DR.Q: MULTI-GRANULARITY CREDIT ASSIGNMENT FOR LONG-HORIZON TOOL-USING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-using language agents learn from trajectories that interleave planning, execution, recovery, and finalization. A trajectory-level reward combines the effects of these decisions, while local comparisons depend on which work states recur and which tool outcomes the environment can evaluate. We introduce Dr.Q, a multi-granularity credit estimator that combines trajectory outcomes, discounted process rewards, and cross-trajectory work-state comparisons. A Router organizes trajectory comparisons, Rubric and recovery-gate scores expose segment quality, and an Anchor term contrasts outcomes across repeated work states. Position-dependent weighting combines these signals within a shared policy update. We characterize when group-centered credit vanishes and how assigning zero reward to missing tool feedback can penalize exploration. On τ-bench, a 9B agent improves average success from 27.57% with GRPO to 40.26%. On a 200-question 2WikiMultihopQA development subset, reinforcement learning improves a 27B SFT model from 62.5% to 69.0% exact match. Configuration studies favor task grouping over route pooling, while replay analysis links uneven feedback coverage to reduced exploration. Code is available at `https://anonymous.4open.science/r/DRQ-Research-4599`.

1 INTRODUCTION

Tool-using language agents must learn how decisions combine into a successful workflow. Selecting an API and producing valid arguments are essential capabilities (Schick et al., 2023; Li et al., 2023b); success also requires interpreting observations, recovering from errors, and returning the requested answer. ReAct couples reasoning with these actions (Yao et al., 2023), while τ-bench tests the resulting workflow against user goals and domain policies (Yao et al., 2024). Our BIRD cases make the distinction concrete (Li et al., 2023a): executable queries return the wrong answer when they retain NULL scores in a minimum or interpret unit price as a unit-quantity filter (Figure 1). A terminal failure compresses these different decisions into the same outcome. The learning problem is to recover useful credit at the decisions that produced it.

Throughout the manuscript, Dr.Q denotes the revised method name Dr.Q. The labels DR.GRIPO and GRIPO retained in selected figures and historical experiment records refer to the same method family and are preserved for provenance; they should not be confused with the separate GRPO baseline.

Reinforcement learning improves mathematical reasoning and retrieval-augmented interaction using completion rewards (Shao et al., 2024; DeepSeek-AI et al., 2025; Jin et al., 2025). In outcome-supervised group-relative optimization, sampled attempts provide a baseline for each terminal reward. Broadcasting this comparison across a rollout couples decisions with different effects: a failed answer can follow a useful lookup, while a successful answer can include invalid calls. AgentBoard's separate progress and task-completion metrics expose the same distinction in evaluation (Ma et al., 2024).

Process supervision evaluates intermediate reasoning steps (Uesato et al., 2022; Lightman et al., 2023). For interactive agents, GiGPO compares return-to-go at repeated states (Feng et al., 2025), SALT derives advantages from trajectory graphs (Li et al., 2026a), Turn-PPO optimizes turns (Li et al., 2026b), and Agent Lightning decomposes traces into trainable transitions (Luo et al., 2025). Local comparisons depend on their population: the same segment index can denote schema inspection or recovery, and a route label can pool tasks with different outcome distributions. Even compatible groups provide no relative signal when their scores are identical.

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.

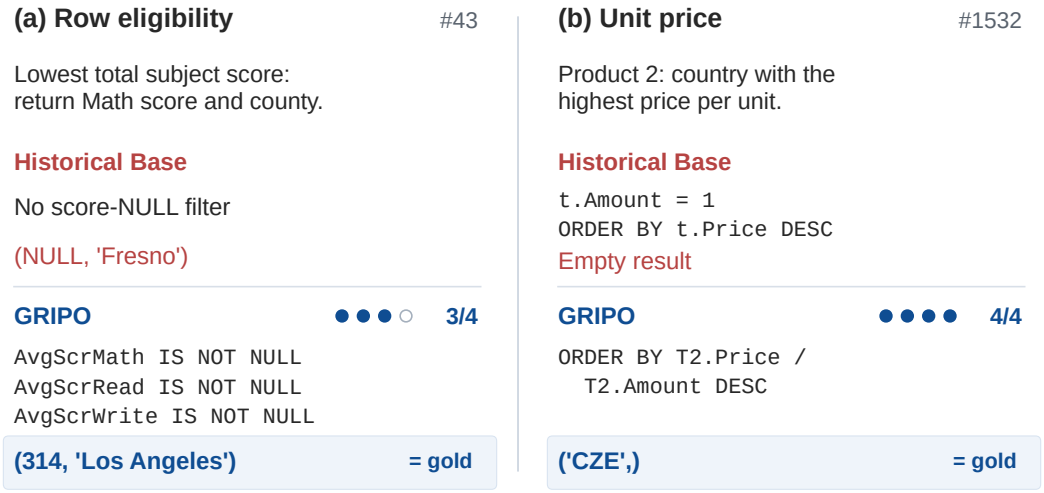

Figure 1: Concrete SQL decisions change the answer. Two BIRD-dev cases compare historical Qwen3.5-27B-Base predictions with selected Qwen3.5-27B-DR.GRIPO MAJ@4 predictions. Filtering NULL scores changes the selected minimum; ordering by unit price recovers the requested country. SQL predicates and ordering expressions are excerpted. Dots count candidates sharing the selected execution result. Section 5.6 and Appendix F analyze all five cases.

We study how credit granularity, comparison populations, and available feedback jointly determine the learning signal. *Compatibility* gives a comparison meaning; *reward contrast* gives centering a nonzero value. Complete trajectories, execution segments, and recurring work states offer complementary comparison units. Giving each its own score and population lets task completion, process quality, and state-conditioned outcomes contribute separately to learning.

Dr.Q implements this decomposition (Figure 2). A Router groups terminal outcomes by task or workflow route. Rubric and recovery-Gate scores define discounted segment returns. An Anchor compares terminal outcomes across distinct rollouts sharing a task-conditioned, realized work-state descriptor, including matches at different positions. Support weights regulate sparse anchor groups. Position weights and role masks combine the three signals in a shared PPO objective, separating the score being compared from the unit receiving credit.

DAPO filters equal-reward prompt groups, and RAGEN studies reward-variance collapse (Yu et al., 2025; Wang et al., 2025). We diagnose contrast separately in each credit branch. This exposes a replay-environment failure: scoring missing responses as zero can penalize unavailable feedback when it is centered against positive observed successes, even when terminal outcomes remain informative.

On τ-bench, a 9B Dr.Q agent increases average success from 27.57% with GRPO to 40.26%, a gain of 12.69 percentage points. On the same 200-question 2WikiMultihopQA evaluation, reinforcement learning improves a 27B SFT checkpoint from 62.5% to 69.0% exact match. Two 4B studies favor task grouping over route pooling. Replay measurements connect sparse feedback with declining exploration, and BIRD cases identify the SQL decisions behind contrasting execution outcomes.

2 Tool Trajectories and Comparison Units

For task x with identifier u, policy π_θ generates a rollout

$$\tau_i = (s_{i,1}, a_{i,1}, o_{i,1}, \ldots, s_{i,H_i}, a_{i,H_i}, o_{i,H_i}), \tag{1}$$

where s contains the available interaction context, a is a generated action, and o is the environment observation. The horizon H_i varies with execution and recovery. We sample multiple rollouts per task and denote a rollout's terminal outcome by R_i.

