query_only` configuration passes the question text directly as the retrieval query. Corpus, index, question-set, scorer, and configuration fingerprints are recorded in run manifests.

3.2 Training Conditions

GRPO runs 240 updates with 8 prompts per update, $G = 16$, $\beta_{\text{KL}} = 0.04$, clipping $\epsilon = 0.2$, and AdamW at 2×10^{-5} with 5 warmup steps. Rollouts are strictly on-policy: temperature 1.0, top-p 1.0, no truncation, and BM25 returns the top three passages. Of 3,413 sampled groups, 1,493 (43.74%) have zero variance and are discarded; the run consumed 44.06 GPU hours over 54,608 trajectories. OPSD runs 20 updates with micro-batch 4 and a 131,072-token budget per update, minimizing generalized Jensen–Shannon divergence at $\beta = 0.5$ with per-token vocabulary contributions clipped at 0.05; the teacher sees privileged context (reference answer and supporting sentences), whereas the student does not. These are separate audited case studies, not a compute-matched comparison.

3.3 Evaluation Protocol and Statistical Units

Each evaluation run covers the same 7,405 questions with one generation seed held fixed across requests; twelve runs use seeds 42–53. Conditions are paired by question identifier and seed for checkpoint differences. Evaluation uses temperature 1.0, top-p 0.95, top-k 20, and presence penalty 1.5. Agents may make at most five tool calls and eight model calls in a 32,768-token context. We report EM and Answer F1 averaged over runs; 95% intervals for paired EM differences are paired t intervals over the twelve seed-level differences. They condition on the fixed checkpoints and do not include uncertainty from independent training runs. The variance decomposition and MDE projections are conditional on this protocol and fixed run count.

The gradient panel contains 64 prompts with $G = 16$ rollouts and one forward–backward pass per prompt. It uses $\beta_{\text{KL}} = 0$ and stricter tool definitions than the GRPO training run, so it is exploratory rather than a matched training diagnostic. A protocol control evaluates the base checkpoint with ReAct-style text markers (Yao et al., 2023) instead of native structured tool calls. This is a comparison of the complete protocol change, not an isolated causal test of serialization. Detailed configurations appear in Appendix B.

3.4 Analyses Mapped to the RQs

RQ1. We reconstruct filtered and unfiltered reward summaries from the same GRPO rollout records and evaluate four constructed shifts of a fitted success-probability distribution. Each shift is calibrated to the same change in its mean before its filtered response is computed; these transformations are illustrative, not observed training trajectories.

RQ2. We estimate question- and run-level components from the twelve-run panels, report cross-condition correlations, and project MDEs as the question count changes while the run count is fixed. The paired projection uses an explicit covariance assumption; same-seed pairing is not treated as automatic variance cancellation.

RQ3. Candidate coverage is measured from twelve rollouts per question. We report identified pass@k at the observed depth, moment bounds and model-conditional extrapolations beyond it, and an exploratory gradient signal-to-noise estimate with a prompt-level jackknife interval. A six-run/six-run voting control checks answer-distribution prediction at a larger effect scale.

RQ4. We compare observed checkpoint-panel differences with the base-native panel, inspect GRPO and OPSD optimization traces, and report a conditional split-sample per-question effect-variance analysis. The same-model pseudo-null reuses the base-native outcome matrix on both sides; it is an estimator check, not an independent true-zero training control. All analyses condition on the observed evaluation runs.

4 Results

We answer the four RQs in the same order as the experimental design. Each subsection states the comparison, reports the evidence, and then gives a scope-limited answer.

4.1 RQ1: How Does Group Filtering Affect Reward Interpretation?

We first compare two summaries of the same GRPO rollout records, so their difference is not caused by collecting different trajectories. **Filtering changes the estimand substantially.** Of the sampled groups, 43.74% have zero reward variance and are discarded. The retained-group reward is 51.48%, whereas the unfiltered mean is 35.00%, a 16.49 pp gap (Figure 2a). The former is a conditional mean over groups satisfying $0 < k < G$; it should not be interpreted as a direct measure of the unconditional success rate.

The filtered statistic can move in the opposite direction from the underlying rate. Under the four constructed shifts in Figure 2b, the same $+1.00$ pp increase in the fitted mean success probability produces filtered-reward changes of -3.93, $+0.52$, -2.61, and $+20.77$ pp. These responses depend on the fitted distribution and specified transformations; they demonstrate non-monotonicity of the statistic rather than a universal calibration.

The two readings also diverge during the observed training run. Across four 60-update segments, retained rewards are 51.69, 51.22, 52.34, and 50.68%, while unfiltered means are 36.21, 33.59, 33.21, and 37.35%. The weighted trend of the unfiltered reading is $+1.96$ pp with bootstrap interval $[-2.60, +6.42]$ pp. **Answer to RQ1:** reward logging after group filtering changes the quantity being tracked and can distort progress interpretation. Logging the unfiltered mean before filtering makes this selection effect visible without additional rollouts, provided the target prompt distribution is fixed.

4.2 RQ2: What Limits Evaluation Resolution?

We next use the repeated panels to separate question-level sampling variation from variation across evaluation runs. **Run-level variation is substantial under the tested protocol.** For the base-native condition, the estimated single-run standard deviation is 0.6355 pp and the estimated run-level component is 0.5295 pp, corresponding to 69.4% of the variance. Across the four panels, the estimated share ranges from 63% to 84%; these are conditional estimates under the stated decomposition, not universal constants.

Pairing helps when run-level fluctuations covary, but it does not guarantee cancellation. Per-seed run means correlate at $r = 0.84$ for base–GRPO and $r = 0.81$ for base–OPSD, compared with $r = 0.10$ for base–ReAct. The paired run-level term is $\sigma_1^2 + \sigma_0^2 - 2\,\mathrm{Cov}(\delta^{(1)}, \delta^{(0)})$; hence the benefit depends on the covariance, not on the seed label alone. Same-seed comparisons reduce the observed outcome-flip rates from 18.28% to 14.79% for base–GRPO and from 18.28% to 14.61% for base–OPSD, but the residual combines evaluation noise and model differences.

Question count and run count address different sources of uncertainty. With twelve runs fixed, the unpaired MDE is 0.80 pp at 7,405 questions, 0.68 pp at 100,000 questions, and approaches a 0.67 pp floor. A paired projection under the common-offset planning approximation is 0.44 pp at 7,405 questions and 0.12 pp at 100,000. These are conditional power-planning quantities, not equivalence thresholds. **Answer to RQ2:** additional questions reduce the question-level component, while additional independent runs or covariance-aware pairing are needed to address run-level variation. The seed/process confounding in the twelve runs prevents a causal attribution to seeds.

4.3 RQ3: What Information Do Coverage and Gradient Diagnostics Provide?

Candidate coverage and gradient signal describe different parts of the training problem. **The rollout panel contains a sizeable candidate-selection opportunity.** With twelve sampled trajectories per question, pass@1 is 35.26% and pass@12 is 59.97%, a 24.71 pp difference. These are conditional plug-in summaries under the stated iid-within-question model; the repeated-run offset is not modeled in them. The difference is the gain available to an oracle selector within the observed candidates, not a forecast or upper bound on what training will achieve. At depth 16, moment constraints give pass@16 in $[59.97\%, 62.09\%]$ and the fitted-mixture estimate is 61.63%; the latter is model-conditional extrapolation.

Gradient evidence remains unresolved and configuration-specific. The exploratory 64-prompt gradient panel gives $\hat{\rho} = 0.001298$ with jackknife interval $[-0.000655, 0.003250]$, which includes zero. Its $\beta_{\mathrm{KL}} = 0$ coefficient and stricter tool definitions differ from the trained GRPO run, so

$1/\hat{\rho} \approx 770$ is not a reliable batch-size recommendation. A six-run/six-run voting control predicts a $+3.18$ pp gain at five votes against $+3.08$ pp observed; this checks the prediction procedure at a larger effect scale, not sub-percentage-point sensitivity.

Answer to RQ3: coverage identifies available successful candidates, while the gradient panel does not provide a resolved directional signal for the audited training configuration. Together these diagnostics are inconclusive about training gains; they should be reported with their sampling depth and configuration.

4.4 RQ4: How Should the Training Outcomes and Controls Be Interpreted?

Table 1 summarizes the twelve-run checkpoint panels. **Neither trained checkpoint shows a statistically detectable EM improvement in this evaluation.** The observed differences relative to base native are $+0.098$ pp for GRPO (95% CI $[-0.123, +0.318]$) and $+0.032$ pp for OPSD (95% CI $[-0.221, +0.284]$). Both intervals include zero, and both point estimates are below the 0.44 pp base-panel planning proxy. Because the panels have manifest differences and there are no independent training repetitions, these are checkpoint-panel differences rather than configuration-isolated causal training effects.

Table 1: HotpotQA results (7,405 questions, avg@12 over seeds 42–53). Paired Δ is relative to base native; the question-set hash is identical across runs.

Run	EM (%)	Answer F1 (%)	Paired ΔEM (pp)
Base native	35.26	46.85	—
Base ReAct	25.01	34.05	-10.25 $[-10.91, -9.59]$
GRPO 240 updates	35.36	46.89	$+0.098$ $[-0.123, +0.318]$
OPSD 20 updates	35.29	46.78	$+0.032$ $[-0.221, +0.284]$

Question-level stability is consistent with this interpretation. Re-evaluating base under different seeds flips 18.28% of outcomes; switching to GRPO or OPSD while changing seeds flips 18.18% and 18.22%. Similar flip rates do not establish model equivalence.

Optimization changes are observable without a resolved task gain. During GRPO, gradient norm grows from 0.148 to 0.907 and reference KL from 0.014 to 1.011. Parameter displacement and short-range direction changes are visible in Figure 6, but these diagnostics do not establish implementation correctness or explain the task result. During OPSD, the audited raw divergence decreases from 0.025461 to 0.023765 (6.66%) but is non-monotonic as the teacher context changes. This pattern is consistent with objective mismatch, while limited optimization and other implementation choices remain alternatives.

The protocol control demonstrates sensitivity to a large change. ReAct reduces EM by -10.25 pp, increases output tokens by a factor of 3.78, and raises format failures from 3.15% to 6.93%. It is a complete protocol comparison, not evidence that the evaluation resolves every sub-percentage-point training effect.