Environment events partition each rollout into segments $\sigma_{i,1}, \ldots, \sigma_{i,M_i}$. A tool invocation, an error-recovery transition, or finalization can delimit a segment. Segment j receives process reward $r_{i,j}$

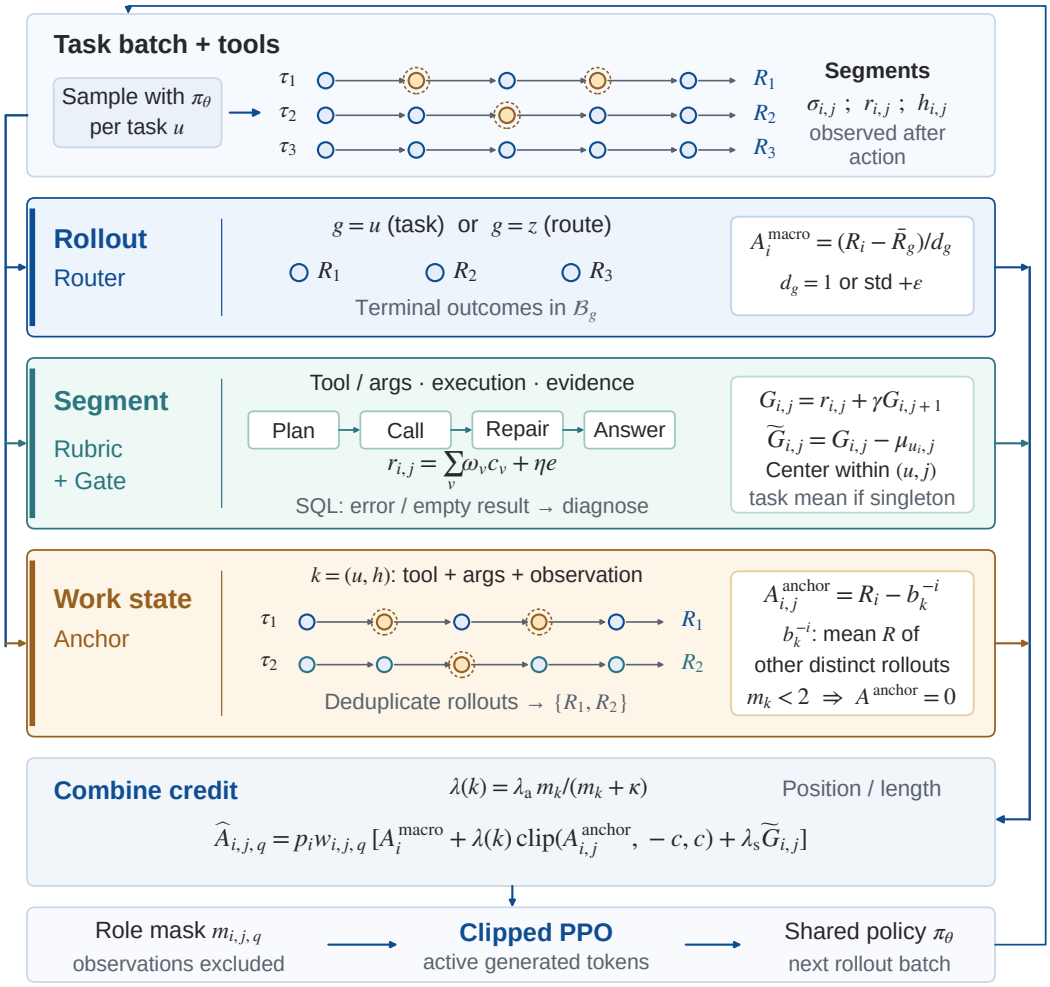

Figure 2: Dr.Q constructs three comparisons from the sampled batch. Router compares terminal outcomes within task or route groups. Rubric and Gate produce discounted segment returns, centered by task and position with a task-mean fallback. Anchor compares terminal outcomes of distinct rollouts sharing (u, h). Orange rings mark the same realized descriptor; repeated visits within a rollout contribute once to the anchor baseline. Support weighting, anchor clipping, and position/length weights compose the credit before role masking and PPO. The SQL recovery sequence and trajectories are schematic.

and contains generated tokens indexed by q. A work-state descriptor $h_{i,j}$ summarizes the realized segment using normalized task-relevant information, such as tool identity, argument structure, an execution outcome class, or a resolved schema hypothesis. These descriptors allow recurrent work configurations to be matched across rollouts even when their absolute positions differ.

Compatibility depends on task identity, role context, and descriptor semantics. Reward contrast depends on variation in the compared scores. A group can satisfy either condition independently; Dr.Q makes both explicit at each granularity.

3 Multi-Granularity Credit Decomposition

3.1 Trajectory Credit and Routing

The Router assigns task x_i a heuristic route z_i, such as query, construction, diagnosis, report, or a composite workflow. Routing can select a generation pipeline and determine the trajectory comparison

key g_i. Task grouping uses $g_i = u_i$, retaining the candidate rollouts of one task. Route grouping uses $g_i = z_i$, pooling outcomes from tasks assigned to the same route. Keeping these keys distinct exposes the effect of pooling across task difficulty and reward distributions.

For comparison set $\mathcal{B}_g = \{i : g_i = g\}$, the macro term is

$$A_i^{\text{macro}} = \frac{R_i - \bar{R}_{g_i}}{d_{g_i}}, \qquad \bar{R}_g = \frac{1}{|\mathcal{B}_g|} \sum_{m \in \mathcal{B}_g} R_m, \tag{2}$$

where d_g is one or the group standard deviation plus a stabilizer. Task grouping with standard-deviation normalization recovers GRPO's outcome-relative construction (Shao et al., 2024). Route pooling changes the baseline population; Section 5.4 tests this choice.

3.2 Segment Credit from Process Rewards

Process rewards expose execution quality before completion. Segment criteria cover tool choice, arguments, syntax, tables and fields, user intent, and supporting evidence, extending rubric-based reward design (Gunjal et al., 2025) to tool workflows. A weighted rubric and recovery gate yield

$$r_{i,j} = \sum_{v=1}^{V} \omega_v c_v(\sigma_{i,j}) + \eta\, e(\sigma_{i,j}). \tag{3}$$

Here c_v is a criterion score, ω_v is its domain-specific weight, and e evaluates a recovery decision. In the SQL agent, an execution error or empty result can activate a diagnostic prompt. The recovery score rewards a successful eligible diagnostic action and penalizes ignoring an active diagnostic request. This makes recovery behavior visible before the final query succeeds.

To propagate such information across segments, we compute

$$G_{i,j} = r_{i,j} + \gamma G_{i,j+1}, \qquad G_{i,M_i+1} = 0. \tag{4}$$

Let $\mathcal{C}_{u,j}$ contain the segments with task identifier u and segment index j. The baseline uses their mean when at least two members are available and falls back to the task-level return mean $\bar{G}_u^{\text{task}}$ otherwise:

$$\mu_{u,j} = \begin{cases} \dfrac{1}{|\mathcal{C}_{u,j}|} \displaystyle\sum_{m \in \mathcal{C}_{u,j}} G_{m,j}, & |\mathcal{C}_{u,j}| \geq 2, \\ \bar{G}_u^{\text{task}}, & |\mathcal{C}_{u,j}| < 2, \end{cases} \tag{5}$$

$$\widetilde{G}_{i,j} = G_{i,j} - \mu_{u_i,j}. \tag{6}$$

The fallback retains a task-level comparison for trajectories with different numbers of segments. Segment-index grouping aligns the work sequence coarsely; work-state anchors provide a separate comparison when equivalent configurations occur at different positions.

3.3 Cross-Trajectory Work-State Anchors

Repeated work states connect intermediate execution to the terminal outcome. We construct anchor key $k_{i,j} = (u_i, h_{i,j})$ and collect the distinct rollouts visiting that key in $\mathcal{D}_k$. A rollout contributes once to the baseline even when it revisits a state. Let $m_k = |\mathcal{D}_k|$. The leave-one-rollout-out baseline and anchor term are

$$b_k^{-i} = \frac{1}{m_k - 1} \sum_{m \in \mathcal{D}_k \setminus \{i\}} R_m, \qquad m_k \geq 2, \tag{7}$$

$$A_{i,j}^{\text{anchor}} = \begin{cases} R_i - b_{k_{i,j}}^{-i}, & m_{k_{i,j}} \geq 2, \\ 0, & m_{k_{i,j}} < 2. \end{cases} \tag{8}$$

This branch compares terminal outcomes across rollouts reaching a recurring work configuration. Related repeated-state estimators compare return-to-go (Feng et al., 2025); here each distinct rollout contributes one terminal outcome. The segment branch separately compares discounted process returns.

The realized descriptor can include the chosen tool, normalized arguments, and the resulting observation. The Anchor therefore shapes credit using both an operation and its outcome. Task conditioning retains problem identity, while normalization reduces superficial variation in arguments and observations.

Sparse recurrence reduces the amount of comparative evidence. We scale the branch by

$$\lambda(k) = \lambda_{\mathrm{a}} \frac{m_k}{m_k + \kappa}, \qquad \kappa > 0, \tag{9}$$

and clip the anchor advantage to $[-c, c]$. The singleton rule in Equation 8 gives zero contribution when there is no cross-rollout comparator. Larger groups receive greater weight, while clipping limits the influence of an extreme terminal-outcome difference.

3.4 Advantage Composition and Policy Update

For token q in segment j of rollout i, the composed advantage is

$$\widehat{A}_{i,j,q} = p_i w_{i,j,q} \left[A_i^{\mathrm{macro}} + \lambda(k_{i,j}) \operatorname{clip}(A_{i,j}^{\mathrm{anchor}}, -c, c) + \lambda_{\mathrm{s}} \widetilde{G}_{i,j} \right]. \tag{10}$$

The position weight w distinguishes planning or start segments, intermediate execution, and finalization. The optional length factor p_i is one within the generation budget and a configured penalty weight for overlong rollouts, which retain their rewards and comparison-group membership.

Planner, executor, and diagnostic generations share parameters but can follow different training schedules. A role mask $m_{i,j,q} \in \{0, 1\}$ selects active generated tokens after advantage construction. Environment observations are excluded from the policy loss. With likelihood ratio $\rho_{i,j,q} = \pi_\theta(a_{i,j,q} \mid s_{i,j,q}) / \pi_{\theta_{\mathrm{old}}}(a_{i,j,q} \mid s_{i,j,q})$, we minimize

$$\mathcal{L}(\theta) = -\mathbb{E}\left[m_{i,j,q} \min\left(\rho_{i,j,q} \widehat{A}_{i,j,q}, \operatorname{clip}(\rho_{i,j,q}, 1-\epsilon, 1+\epsilon) \widehat{A}_{i,j,q} \right) \right]. \tag{11}$$

This uses the clipped surrogate of PPO (Schulman et al., 2017). Masks are applied without renormalizing the advantages, preserving their computed scale. The estimator is implemented through generation, reward, and advantage hooks in the veRL training stack (Sheng et al., 2025). Appendix A gives the complete update sequence.

4 When Local Comparisons Carry Credit

Reward contrast must be evaluated for each branch's actual score: terminal outcomes for macro and Anchor credit, discounted returns for segment credit.

Proposition 1 (Local reward contrast)**.** *For a finite comparison group $\mathcal{C}$ of size $m \geq 1$, let y_i be its scores, $\bar{y}$ their mean, and $a_i = y_i - \bar{y}$. Then*

$$\sum_{i \in \mathcal{C}} a_i^2 = m \operatorname{Var}_{i \in \mathcal{C}}(y_i). \tag{12}$$

Thus all centered scores vanish exactly when the scores are constant within the group. If a fraction $0 < \alpha < 1$ receives c and the rest receives $c' > c$, each member at c has centered score $(1-\alpha)(c-c') < 0$.

Appendix B derives this identity and its leave-one-out counterpart. We call loss of contrast in one branch *granularity degeneracy*; other branches can still contribute. Future rewards can preserve contrast in discounted segment returns even when the current reward is constant.

In partially covered environments, unmatched responses may receive zero while observed successes receive positive scores. Centering then penalizes unmatched calls in mixed groups. Such groups retain contrast and survive equal-reward filtering (Yu et al., 2025), so the local signal can favor well-covered tool families. Section 5.5 examines this pattern in replay trajectories.

State recurrence introduces a second condition. A hash match identifies a common descriptor, while histories leading to that descriptor can carry different predictive information. Let c_i denote such prefix information. For a comparison pool satisfying $\mathbb{E}[b_k^{-i} \mid k, c_i] = \mathbb{E}[R \mid k]$,

$$\mathbb{E}[R_i - b_k^{-i} \mid k, c_i] = \mathbb{E}[R_i \mid k, c_i] - \mathbb{E}[R \mid k]. \tag{13}$$

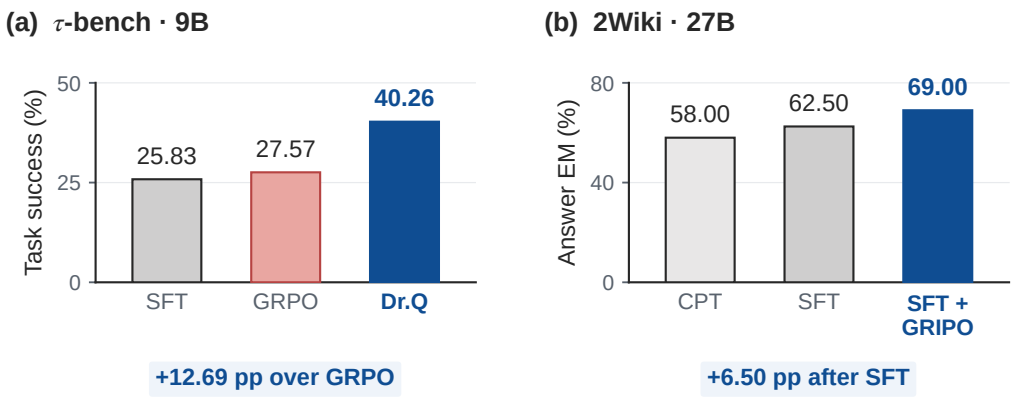

Figure 3: Task and answer success under the two main evaluation settings. (a) Qwen3.5-9B on τ-bench, averaging retail and airline success equally. (b) A 27B checkpoint lineage evaluated on the same 200-question 2Wiki development subset with greedy decoding. Gains are percentage-point differences. Domain scores and the full 2Wiki metric breakdown appear in Tables 1 and 2.

The resulting score can therefore reflect both continuation quality and information retained in the prefix. This motivates task conditioning and explicit descriptor semantics. Support weighting controls the contribution of small groups, and clipping controls scale. Equation 11 directly optimizes the composed surrogate, including its action- and observation-dependent shaping terms.

Nested recovery or delegated sub-loops require the same ingredients at each level: a defined unit, compatible comparison pool, and observable score contrast. Coverage and recurrence determine where additional local credit is useful.

5 Experiments

5.1 Tasks, Models, and Evaluation

The τ-bench study compares Qwen3.5-9B SFT, outcome-supervised GRPO, and Dr.Q on retail and airline tasks, with up to 20 interaction steps. We report domain success and its unweighted average.

The 2WikiMultihopQA environment provides entity search, page lookup, passage retrieval, and answer submission over a BM25 index of each question's context (Ho et al., 2020). A 27B continued-pretraining (CPT) checkpoint, its cold-start SFT descendant, and the checkpoint after 100 joint Dr.Q steps share a 200-question development subset, harness, and greedy decoding. Reinforcement learning samples four rollouts per question. Metrics cover answer exact match (EM) and F1, execution match, grounding, and protocol parsing.

A 4B configuration study uses 100 training steps, eight rollouts per task, and 6,400 rollouts per configuration, reporting the last 25 steps. Replay analysis covers 2,992 rollouts against a partially covered response library. Appendix C details the protocols; Section 5.6 introduces the BIRD case comparison.

5.2 Task Success on τ-bench

Figure 3a shows that Dr.Q improves average success from 27.57% to 40.26%, a gain of 12.69 percentage points over GRPO. Retail success rises by 17.39 points and airline success by 8.00 points. Relative to SFT, the average gain is 14.43 points. The gains occur in both domains, with the greatest improvement in the retail setting where the outcome-supervised baseline is weakest.