The split-sample per-question analysis places GRPO and OPSD near the pseudo-null that reuses the base-native matrix, whereas ReAct shows large heterogeneity (the full visualization is in Appendix C). The conditional standard-deviation bounds of 2.22 and 2.21 pp are derived from bootstrap interval endpoints; they do not prove that all question-level effects are zero or exclude effects outside the observed runs. **Answer to RQ4:** the audited optimizers changed parameters or their objectives, but the measured average task differences are small and uncertain. The controls constrain interpretation without establishing model equivalence or a unique failure mechanism.

5 Related Work

Outcome-reward agents include Search-R1, ReSearch, DeepSeek-R1, and the GRPO family (Jin et al., 2025; Chen et al., 2025; Guo et al., 2025; Shao et al., 2024; Yu et al., 2025; Liu et al., 2025b). Prior work studies optimization or filtering bias, including phantom advantages (Yu, 2026b), and shows that task gains can appear even under corrupted rewards (Shao et al., 2026); REA instead

asks whether a retained-group log remains interpretable as a progress signal. On-policy distillation methods such as GKD, MiniLLM, and OPSD (Agarwal et al., 2024; Gu et al., 2024; Zhao et al., 2026) motivate our audit: an objective can decrease without a detectable task change, consistent with objective mismatch but not proving it (Lambert et al., 2020; Lambert & Calandra, 2023).

REA is closest to work on paired resolution and structured evaluation noise (Kotawala, 2026; Kaliyev & Maryanskyy, 2026; Messing, 2026; Potamitis et al., 2025) and on variance-based tests for effect heterogeneity (Yu, 2026a); we add run-level variance and a conditional split-sample effect-variance diagnostic while warning that our checkpoint panels are not configuration-identical. Pass@k estimators and identifiability results (Chen et al., 2021; Brown et al., 2024; Singh & Singh, 2026) motivate our moment bounds and model-conditional deconvolution (Laird, 1978; Dempster et al., 1977; Wood, 1999). Gradient diagnostics build on McCandlish et al. (2018) and critical-batch/GRPO analyses (Merrill et al., 2025; Chang et al., 2026; Asadulaev et al., 2026; Yuan, 2026); forecasting work (Luo et al., 2026; Li et al., 2026; Patel et al., 2026) asks what training will reach, whereas REA asks what an evaluation can detect. Appendix D gives further context.

6 Discussion and Limitations

REA separates the interpretation of a training signal from the precision of an evaluation. In this case study both sources of risk were present: filtered reward logging changed the tracked estimand, and run-level variation accounted for a substantial share of repeated-evaluation variance under the tested seed protocol. These observations show how the framework can be used to report measurement risk; broader decision utility remains to be evaluated.

Scope and limitations. Evidence covers one model–task setting, one GRPO run and one OPSD run, with no independent training repeats. Manifest differences (resource paths, model-config fields, tool-schema descriptions) mean paired deltas are checkpoint-panel observations rather than configuration-isolated causal effects. The gradient panel used a mismatched configuration ($\beta_{\mathrm{KL}} = 0$, strict tool definitions). Pass@G summaries are conditional plug-in diagnostics under an iid-within-question model; our panels exhibit run-shared offsets that violate this assumption. Split-sample effect-variance intervals condition on twelve observed runs and use a pseudo-null (base-native matrix reused on both sides) rather than an independent same-model control. The run-level component is compatible with, but does not causally establish, a seed effect; a seed ablation varying seed and process independently would be needed for identification.

Cost accounting. The 1.88% figure is a 1,024-trajectory sampling-panel ratio; full REA also includes gradient computation (64 prompts, one forward-backward pass each), 355,440 evaluation trajectories across four twelve-seed panels, and CPU analysis. Caching opportunities exist but are not combined into a single percentage here.

Recommendations. We recommend: (1) log the unfiltered mean before any group filter; (2) make variance assumptions explicit and use same-seed pairing when run-level components are present; (3) establish an independent same-model null for effect-variance claims; (4) separate observed checkpoint differences from configuration-isolated causal claims; (5) recognize that REA establishes pre-training risk, not training success.

7 Conclusion

We introduced Resolvable-Effect Accounting (REA) as a framework for auditing reward summaries and evaluation uncertainty in tool-calling agent training. In the Qwen3.5-2B/HotpotQA case study, filtering zero-variance groups changed the reward estimand and produced a logged trajectory that differed from the unfiltered mean. Repeated evaluations exhibited substantial run-level variation, and same-seed pairing reduced comparison uncertainty under the observed cross-condition covariance. The GRPO and OPSD checkpoint differences were $+0.098$ and $+0.032$ pp, respectively, with 95% intervals including zero; these are observed, configuration-dependent differences rather than isolated causal training effects.

Practically, this argues for logging the unfiltered reward and stating the run and question units behind any resolution projection; broader decision utility still requires evaluation under matched configurations and additional training repetitions.