5.3 Separating SFT and Reinforcement-Learning Gains

Figure 3b separates two improvements along one weight lineage. Cold-start SFT increases EM from 58.00% to 62.50% and grounding from 75.00% to 85.00%. Subsequent reinforcement learning adds

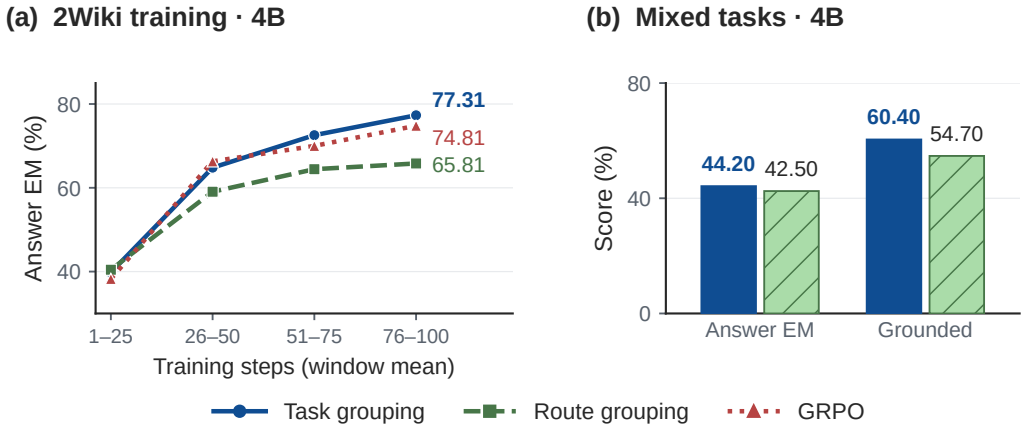

Figure 4: The choice of comparison group changes 4B training outcomes. (a) Answer EM averaged over each 25-step training window on 2Wiki. Markers show the four observed window means. (b) Scores after training on 2,000 2Wiki samples and 500 GSM8K-derived distractor samples. Task grouping retains question identity; route grouping pools tasks with the same route label.

6.50 EM points and 7.00 execution-match points. It also reduces the protocol parse-failure rate from 28.50% to zero in this evaluation. The answer improvement accompanies a reduction in mean interaction length from 6.03 to 5.41 turns.

The grounding metric moves from 85.00% after SFT to 83.00% after reinforcement learning, showing that answer accuracy and evidence support respond differently to post-training. The per-type breakdown in Appendix D localizes the answer gains. Compositional questions improve from 41.77% to 53.16% EM, while inference questions remain at 47.06%. This identifies a concrete distinction between improved tool execution and the residual reasoning demands of the task.

5.4 The Comparison Group Changes the Learning Signal

The 4B study makes the population defining the macro baseline explicit. With task grouping, Dr.Q reaches 77.31% training EM, compared with 65.81% under route grouping and 74.81% for GRPO (Figure 4a). Route pooling combines tasks that share a coarse label but can differ in difficulty, available evidence, and outcome distribution. The observed 11.50-point difference favors preserving task identity in this setting.

The mixed-task study adds 500 distractor samples derived from GSM8K (Cobbe et al., 2021) to 2,000 multi-hop QA samples and uses five route labels. Task grouping again yields higher EM, 44.20% versus 42.50%, and higher grounding, 60.40% versus 54.70%. These results connect the Router to the compatibility requirement: a coarse semantic route does not automatically define the most informative normalization pool. Routing a task to a suitable workflow and pooling its reward baseline are separate design decisions.

Appendix D.1 reports additional credit configurations and their training dynamics.

5.5 Feedback Coverage and Local Degeneracy

In the replay environment, a call with no matching recorded response receives local reward zero, while a recorded successful call receives $+0.1$. Missing-response rates vary from 7.6–12.6% for query and metadata tools to 61.3–84.5% for the model-identifier tool chain. Thus the feedback available to the estimator depends strongly on which tools the policy explores.

Figure 5 relates this coverage to the trajectories generated during training. Rollouts touching the sparse chain have lower mean reward and lower terminal outcome match. Their prevalence falls from approximately 30% early in training to 2% later. A counter of adherence to the prescribed chain order falls from 146 to one, while repeated identical calls increase from 178 to 623. This pattern

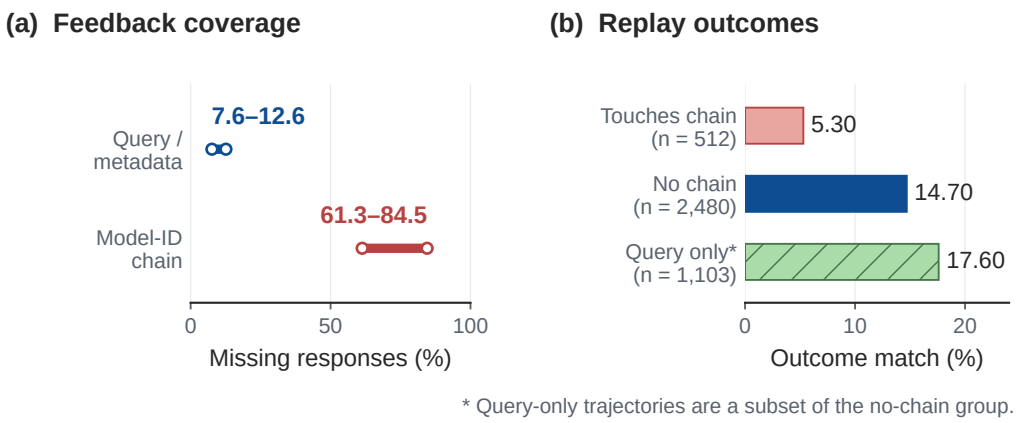

Figure 5: Uneven feedback coverage and replay outcomes. (a) Missing-response ranges across tool types in the query/metadata and model-ID families. The intervals represent across-tool ranges. (b) Terminal outcome match among 2,992 replay trajectories. The query-only group is a subset of the no-chain group. Table 3 also gives mean trajectory rewards.

is consistent with the relative penalty induced when unobserved tool outcomes compete against positively scored calls.

The replay outcome metric combines programmatic checks and semantic judging: 363 of 391 matched final answers are resolved by the semantic judge. These rates describe behavior under the replay library. The measurement establishes an association between feedback coverage and learned tool use, with tool choice and task difficulty also varying across the trajectory groups. It motivates distinguishing unavailable feedback from observed failure when constructing process returns.

5.6 Case Study: Decisions Behind SQL Execution Outcomes

We examine five cases from a completed 1,533-question BIRD-dev MAJ@4 evaluation of Qwen3.5-27B-DR.GRIPO, using historical Qwen3.5-27B-Base predictions as a cross-version reference. Correctness compares normalized SQLite results. Appendix F records all queries, outputs, and selected-group support.

Three cases expose data-selection errors. Using the charter-funding field in `frpm` instead of `schools` returns 751 rows matching gold, compared with 747 in the historical prediction. Filtering NULL subject scores changes the selected minimum from `(NULL, Fresno)` to `(314, Los Angeles)`. Ordering by `Price / Amount` replaces an empty unit-quantity query with the gold answer `CZE` (Figure 1).

Two cases concern answer structure. An explicit `RANK()` column changes a 50×2 result into the required 50×3 output. Six separate administrator-name fields match the gold layout. The selected gold-matching outputs receive two to four votes. These examples connect terminal execution match to decisions about schema, row eligibility, derived quantities, and answer representation, the process distinctions that motivate multi-granularity credit.

6 Related Work

Learning tool workflows. ToolLLM, Gorilla, and ToolAlpaca learn interfaces through instruction data, documentation retrieval, and simulated interactions (Qin et al., 2023; Patil et al., 2023; Tang et al., 2023). Agent Q adds search and preference optimization, Agent-R learns reflection from corrected traces, and WebRL adapts training using unsuccessful attempts (Putta et al., 2024; Yuan et al., 2025; Qi et al., 2024). Dr.Q targets credit construction within sampled workflows.

Credit estimation and process feedback. Policy gradients and advantage estimation address delayed outcomes (Williams, 1992; Schulman et al., 2015). RUDDER redistributes returns, while potential-based shaping preserves optimal policies under specified reward transformations (Arjona-Medina et al., 2019; Ng et al., 1999). RLOO and GRPO construct sampled baselines (Ahmadian et al., 2024; Shao et al., 2024); VinePPO estimates intermediate values through continuations (Kazemnejad et al., 2024). Math-Shepherd and OmegaPRM automate step supervision (Wang et al., 2023; Luo et al., 2024), PRIME learns implicit process rewards (Cui et al., 2025), and RaR specifies evaluation criteria (Gunjal et al., 2025). Our segment scores use execution rubrics and recovery feedback.

Structured agent credit. GiGPO compares repeated-state returns, SALT uses trajectory graphs, and Turn-PPO localizes advantages to turns (Feng et al., 2025; Li et al., 2026a;b). Agent Lightning separates execution from training, with initial LightningRL assigning final returns to transitions (Luo et al., 2025). Dr.Q combines task- or route-level outcomes, discounted segment returns, and terminal outcomes at recurring realized states. This separates the populations in which contrast and coverage are diagnosed, extending DAPO and RAGEN's reward-variance analyses (Yu et al., 2025; Wang et al., 2025).

Interactive evaluation. AgentBench tests reasoning and decisions across environments (Liu et al., 2023). WebShop, WebArena, Mind2Web, and WorkArena cover shopping, websites, and enterprise workflows (Yao et al., 2022; Zhou et al., 2023; Deng et al., 2023; Drouin et al., 2024). VisualWebArena and OSWorld add visual and desktop interaction (Koh et al., 2024; Xie et al., 2024). InterCode supplies execution feedback, SWE-bench tests repository repairs, and SWE-agent studies action interfaces (Yang et al., 2023; Jimenez et al., 2023; Yang et al., 2024). AppWorld checks cross-application state (Trivedi et al., 2024). These tasks connect intermediate decisions to final objectives.

Grounding and feedback availability. ALFWorld grounds plans in actions; GAIA combines reasoning, browsing, and tools (Shridhar et al., 2020; Mialon et al., 2023). HotpotQA supplies supporting facts, while Spider tests cross-schema generalization (Yang et al., 2018; Yu et al., 2018). ToolSandbox models stateful dependencies, τ^2-bench adds user coordination, and ToolEmu emulates tool execution (Lu et al., 2024; Barres et al., 2025; Ruan et al., 2023). Our replay analysis connects response availability to relative credit.

7 Discussion and Conclusion

Dr.Q combines task completion, process quality, and repeated work-state outcomes within one policy update. The 9B τ-bench comparison shows higher task success, and reinforcement learning improves answer accuracy along the 27B multi-hop QA training pipeline. Task grouping outperforms route pooling in both 4B configuration studies. In the replay environment, sparse feedback coverage accompanies a decline in exploration of the affected tools.

The current formulation uses heuristic routes, domain-specific rubric weights, and environment-defined segment boundaries. Anchor coverage depends on recurrent descriptors, and the main tool-agent setting is bounded at 20 steps. Extending to deeper nested workflows requires comparison groups and observable scores at each additional level. The design principle is to align the resolution of credit with the structure and feedback of the task: decomposition is useful when its local comparisons carry information.

AI USE STATEMENT

For the preparation of this revision, Codex was used to organize the scientific argument, draft and edit English prose, inspect supplied method and experiment records, check algebraic statements, verify bibliographic metadata, and prepare the LaTeX document. GPT Image 2 was used to produce exploratory method illustrations and a presentation poster. The manuscript figures are rendered as vectors with Python, and the numerical plots use the supplied experiment values. Experimental measurements were taken from pre-existing training and evaluation records. This statement covers the assistance used in the present revision.

REPRODUCIBILITY STATEMENT

Section 3 defines the comparison groups, reward terms, anchor construction, and policy objective. Appendix A gives the update algorithm, Appendix B proves the credit identities, and Appendix C describes the evaluation settings and supplementary measurements. The 2Wiki environment constructs its retrieval index from each question's context passages. Appendix F provides the five BIRD SQL pairs, execution outputs, and selected-result support. The tables distinguish held-out evaluation from training-window statistics and retain the configurations needed to interpret each comparison.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs. arXiv preprint arXiv:2402.14740, 2024. URL https://arxiv.org/abs/2402.14740.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return Decomposition for Delayed Rewards. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/16105fb9cc614fc29e1bda00dab60d41-Abstract.html.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating Conversational Agents in a Dual-Control Environment. arXiv preprint arXiv:2506.07982, 2025. URL https://arxiv.org/abs/2506.07982.

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

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process Reinforcement through Implicit Rewards. arXiv preprint arXiv:2502.01456, 2025. URL https://arxiv.org/abs/2502.01456.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang,

Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv preprint arXiv:2501.12948, 2025. URL https://arxiv.org/abs/2501.12948.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a Generalist Agent for the Web. arXiv preprint arXiv:2306.06070, 2023. URL https://arxiv.org/abs/2306.06070.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How Capable Are Web Agents at Solving Common Knowledge Work Tasks? arXiv preprint arXiv:2403.07718, 2024. URL https://arxiv.org/abs/2403.07718.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. arXiv preprint arXiv:2505.10978, 2025. URL https://arxiv.org/abs/2505.10978.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. arXiv preprint arXiv:2507.17746, 2025. URL https://arxiv.org/abs/2507.17746.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? arXiv preprint arXiv:2310.06770, 2023. URL https://arxiv.org/abs/2310.06770.

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. URL https://arxiv.org/abs/2503.09516.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining Credit Assignment in RL Training of LLMs. arXiv preprint arXiv:2410.01679, 2024. URL https://arxiv.org/abs/2410.01679.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks. arXiv preprint arXiv:2401.13649, 2024. URL https://arxiv.org/abs/2401.13649.

Jiazheng Li, Yawei Wang, Qiaojing Yan, Yijun Tian, Zhichao Xu, Huan Song, Panpan Xu, and Lin Lee Cheong. SALT: Step-level advantage assignment for long-horizon agents via trajectory graph. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 4709–4725. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.findings-eacl.247. URL https://aclanthology.org/2026.findings-eacl.247/.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? a BIg bench for large-scale database grounded Text-to-SQLs. arXiv preprint arXiv:2305.03111, 2023a. URL https://arxiv.org/abs/2305.03111.

Junbo Li, Peng Zhou, Rui Meng, Meet P. Vadera, Lihong Li, and Yang Li. Turn-PPO: Turn-level advantage estimation with PPO for improved multi-turn RL in agentic LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6227–6243. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.findings-eacl.328. URL https://aclanthology.org/2026.findings-eacl.328/.

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. arXiv preprint arXiv:2304.08244, 2023b. URL https://arxiv.org/abs/2304.08244.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. arXiv preprint arXiv:2305.20050, 2023. URL https://arxiv.org/abs/2305.20050.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as Agents. arXiv preprint arXiv:2308.03688, 2023. URL https://arxiv.org/abs/2308.03688.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A Stateful, Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities. arXiv preprint arXiv:2408.04682, 2024. URL https://arxiv.org/abs/2408.04682.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve Mathematical Reasoning in Language Models by Automated Process Supervision. arXiv preprint arXiv:2406.06592, 2024. URL https://arxiv.org/abs/2406.06592.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent Lightning: Train ANY AI Agents with Reinforcement Learning. arXiv preprint arXiv:2508.03680, 2025. URL https://arxiv.org/abs/2508.03680.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. arXiv preprint arXiv:2401.13178, 2024. URL https://arxiv.org/abs/2401.13178.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for General AI Assistants. arXiv preprint arXiv:2311.12983, 2023. URL https://arxiv.org/abs/2311.12983.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999. URL https://people.eecs.berkeley.edu/~russell/papers/icml99-shaping.pdf.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large Language Model Connected with Massive APIs. arXiv preprint arXiv:2305.15334, 2023. URL https://arxiv.org/abs/2305.15334.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent Q: Advanced Reasoning and Learning for Autonomous AI Agents. arXiv preprint arXiv:2408.07199, 2024. URL https://arxiv.org/abs/2408.07199.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, Tianjie Zhang, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM Web Agents via Self-Evolving Online Curriculum Reinforcement Learning. arXiv preprint arXiv:2411.02337, 2024. URL https://arxiv.org/abs/2411.02337.