AI USE STATEMENT

We used AI coding assistants (Claude Code and Codex, each backed by several underlying models over the course of the project) throughout this work, matching the scope declared on our submission form: writing and language polishing, literature search, research ideation and execution, section drafting, and assistance with the derivations in Appendix A. Concretely, these assistants drafted the analysis scripts, data-processing pipelines and statistical estimators, helped derive the moment bounds and variance decompositions, and drafted and revised sections of this manuscript. The authors directed the work, ran the experiments, and audited the output: every quantity reported here is reproduced from a logged run configuration, evaluation manifest or analysis artifact, and the estimators carry automated tests against known distributions. Literature retrieved with AI assistance was checked against the original sources. The authors take full responsibility for the final text, the experimental design, the statistical claims and all conclusions, including any error that survived this process.

ETHICS STATEMENT

This work studies measurement methodology for RL training of tool-calling agents. It involves no human subjects, releases no new dataset, and introduces no technology with direct harmful application. The primary ethical consideration is research integrity, so we state explicitly that all evaluation panels postdate the training runs, that the gradient panel's configuration differs from the trained run, and that results cover one model and one task without independent training repeats. We distinguish identified quantities from extrapolated ones throughout.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B give complete hyperparameters for both training runs, the generation seeds and pairing protocol, retrieval configuration and environment versions; Appendix A gives explicit formulas for every estimator. Each reported number traces to a run configuration or analysis artifact, and evaluation products carry cryptographic fingerprints of the question set, scorer and configuration. We will release the framework code, trained adapters and full evaluation rollouts.

ACKNOWLEDGMENTS

We thank the open-source community for the Qwen models, the HotpotQA dataset and the Pyserini retrieval toolkit.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

Arip Asadulaev, Daniil Ognev, Karim Salta, and Martin Takac. Value-gradient hypothesis of RL for LLMs. *arXiv preprint arXiv:2605.21654*, 2026.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Zhepeng Cen, Yihang Yao, William Han, Zuxin Liu, and Ding Zhao. Behavior injection: Preparing language models for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Powei Chang, Jinpeng Zhang, Chaoqun Sun, MiniWell Tsao, Lianrui Li, Jianxiang Xiang, Chenyu Wang, Yukang Gao, and Dongying Kong. SALT: When more rollouts don't help in group-based policy optimization and how to make them matter. *arXiv preprint arXiv:2606.05800*, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri,

Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. ReSearch: Learning to reason with search for LLMs via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22, 1977.

Felix Elwert and Christopher Winship. Endogenous selection bias: The problem of conditioning on a collider variable. *Annual Review of Sociology*, 40(1):31–53, 2014.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

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. In *Proceedings of the 2nd Conference on Language Modeling*, 2025.

Alibek T Kaliyev and Artem Maryanskyy. How much coordination gain is real? A paired noise-floor protocol for multi-agent LLM benchmarks. *arXiv preprint arXiv:2606.20695*, 2026.

Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S. Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for LLMs. In *International Conference on Learning Representations*, 2026.

Anany Kotawala. Resolution diagnostics for paired LLM evaluation. *arXiv preprint arXiv:2605.30315*, 2026.

Nan Laird. Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association*, 73(364):805–811, 1978.

Nathan Lambert and Roberto Calandra. The alignment ceiling: Objective mismatch in reinforcement learning from human feedback. *arXiv preprint arXiv:2311.00168*, 2023.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *PMLR*, pp. 761–770, 2020.

Xiaoyuan Li, Yubo Ma, Kexin Yang, Moxin Li, Keqin Bao, Wenie Wang, Fuli Feng, and Dayiheng Liu. On predicting the post-training potential of pre-trained LLMs. *arXiv preprint arXiv:2605.11978*, 2026.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025a.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. In *Proceedings of the 2nd Conference on Language Modeling*, 2025b.

Yuxiang Luo, Haonan Long, Chen Wang, Qiqi Duan, Xiaotian Lin, Yanwei Xu, Yuyu Luo, Weikai Yang, and Nan Tang. TuneAhead: Predicting fine-tuning performance before training begins. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

William Merrill, Shane Arora, Dirk Groeneveld, and Hannaneh Hajishirzi. Critical batch size revisited: A simple empirical approach to large-batch language model training. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Solomon Messing. Hidden measurement error in LLM pipelines distorts annotation, evaluation, and benchmarking. *arXiv preprint arXiv:2604.11581*, 2026.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Arkil Patel, Siva Reddy, Marius Mosbach, and Dzmitry Bahdanau. Forecasting downstream performance of LLMs with proxy metrics. *arXiv preprint arXiv:2605.18607*, 2026.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Nearchos Potamitis, Vansh Ramani, Har Ashish Arora, Dhairya Kuchhal, Lars Klein, and Akhil Arora. ReasonBENCH: Benchmarking the (in)stability of LLM reasoning. *arXiv preprint arXiv:2512.07795*, 2025.

Qwen Team. Qwen3.5-2B. Hugging Face model card, 2026. URL `https://huggingface.co/Qwen/Qwen3.5-2B`.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in RLVR. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

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.

Pranav Singh and Prashant Singh. What fixed-rollout pass@k evaluations can identify. *arXiv preprint arXiv:2609.09245*, 2026.

Henry Teicher. Identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 34(4): 1265–1269, 1963.

G. R. Wood. Binomial mixtures: Geometric estimation of the mixing distribution. *The Annals of Statistics*, 27(5):1706–1721, 1999.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Fangzhou Yu. A variance-based test for heterogeneous treatment effects. *arXiv preprint arXiv:2607.17451*, 2026a.