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. URL https://arxiv.org/abs/2307.16789.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the Risks of LM Agents with an LM-Emulated Sandbox. arXiv preprint arXiv:2309.15817, 2023. URL https://arxiv.org/abs/2309.15817.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. arXiv preprint arXiv:2302.04761, 2023. URL https://arxiv.org/abs/2302.04761.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. arXiv preprint arXiv:1506.02438, 2015. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

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

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*, pp. 1279–1297, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. arXiv preprint arXiv:2010.03768, 2020. URL https://arxiv.org/abs/2010.03768.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. ToolAlpaca: Generalized Tool Learning for Language Models with 3000 Simulated Cases. arXiv preprint arXiv:2306.05301, 2023. URL https://arxiv.org/abs/2306.05301.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents. arXiv preprint arXiv:2407.18901, 2024. URL https://arxiv.org/abs/2407.18901.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. arXiv preprint arXiv:2211.14275, 2022. URL https://arxiv.org/abs/2211.14275.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. arXiv preprint arXiv:2312.08935, 2023. URL https://arxiv.org/abs/2312.08935.

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, 2025. URL https://arxiv.org/abs/2504.20073.

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

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. arXiv preprint arXiv:2404.07972, 2024. URL https://arxiv.org/abs/2404.07972.

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. InterCode: Standardizing and Benchmarking Interactive Coding with Execution Feedback. arXiv preprint arXiv:2306.14898, 2023. URL https://arxiv.org/abs/2306.14898.

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. arXiv preprint arXiv:2405.15793, 2024. URL https://arxiv.org/abs/2405.15793.

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. arXiv preprint arXiv:1809.09600, 2018. URL https://arxiv.org/abs/1809.09600.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. arXiv preprint arXiv:2207.01206, 2022. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

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. URL https://arxiv.org/abs/2406.12045.

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. arXiv preprint arXiv:2503.14476, 2025. URL https://arxiv.org/abs/2503.14476.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. arXiv preprint arXiv:1809.08887, 2018. URL https://arxiv.org/abs/1809.08887.

Siyu Yuan, Zehui Chen, Zhiheng Xi, Junjie Ye, Zhengyin Du, and Jiecao Chen. Agent-R: Training Language Model Agents to Reflect via Iterative Self-Training. arXiv preprint arXiv:2501.11425, 2025. URL https://arxiv.org/abs/2501.11425.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A Realistic Web Environment for Building Autonomous Agents. arXiv preprint arXiv:2307.13854, 2023. URL https://arxiv.org/abs/2307.13854.

A Training Algorithm and Implementation

Algorithm 1 Dr.Q update for a batch of tool trajectories

Require: Policy π_θ, task batch, route and grouping rules, rubric, gate, work-state descriptor, role schedule

Sample candidate rollouts for each task under the selected workflow
Partition each rollout into generated-action segments and environment observations
Recover terminal outcome R_i from the final segments of each rollout
for each generated segment $\sigma_{i,j}$ **do**
    Compute process reward $r_{i,j}$ and work-state key $k_{i,j} = (u_i, h_{i,j})$
end for
Accumulate $G_{i,j} = r_{i,j} + \gamma G_{i,j+1}$ in reverse segment order
Center terminal outcomes within macro comparison groups (Equation 2)
Center process returns within task–segment groups, with task-mean fallback (Equation 6)
for each anchor key k **do**
    Deduplicate member rollout identifiers to form $\mathcal{D}_k$
    Compute leave-one-rollout-out outcome differences, or zero for a singleton
end for
Apply anchor support weights, clipping, position weights, and overlong weighting
Construct token advantages with Equation 10
Select active-role generated tokens; mask environment observations
Update policy parameters with the clipped surrogate in Equation 11

The estimator operates on segment records produced by the agent loop. A record identifies the task, rollout, segment position, role, process reward, terminal status, and work-state descriptor. The final flag supplies the rollout outcome to the macro and anchor branches. Observation-only records receive no policy gradient. The same terminal outcome can be broadcast to several action segments for advantage construction without counting repeated visits as independent anchor evidence.

For tool workflows, descriptors normalize the operation and its outcome. SQL-oriented descriptors can include a query fingerprint and an execution class. Other tool descriptors can include a normalized tool name, argument signature, success status, and work-state category. They are attached after the environment call. The key is (u, h); role-specific namespaces in h, such as a diagnostic prefix or a prompt-mode field, determine which roles share an anchor pool. Prompt-mode changes inside a shared context and context-isolated sub-loops can therefore define different comparison populations.

The segment-return recursion consumes the per-segment process-reward stream. A terminal event may receive a process reward through this stream. Its separately recorded rollout outcome enters the macro and anchor branches, without being appended again to the segment-return recursion.

Deduplication is essential for a leave-one-rollout-out comparison. Otherwise, a rollout revisiting the same descriptor can contribute its own terminal outcome through another visit and inflate the apparent group support. Distinct-rollout aggregation yields one sufficient statistic per rollout and anchor key. For S segment records and K distinct anchor memberships, task and segment aggregation take $O(S)$ expected time using hash maps, and anchor statistics take $O(K)$ expected time. This bookkeeping follows trajectory generation and adds no learned value network.

The position factors distinguish start, intermediate, and final segments. A segment tagged as final takes precedence over a start tag, including a rollout with a single segment. Grouping rules, standard-deviation normalization, length weighting, and role schedules are set in the training configuration.

B Credit Identities and Prefix Dependence

B.1 Group-Centered Credit

For m scores $y_1, \ldots, y_m$, empirical variance is defined with divisor m:

$$\mathrm{Var}_{\mathcal{C}}(y) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y})^2. \tag{14}$$

Multiplication by m proves Equation 12. Every summand is nonnegative, so the sum is zero exactly when every score equals the group mean. If αm scores equal c and the rest equal c', the mean is $\alpha c + (1-\alpha)c'$. A member at c therefore has centered score $(1-\alpha)(c-c')$, which is negative when $c < c'$ and $\alpha < 1$.

For $m \geq 2$, the leave-one-out difference obeys

$$y_i - \frac{1}{m-1}\sum_{j\neq i} y_j = \frac{m}{m-1}(y_i - \bar{y}). \tag{15}$$

It has the same zero-signal condition and sign as ordinary centering. Applied to the anchor branch, the scores are terminal outcomes of distinct rollouts. Applied to the segment branch, the scores are discounted process returns. A constant immediate process reward can coexist with variable discounted returns when later rewards differ.

These identities concern numerical credit. A nonzero advantage does not by itself establish a nonzero expected parameter gradient: score functions, masking, cancellation across samples, and the other branches also enter the update. Conversely, a zero local branch leaves the macro and other local branches available. This distinction separates a measurable signal property from a claim about the full training dynamics.

B.2 Prefix-Conditional Outcomes

Let k be a work-state descriptor and c_i retain information about the history of rollout i. Suppose the other distinct rollouts in its anchor pool have expected outcome $\mu_k = \mathbb{E}[R \mid k]$ and their mean is conditionally independent of c_i given the pool definition. Linearity gives

$$\mathbb{E}[R_i - b_k^{-i} \mid k, c_i] = \mathbb{E}[R_i \mid k, c_i] - \mathbb{E}[b_k^{-i} \mid k, c_i] \tag{16}$$

$$= \mathbb{E}[R_i \mid k, c_i] - \mu_k. \tag{17}$$

Thus histories with different conditional outcome distributions can receive different anchor scores despite sharing the descriptor. Task conditioning removes cross-task variation from this comparison, while descriptor design determines which within-task information is retained.

An action-independent baseline can preserve an expected policy gradient even when it differs from a conditional value function. Here the realized descriptor can contain the chosen action and its observation, and the estimator also includes normalization, clipping, and position weights. We consequently define the training procedure by its composed surrogate objective. The prefix identity explains the contents of its anchor score without requiring a counterfactual interpretation of the group members.

C Experimental Protocols

C.1 τ-bench

The main comparison uses the retail and airline domains and reports the task-success metric of τ-bench (Yao et al., 2024). The two domain scores receive equal weight in the reported average. SFT, GRPO, and Dr.Q rows use the 9B model family. The 27B Dr.Q evaluation reaches 49.57% retail success, 50.00% airline success, and 49.79% average. Table 1 compares training methods at 9B.

Table 1: τ-bench task success (%) for Qwen3.5-9B. Average is the unweighted mean of retail and airline success. Dr.Q improves both domains, with the larger gain on retail.

Training	Retail	Airline	Average
SFT	15.65	36.00	25.83
GRPO	19.13	36.00	27.57
Dr.Q	**36.52**	**44.00**	**40.26**

C.2 2WIKIMULTIHOPQA

The environment builds a BM25 index from the context passages accompanying each question. Its four tools are `search_entity`, `get_entity_page`, `retrieve_passage`, and `submit_answer`. The 27B checkpoints share a weight lineage from CPT to cold-start SFT and then 100 steps of joint reinforcement learning with rollout group size four. All three checkpoints are evaluated on the same 200 development questions with temperature zero.

The 4B configuration study uses 100 training steps, rollout group size eight, and 6,400 rollouts per configuration. Its last-quarter metrics average over steps 76–100. The mixed-task grouping comparison uses 2,500 samples, comprising 2,000 2Wiki questions and 500 GSM8K-derived distractor samples, with five route labels. The task-grouped and route-grouped runs use the same mixed dataset.

Answer EM and F1 evaluate the submitted answer. Execution match tracks successful task completion through the tool protocol. Grounding evaluates support in retrieved evidence. The parse-failure rate is the percentage of questions with at least one invalid tool-protocol output. Table 2 reports each measure separately.

Table 2: 27B results on the same 200-question 2WikiMultihopQA development subset with matched harness and greedy decoding. Rates and F1 are on a 0–100 scale. Reinforcement learning improves answer and execution matching while eliminating observed protocol parse failures.

Metric	CPT	SFT	SFT + DR.GRIPO
Answer EM ↑	58.00	62.50	**69.00**
Answer F1 ↑	63.05	68.75	**74.90**
Execution match ↑	51.00	60.50	**67.50**
Grounded ↑	75.00	**85.00**	83.00
Parse failures ↓	51.00	28.50	**0.00**
Mean turns	5.83	6.03	5.41

C.3 TOOL-RESPONSE REPLAY

The replay environment returns recorded tool responses for matched calls. The analysis includes query and metadata tools as well as a model-identifier chain. Unmatched calls receive process reward zero; recorded successful calls receive $+0.1$. Across trainable calls, 20.4% have no recorded response. The 2,992-rollout analysis groups trajectories by whether they use the sparsely covered chain. Its terminal matching procedure includes semantic judging, as specified in Section 5.5.

Table 3: Replay trajectories grouped by whether they touch a sparsely covered tool chain. Outcome match is the replay environment's terminal metric. The query-only row is a subset of trajectories that do not touch the chain.

Trajectory group	Rollouts	Mean reward	Outcome match (%)
Touches sparse chain	512	0.253	5.30
Does not touch chain	2,480	0.392	14.70
Query tools only	1,103	0.420	17.60

Table 4: Per-type answer EM (%) on the 200-question 27B development evaluation. Counts sum to 200. Compositional questions show the largest post-SFT gain; inference accuracy remains unchanged.

Question type	Count	CPT	SFT	SFT + DR.GRIPO
Bridge comparison	41	85.37	87.81	**92.68**
Comparison	46	82.61	86.96	**91.30**
Compositional	79	31.65	41.77	**53.16**
Inference	34	**52.94**	47.06	47.06

D Additional Multi-Hop QA Results

The transition from SFT to reinforcement learning improves bridge-comparison EM by 4.87 points, comparison EM by 4.34 points, and compositional EM by 11.39 points. Inference EM remains at 47.06%. The aggregate improvement is therefore concentrated in task types where the retrieval-and-execution workflow can convert additional progress into an exact answer.

D.1 Credit Configurations

Table 5: Grouping configurations on 2WikiMultihopQA. The 4B single-task-distribution study reports EM over the last 25 training steps. The mixed-task study uses 2,000 2Wiki samples and 500 GSM8K-derived distractor samples. Task grouping retains task identity; route grouping pools tasks within each route label.

Setting	Grouping / estimator	EM (%)	Grounded (%)
2Wiki	Task grouping, Dr.Q	**77.31**	**88.38**
2Wiki	Route grouping, Dr.Q	65.81	73.19
2Wiki	GRPO	74.81	88.25
Mixed tasks	Task grouping	**44.20**	**60.40**
Mixed tasks	Route grouping	42.50	54.70

Table 6: Credit configurations for the 4B model on 2WikiMultihopQA. Values are last-25-step training metrics (%). Task and route grouping are identified explicitly for the corresponding reference runs. The gate-disabled configuration retains per-turn rewards.

Configuration	EM	F1	Grounded	Exec. match
Dr.Q, task grouping	**77.31**	**79.28**	**88.38**	**76.50**
Dr.Q, route grouping	65.81	67.75	73.19	60.69
Gate disabled	70.50	72.01	80.94	68.31
Anchor disabled	63.50	65.11	67.50	60.25
Segment-return term disabled	67.75	68.92	75.06	66.69
Position weighting disabled	70.94	72.57	80.44	67.88
GRPO	74.81	76.63	88.25	73.81

Table 6 compares the credit configurations. The gate-disabled setting retains per-turn process rewards. Task grouping reaches 77.31% EM, while route grouping reaches 65.81%, so interpreting a local-credit change requires specifying the macro comparison setting. Table 5 compares these grouping choices directly.

Table 7: Training EM (%) by quarter for selected 4B runs. The task-grouped configuration leads in the last two quarters.

Configuration	1–25	26–50	51–75	76–100
Dr.Q, task grouping	39.81	64.81	72.56	**77.31**
Dr.Q, route grouping	40.44	59.06	64.44	65.81
Anchor disabled	40.25	55.50	62.56	63.50
Segment-return term disabled	39.19	57.44	62.31	67.75
Position weighting disabled	36.06	55.88	64.56	70.94
GRPO	38.25	**66.31**	70.00	74.81

E Additional Task and Scale Evaluations

E.1 Mathematical Reasoning

Table 8: GSM8K exact-match accuracy (%) for the earlier direct-generation variant. Training uses 180 steps at 1.7B and 320 steps at 9B and 27B.

Estimator	1.7B	9B	27B
GRPO	59.9	**78.5**	79.8
Dr.Q	**72.0**	78.3	**80.1**

GSM8K evaluates mathematical word-problem solving (Cobbe et al., 2021). These experiments use the earlier direct-generation DR.GRIPO variant with Qwen3-1.7B, Qwen3.5-9B, and Qwen3.5-27B. This variant applies credit and position weighting within a generated response. Table 8 shows a 12.1-point advantage over GRPO at 1.7B and differences of -0.2 and $+0.3$ points at 9B and 27B. The 1.7B responses average 321 tokens under DR.GRIPO and 469 under GRPO.

A separate 1.7B two-turn self-check setting reaches training score 0.596 with DR.GRIPO versus 0.440 with GRPO, but validation exact match is 43.0% versus 50.0%. The respective response lengths are 512 and 504 tokens, with clipping ratios 1.00 and 0.75. This setting illustrates how improvements in the training objective can coexist with poorer validation accuracy when the interaction and response budget change.

E.2 Database Execution

In the BIRD database environment (Li et al., 2023a), the segmented agent performs schema exploration, candidate execution, diagnostics, and final SQL submission against a live database. A 240-step training run contains 3,120 rollouts with 53.8% aggregate execution match. Training execution match increases from 51.3% over steps 51–100 to 57.5% over steps 151–200.

F BIRD EXECUTION CASE STUDY

The five cases come from a completed 1,533-question Qwen3.5-27B-GRIPO evaluation on BIRD-dev (Li et al., 2023a). Historical Qwen3.5-27B-Base predictions provide the reference. This is a cross-version comparison: the historical model is a different checkpoint from the immediate initialization of the trained model. MAJ@4 selects an execution-result group from four generated candidates, and correctness is equality with the normalized gold SQLite result. The counts below give the number of candidates sharing the selected result.

Case selection and execution records. These five examples were selected for a historical mismatch and a gold-matching GRIPO result. Together they cover schema grounding, returned columns, NULL handling, and unit-price semantics. Their SQL, output dimensions, displayed rows, group support, and result-hash prefixes are taken from the supplied case records. A support of $2/4$ specifies two matching candidates; the full vote partition and tie-breaking rule are not recorded. Table 9 summarizes the cases, and Table 10 retains their result identifiers.

Table 9: Selected BIRD-dev cases. Every selected GRIPO result matches gold. Support counts candidates with the selected normalized execution result out of four. Output sizes are rows $\times$ columns.