Juntao Yu. The filter metric is safety-critical: Phantom advantages in group-relative RL under shaped rewards. *arXiv preprint arXiv:2609.13866*, 2026b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Ying Yuan. Detecting an effect is not learning to act on it: A reward-SNR floor for LLM acquisition agents. *arXiv preprint arXiv:2608.10441*, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, David Z. Pan, Zhangyang Wang, Yuandong Tian, and Kai Sheng Tai. The path not taken: RLVR provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

A Statistical Estimators

A.1 Binomial Deconvolution for the Success Distribution

Under the working iid-within-question model, $K_i \mid q_i \sim \mathrm{Binomial}(n, q_i)$ and we observe counts $k_i \in \{0, \ldots, n\}$. The likelihood of the count vector under mixing distribution F is

$$L(F) = \prod_{i=1}^{N} \int_0^1 \binom{n}{k_i} q^{k_i} (1-q)^{n-k_i} \, dF(q). \tag{3}$$

We fit a nonparametric mixing distribution on a fixed discrete support by EM (Laird, 1978; Dempster et al., 1977), updating weights until the weight shift falls below 10^{-12} or 50,000 iterations elapse. We verify that the fitted mean matches the observed success rate and that the maximum absolute residual between observed and expected count frequencies stays small: 0.0025 on the twelve-run evaluation panel and 0.0040 on the training-group panel. The fitted distribution is a model-dependent representation of the observed finite-depth counts; we do not claim to recover the full latent F nonparametrically. Because each evaluation run reuses one seed across questions and exhibits a run-level offset, the iid model is not a dependence-robust description of those panels; the fitted F and derived pass@G quantities are therefore conditional plug-in diagnostics.

For $G > n$ pass@G is not identified, and we report two distinct objects. The moment bounds use monotonicity and log-convexity of $R(t) = \mathbb{E}[(1-q)^t]$,

$$R(n) \left[\frac{R(n)}{R(n-1)} \right]^{G-n} \leq R(G) \leq R(n), \tag{4}$$

giving pass@$G \in \big[1 - R(n),\, 1 - R(n)[R(n)/R(n-1)]^{G-n}\big]$. These are plug-in constraints from sample moments, not sharp bounds and not confidence intervals. Separately, the model-conditional point estimate evaluates pass@$G = 1 - \int_0^1 (1-q)^G dF(q)$ on the fitted F, which is extrapolation conditioned on the model class. Question-level uncertainty is reported through a third channel: within the observed depth we use a Hoeffding interval on the question-level unbiased estimator, and beyond it a Clopper–Pearson interval on the zero-success frequency combined with Jensen and monotonicity for conservative coverage. The extrapolated certificate additionally assumes a common zero-success probability under a pooled random-task model and does not correct for the run-level offset or other within-run dependence.

A.2 Run-Level Variance Decomposition

For a fixed evaluation condition a, use the working decomposition

$$Y_{ij}^{(a)} = q_i^{(a)} + \delta_j^{(a)} + \epsilon_{ij}^{(a)}, \tag{5}$$

where $\delta_j^{(a)}$ is a run-level offset and $\epsilon_{ij}^{(a)}$ is question-level sampling noise. This is a variance model for run means; because Y is binary, the additive representation is an approximation rather than a literal probability model. Define $\sigma^2_{\text{question,mean},a} = \mathbb{E}[q_i^{(a)}(1 - q_i^{(a)})]/N$, the question-level contribution to one run mean. Under the working assumptions, a run mean has variance $\sigma^2_{\text{question,mean},a} + \sigma^2_{\text{run},a}$. We estimate it by computing the R run means $\bar{Y}_j^{(a)}$, estimating this mean-level question term from the unbiased per-question sampling variance, and subtracting:

$$\hat{\sigma}^2_{\text{run},a} = \mathrm{Var}(\bar{Y}_j^{(a)}) - \hat{\sigma}^2_{\text{question,mean},a}. \tag{6}$$

With m the noncentral-t multiplier for R runs at power 0.8 and $\alpha = 0.05$ (here $m = 3.078$), the two designs give

$$\mathrm{MDE}_{\text{unpaired}} = \frac{m\sqrt{2}\,\sigma_{\text{single}}}{\sqrt{R}}, \qquad \mathrm{MDE}_{\text{paired}} = \frac{m\,\sigma_{\text{paired}}}{\sqrt{R}}, \tag{7}$$

where $\sigma^2_{\text{single}} = \sigma^2_{\text{question,mean}} + \sigma^2_{\text{run}}$ and σ_{paired} is the observed sd of per-seed paired differences when a paired panel exists. The paired run-level contribution is $\sigma_1^2 + \sigma_0^2 - 2\,\mathrm{Cov}(\delta^{(1)}, \delta^{(0)})$; replacing it by zero is only a common-offset planning approximation. For a projection from N_0 to N questions, the question-level term becomes $\sigma^2_{\text{question,mean}}(N) = \sigma^2_{\text{question,mean}}(N_0)N_0/N$, while the unpaired run component stays fixed. This is why the unpaired projection approaches $m\sqrt{2}\,\sigma_{\text{run}}/\sqrt{R} = 0.67$ pp when R is fixed. For paired comparisons across variants we also report $\mathrm{Cor}(\bar{Y}_j^{\text{base}}, \bar{Y}_j^{\text{variant}})$ as evidence about shared run-level variation.