ID	Decision	Historical Base	GRIPO selected	Support
4	Funding column	747×1	751×1	2/4
17	Ranking column	50×2	50×3	2/4
36	Name-field layout	1×3; concatenated	1×6; separate	3/4
43	NULL filtering	`(NULL, Fresno)`	`(314, Los Angeles)`	3/4
1532	Unit-price order	Empty result	`CZE`	4/4

Table 10: Reported SHA-256 prefixes over the execution result rows. The selected GRIPO prefix equals the gold prefix in each supplied record.

ID	Historical Base	GRIPO selected / gold
4	`c526aa83ead065ec`	`6bf00d40a0443eba`
17	`475f4916398588c8`	`eb74bf65157d6b8a`
36	`20bd3575759dcb17`	`8199ae328ac30f77`
43	`958d9c52ffe84d6c`	`a0e84eeb5e80c264`
1532	`4f53cda18c2baa0c`	`36998a02faf0316a`

F.1 Ground charter funding in the correct table

Question 4. List phone numbers of directly charter-funded schools opened after 2000-01-01. Database: `california_schools`.

Both queries join schools and frpm by CDSCode, require charter-school status, and apply the same opening-date threshold. The decisive predicate uses schools.FundingType in the historical prediction and frpm.Charter Funding Type in the selected GRIPO prediction. The latter returns 751 phone-number rows, matching gold, compared with 747 for the historical prediction. The identical displayed head does not reveal the difference in the full result.

Historical Base.

```
SELECT s.Phone FROM schools s JOIN frpm f ON s.CDSCode = f.CDSCode
    WHERE f."Charter School (Y/N)" = 1 AND s.FundingType = 'Directly
    funded' AND s.OpenDate > '2000-01-01'
```

747 rows × 1 column. Displayed head: `[(NULL,), ('(510) 596-8901',), (NULL,)]`.

GRIPO, selected execution group: 2/4.

```
SELECT schools.Phone
FROM schools
JOIN frpm ON schools.CDSCode = frpm.CDSCode
WHERE frpm.`Charter School (Y/N)` = 1
  AND frpm.`Charter Funding Type` = 'Directly funded'
  AND schools.OpenDate > '2000-01-01'
```

751 rows × 1 column. Displayed head: `[(NULL,), ('(510) 596-8901',), (NULL,)]`. The normalized result matches gold.

F.2 Return the requested ranking column

Question 17. Rank schools by Writing score above 499 and show their charter numbers. Database: `california_schools`.

Ordering the rows by writing score leaves the requested rank implicit. GRIPO includes a RANK window expression in the projection, making rank a returned value. Both results contain 50 rows; the selected prediction has the three columns required by gold, including ranks 1, 2, and 3 for the displayed schools.

Historical Base.

```
SELECT s.CharterNum, sa.AvgScrWrite FROM schools s JOIN satscores sa ON
    s.CDSCode = sa.cds WHERE sa.AvgScrWrite > 499 AND s.CharterNum IS
    NOT NULL ORDER BY sa.AvgScrWrite DESC
```

50 rows × 2 columns. Displayed head: `[('0210', 630), ('0890', 593), ('0290', 582)]`.

GRIPO, selected execution group: 2/4.

```
SELECT
    s.CharterNum,
    sa.AvgScrWrite,
    RANK() OVER (ORDER BY sa.AvgScrWrite DESC) AS writing_score_rank
FROM schools s
JOIN satscores sa ON s.CDSCode = sa.cds
WHERE sa.AvgScrWrite > 499
  AND s.CharterNum IS NOT NULL
ORDER BY writing_score_rank
```

50 rows × 3 columns. Displayed head: `[('0210', 630, 1), ('0890', 593, 2), ('0290', 582, 3)]`. The normalized result matches gold.

F.3 PRESERVE THE GOLD OUTPUT FIELDS FOR ADMINISTRATOR NAMES

Question 36. Return the full names of administrators of the school with the highest NumGE1500. Database: `california_schools`.

The historical prediction concatenates each first and last name, producing three administrator columns. GRIPO returns the six source fields separately, matching the benchmark output layout. The displayed first administrator is Michelle King in both results. This example locates the execution-match difference in field representation.

Historical Base.

```
SELECT s.AdmFName1 || ' ' || s.AdmLName1 AS Admin1, s.AdmFName2 || ' '
    || s.AdmLName2 AS Admin2, s.AdmFName3 || ' ' || s.AdmLName3 AS
    Admin3 FROM schools s JOIN satscores sa ON s.CDSCode = sa.cds ORDER
    BY sa.NumGE1500 DESC LIMIT 1
```

1 row × 3 columns. Result rows: `[('Michelle King', NULL, NULL)]`.

GRIPO, selected execution group: 3/4.

```
SELECT
    s.AdmFName1, s.AdmLName1,
    s.AdmFName2, s.AdmLName2,
    s.AdmFName3, s.AdmLName3
FROM schools s
JOIN satscores sc ON s.CDSCode = sc.cds
ORDER BY sc.NumGE1500 DESC
LIMIT 1
```

1 row × 6 columns. Result rows: `[('Michelle', 'King', NULL, NULL, NULL, NULL)]`. The normalized result matches gold.

F.4 FILTER NULL SCORES BEFORE SELECTING A MINIMUM

Question 43. Find the school with the lowest sum of Math, Reading, and Writing averages, returning Math score and county. Database: `california_schools`.

In SQLite, a sum containing NULL evaluates to NULL, and NULL sorts before numeric values in ascending order. The historical query therefore admits an incomplete score row and returns (NULL, Fresno). GRIPO requires all three subject scores to be non-NULL before ordering their sum. Its result is (314, Los Angeles), matching gold. The predicates define which rows are eligible for the requested minimum.

Historical Base.

```
SELECT s.AvgScrMath, sc.County FROM satscores s JOIN schools sc ON
    s.cds = sc.CDSCode ORDER BY (s.AvgScrMath + s.AvgScrRead +
    s.AvgScrWrite) ASC LIMIT 1
```

1 row × 2 columns. Result rows: `[(NULL, 'Fresno')]`.

GRIPO, selected execution group: 3/4.

```
SELECT satscores.AvgScrMath, schools.County
FROM satscores
JOIN schools ON satscores.cds = schools.CDSCode
WHERE satscores.AvgScrMath IS NOT NULL
  AND satscores.AvgScrRead IS NOT NULL
  AND satscores.AvgScrWrite IS NOT NULL
ORDER BY satscores.AvgScrMath + satscores.AvgScrRead +
    satscores.AvgScrWrite ASC
LIMIT 1
```

1 row × 2 columns. Result rows: `[(314, 'Los Angeles')]`. The normalized result matches gold.

F.5 ORDER BY UNIT PRICE OVER THE ELIGIBLE TRANSACTIONS

Question 1532. Which country had the gas station selling the most expensive product ID 2 per unit? Database: `debit_card_specializing`.

The historical query interprets per unit as an Amount = 1 filter, which yields an empty result. GRIPO keeps the ProductID = 2 constraint and orders the transactions by Price / Amount. The selected result is CZE, matching gold, and all four sampled candidates share that normalized execution result. The change expresses unit price as a derived quantity over transactions of any amount.

Historical Base.

```
SELECT g.Country FROM transactions_1k t JOIN gasstations g ON
    t.GasStationID = g.GasStationID WHERE t.ProductID = 2 AND t.Amount
    = 1 ORDER BY t.Price DESC LIMIT 1
```

Empty result (0 rows). Result rows: `[]`.

GRIPO, selected execution group: 4/4.

```
SELECT country
FROM gasstations AS T1
INNER JOIN transactions_1k AS T2
    ON T1.GasStationID = T2.GasStationID
WHERE T2.ProductID = 2
ORDER BY T2.Price / T2.Amount DESC
LIMIT 1
```

1 row × 1 column. Result rows: `[('CZE',)]`. The normalized result matches gold.

What the cases reveal. The funding, NULL, and unit-price cases change which database rows answer the question. The ranking and administrator cases change the representation of the returned answer. This separates data-selection errors from output-structure errors within an execution-match failure. For multi-granularity credit, the distinction identifies concrete decisions whose consequences must remain visible alongside the terminal score.