A.3 Per-Question Effect Variance

Let $\Delta_i = q_i^{\text{trained}} - q_i^{\text{base}}$ and $V = \mathrm{Var}_i(\Delta_i)$. For each of the $\binom{12}{6} = 924$ partitions of the runs into disjoint halves A and B, compute the trained–base contrast $\hat{\Delta}_i = \bar{Y}_{i,B}^{\text{trained}} - \bar{Y}_{i,A}^{\text{base}}$. If $k_i^{(a)}$ is the number of successes for question i in the six-run half for condition a, the unbiased variance estimate for that half-mean is $\hat{v}_i^{(a)} = k_i^{(a)}(6 - k_i^{(a)})/(6^2 \cdot 5)$, so we use

$$\hat{V} = \mathrm{Var}_i(\hat{\Delta}_i) - \frac{1}{N}\sum_{i=1}^{N}\left[\hat{v}_i^{\text{trained}} + \hat{v}_i^{\text{base}}\right]. \tag{8}$$

This correction uses the two condition-specific sampling variances for six-run half-means. Our halves are disjoint, so no same-seed covariance term is included; a seed-paired construction would require that covariance correction. We report the mean over all partitions. The pseudo-null in the current report reuses the observed base-native matrix on both sides, so it checks estimator behavior but is not an independent true-zero calibration. Bootstrap intervals resample questions with the twelve runs held fixed, so they are conditional on the observed runs and do not cover fresh-seed uncertainty.

A.4 Gradient Systematic-to-Noise Ratio

From per-prompt gradients $g_1, \ldots, g_B$,

$$\mu = \frac{1}{B}\sum_{i=1}^{B} g_i, \quad \Sigma = \frac{1}{B}\sum_{i=1}^{B}(g_i - \mu)(g_i - \mu)^\top, \quad \rho = \frac{\|\mu\|^2}{\mathrm{tr}(\Sigma)}, \tag{9}$$

where the displayed ρ is the population target. We estimate the signal using a Gram matrix with the finite-panel correction $\widehat{\|\mu\|^2} = \|\bar{g}\|^2 - \widehat{\mathrm{tr}(\Sigma)}/B$, then report $\hat{\rho} = \widehat{\|\mu\|^2}/\widehat{\mathrm{tr}(\Sigma)}$. The implied critical-batch scale $1/\hat{\rho}$ is unstable when the interval includes zero; it is not a calibrated recommendation. Jackknife intervals leave out one prompt at a time. The batch-B signal proxy we report is a second-moment quantity, not an expected cosine.

B IMPLEMENTATION DETAILS

Tables 2–4 collect the configurations and accounting used in the experiments. Both training runs use a frozen Qwen3.5-2B checkpoint with LoRA (rank 16, $\alpha = 32$, dropout 0), leaving 16,819,200 trainable scalars; evaluation-only panels do not train parameters. The training loop is our own implementation rather than a third-party RL library.

Table 2: Training configurations. Entries in the two method columns are not intended to make GRPO and OPSD comparable objectives; they document the exact runs audited in this paper.

Setting	GRPO	OPSD
Training data	HotpotQA hard train, 15,661 records	HotpotQA hard train, 15,661 records
Updates / batching	240 updates; 8 prompts/update; $G = 16$	20 updates; micro-batch 4; 131,072-token budget/update
Objective	Group-relative outcome reward; $\beta_{\mathrm{KL}} = 0.04$; clip $\epsilon = 0.2$	Generalized JSD, $\beta = 0.5$; vocabulary contribution clip 0.05
Optimizer	AdamW, 2×10^{-5}; 5 warmup steps; constant; no weight decay	AdamW, 2×10^{-5}; 3 warmup steps; cosine to 10%
Optimization details	Advantage $\epsilon = 10^{-4}$; one inner iteration; grad clip 1.0; bf16; checkpointing	Teacher has answer/support context and thinking; student has neither; answer/query/tool-name weights 1
Context / cost	40,960-token limit; 54,608 trajectories; 44.06 GPU h	32,768-token context; 107,475 training tokens

The LoRA targets are `q_proj`, `k_proj`, `v_proj`, `o_proj`, `in_proj_qkv`, `z`, `a`, `b`, `out_proj`, and `gate_proj`, `up_proj`, `down_proj` across the attention and feed-forward blocks. GRPO rollouts use a detached old-policy forward pass and therefore store exact token IDs. OPSD student rollouts stop after the second tool-calling step, without executing that call, so no tool result from the truncated step enters a training context.

Table 3: Retrieval, generation, and evaluation protocol.

Component	Configuration
Retrieval	Pyserini 0.22.1 over frozen Wiki-18 Lucene; BM25 $k_1 = 1.2$, $b = 0.75$; top 3 passages, 1,200-character truncation; `query_only` passes the question text directly, with no expansion or pseudo-relevance feedback.
Training generation	Temperature 1.0, top-p 1.0; top-k/min-p disabled; presence penalty 0; 16,384-token cap per model call.
Evaluation generation	vLLM at concurrency 64; temperature 1.0, top-p 0.95, top-k 20, presence penalty 1.5; 16,384-token cap.
Agent limits	At most 5 tool calls and 8 model calls in a 32,768-token context.
Evaluation panel	Distractor validation split, 7,405 questions; 12 runs with seeds 42–53 held fixed across requests; native, GRPO, OPSD, and ReAct panels share question-set hash and scorer version.

Index, corpus, question-set, scorer, and configuration files are SHA-256 fingerprinted in run manifests. Concurrent rollout files are sorted by question identifier before analysis. Protocol auditing found differences in resource-path representation, raw model-config fields, and full fingerprints across panels; the tool-schema query description also differs between the base and trained manifests. Accordingly, the checkpoint panels are auxiliary monitoring rather than strictly controlled mechanism experiments: their paired EM differences are observed protocol-level differences, not causal estimates that isolate training from prompt or schema changes.

Table 4: Diagnostic and cost accounting. The percentages are trajectory-count ratios, not a complete GPU-cost estimate.

Item	Recorded accounting
Gradient panel	64 prompts, $G = 16$, $\beta_{\mathrm{KL}} = 0$, strict tool definitions; one forward–backward pass per prompt.
GRPO ledger	8.80 GPU h sampling, 35.22 GPU h updating, 0.037 GPU h overhead; 54,608 trajectories total.
Sampling-panel ratio	$64 \times 16 = 1,024$ trajectories, 1.88% of the GRPO trajectory count; this excludes gradient, evaluation, and CPU costs.
Evaluation panels	$4 \times 12 \times 7{,}405 = 355{,}440$ trajectories; GPU time was not recorded in the training ledger.
Software	Training: PyTorch 2.8.0, transformers 5.15.0, peft 0.20.0, accelerate 1.6.0, Python 3.12, CUDA 12.8; inference: vLLM 0.27.1; analysis: NumPy/SciPy.

The analysis package builds evaluation matrices from rollout files, computes the four REA quantities into one report, reconstructs which training groups were filtered, fits answer-level distributions for the voting control, and tracks parameter trajectories and optimizer state. Every estimator carries pytest coverage against known distributions, and reproduction commands ship with the anonymized supplementary material.

C Additional Results

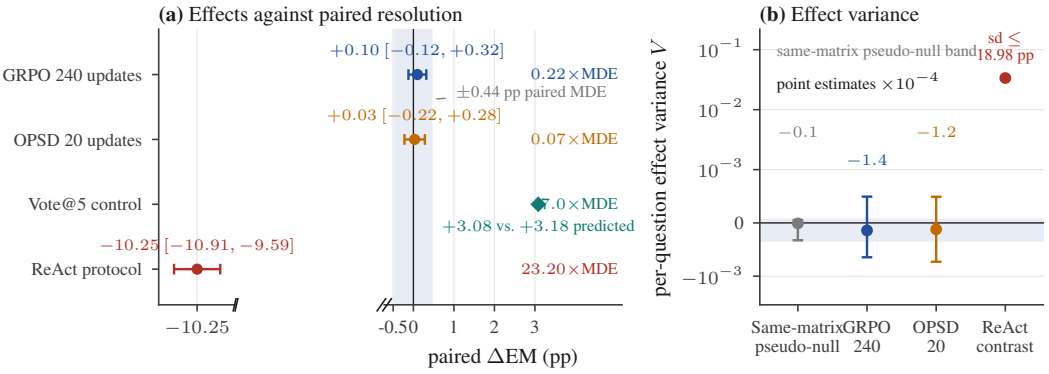

Figure 4: Observed checkpoint differences and controls. **(a)** Paired ΔEM with 95% intervals against the ±0.44 pp planning proxy. **(b)** Split-sample per-question effect-variance estimates: GRPO and OPSD are near the same-matrix pseudo-null, while ReAct shows much larger heterogeneity.

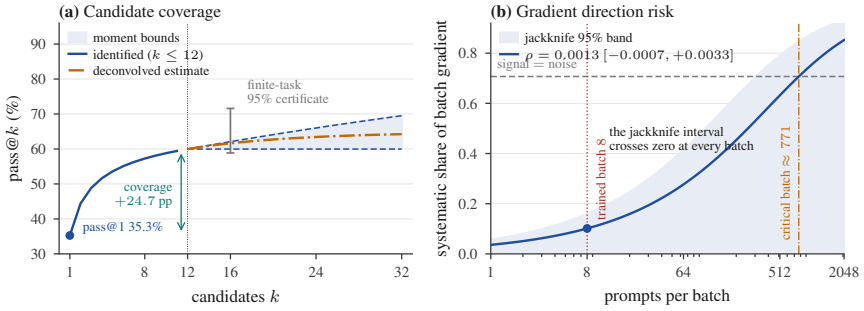

Figure 5: Supporting diagnostics are informative but inconclusive. Candidate coverage is identified at the observed rollout depth, whereas the gradient signal interval crosses zero in the exploratory panel.

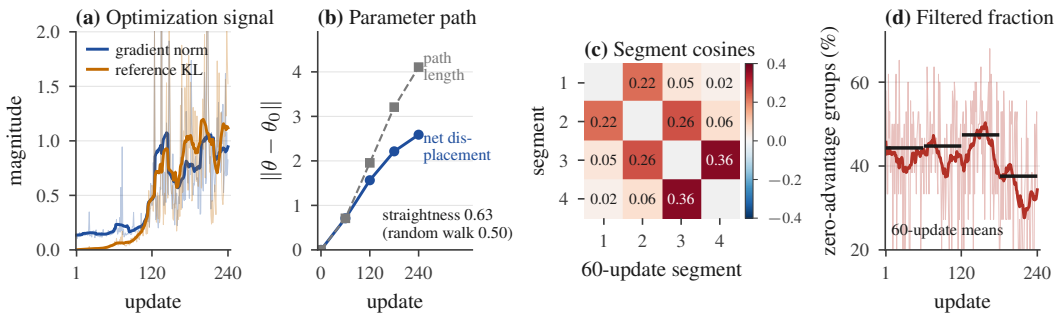

Figure 6: GRPO's optimization signal over 240 updates. **(a)** Gradient norm and sampled KL against the frozen base both grow by roughly an order of magnitude; faint lines are per-update values, heavy lines a 21-update centred mean. **(b)** Net displacement from the initial checkpoint against cumulative path length; the gap between them is what makes straightness 0.63 rather than 1. **(c)** Cosine similarity between the four 60-update displacement segments, diagonal masked. Adjacent segments agree weakly, distant ones barely at all. **(d)** The zero-advantage fraction itself drifts, falling in the final segment, so the set of groups the logged reward averages over is not stable across the run.

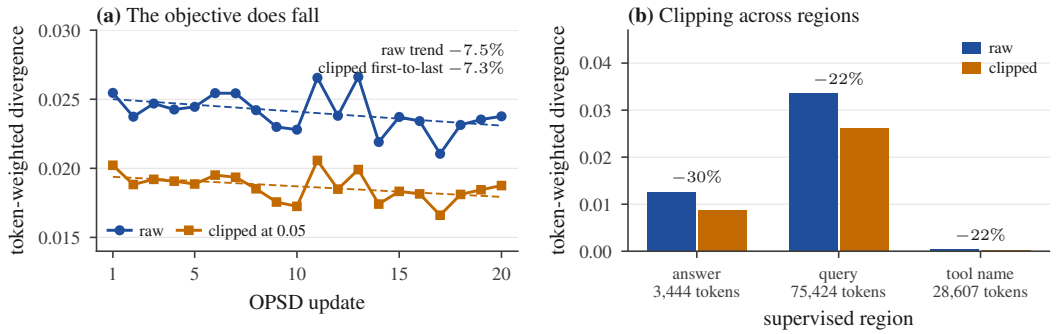

Figure 7: OPSD's divergence falls on both readings. **(a)** Token-weighted divergence per update, raw and clipped at 0.05, with least-squares trends. The decrease is not monotone: the teacher's context shifts as the student's on-policy rollouts evolve. **(b)** The same quantities split by supervised region. Clipping removes a similar fraction in each region, so the 22.42% gap is a property of the clip rather than an artifact of one region dominating the token budget.

C.1 Intermediate Checkpoints

Intermediate GRPO checkpoints evaluated on a 4,000-question panel with a single seed give EM of 35.33%, 35.23%, 35.30% and 35.63% at steps 60, 120, 180 and 240. OPSD at steps 5, 10, 15 and 20 on the same panel gives 34.93%, 34.38%, 35.30% and 35.03%. Both trajectories are non-monotonic and within noise, and neither panel is protocol-identical to the twelve-seed matrices, so they serve as monitoring rather than controlled comparison.

C.2 Format Compliance and Output Cost

Native tool_calls reach 96.85% format compliance against ReAct's 93.07%, so ReAct roughly doubles the failure rate; failures are incomplete JSON, unclosed braces or unparseable tool names. Average output tokens per question across all turns are 901.6 for native and 3405.7 for ReAct, a factor of 3.78, since text markers are more verbose than JSON keys and the model tends to add prose between them. ReAct also issues fewer tool calls (1.55 against 3.17) and fewer model calls (3.05 against 4.28) despite generating far more tokens. Since the agent aborts a trajectory after two

format errors, one candidate explanation is that ReAct trajectories end before exhausting their tool budget, but we did not instrument termination causes for this panel and do not claim it.

D Extended Related Work

Why parameters can move without metrics following. Mukherjee et al. (2025) find RL updates only 5–30% of parameters, and Zhu et al. (2025) show RLVR updates concentrate in low-curvature subspaces with minimal spectral drift. Both help explain how gradient norm and reference KL can grow while task metrics stay flat, as in Section 4.

When RL scaling helps. ProRL (Liu et al., 2025a) extends reasoning boundaries with long-horizon RL well beyond 240 updates, and Khatri et al. (2026) fit compute-performance curves that predict larger-scale gains. Our negative result at 240 updates does not speak to either regime.

Readiness signals. Cen et al. (2025) propose RL-informative rollout accuracy and data co-influence coefficient as RL readiness signals, which offers one account of why identical algorithms succeed on some base models and not others—a question REA does not address, since it measures resolvability rather than learnability.

Alternative evaluation protocols. tinyBenchmarks (Polo et al., 2024) uses item response theory to cut question count while preserving discriminative power. This is orthogonal to REA: IRT selects discriminative questions, whereas REA bounds what any question set can resolve once a run-level component is present.