biCurriculum: Crediting Executable Task and Environment Rewrites by Their Learning Utility

Anonymous authors
Paper under double-blind review

Abstract

A failed agent trajectory identifies where a policy breaks, but not which training intervention will improve the unchanged task. We introduce **biCurriculum**, a framework that turns failures into executable task or environment rewrites and credits them by downstream learning utility relative to matched original-task replay. The framework separates native rewrite validity, solver reachability on the child, and original-task improvement after training. On complete WebArena-Lite165, a Qwen3.5-9B rewritten-data arm solves 44/165 tasks versus 34/165 for a replay control and 29/165 for the starting solver; the paired run has 13 rewrite-only versus 3 replay-only successes (exact McNemar $p = .021$). Across three completed training runs, the rewritten-data arm solves 43–44 tasks while replay solves 33–35, a mean gain of 5.9 percentage points. Controlled ALFWorld results are heterogeneous across the pre-specified matched batches: two matched batches score 107/128 versus 118/128 and 108/128 versus 110/128, while a third reaches 120/128 versus 109/128 (13 versus 2 discordant successes, $p = .0074$). A diagnostic audit shows that only 2/10 originally trained children exercise the parent failure step. On the same three B1 training parents, a failure-step-preserving correction raises native sampled success from 2/12 for the original curriculum to 12/12 (start 5/12; replay 10/12). Together, the results show both when executable rewrites can beat replay and why valid, reachable rewrites can fail: the child must expose the capability missing on the parent.

1 Introduction

Training interactive agents from failure is a data-design problem. A failed rollout reveals where the current policy is insufficient, but a useful intervention must do more than create an easier task: after learning from the intervention, the agent should improve on the unchanged task. Existing work adapts curricula, goals, or environments using difficulty, regret, success rate, learnability, or fresh-rollout feedback (Florensa et al., 2018; Dennis et al., 2020; Jiang et al., 2021; Guo et al., 2025; Qi et al., 2025; Huang et al., 2026a). We ask a narrower causal question: *does training on an executable rewrite improve the original task beyond a matched original-task replay update?*

biCurriculum (Figure 1) represents a curriculum item as an executable edit program: a task rewrite, an environment rewrite, or both, together with a restoration path to the parent. Each child is compiled into an isolated native instance whose verifier matches its actual objective. Solver training uses only execution-verified interactions, and evaluation always returns to unmodified tasks. For a fixed failed parent, the rewriter receives positive credit only when a solver trained on the child improves over both the starting solver and a matched replay copy.

Our contributions are threefold. First, we formulate failure-conditioned curriculum construction as executable rewriting with explicit restoration and a learning-utility credit signal. Second, we give a verification protocol that separates *validity*, *reachability*, and *learning utility*, including matched-copy controls and native outcome checks. Third, we provide evidence that the utility of rewritten training data is heterogeneous and must be measured after learning, rather than inferred from child validity or success. On WebArena-Lite, the rewritten-data arm improves over replay in all three completed training runs. On ALFWorld, three pre-specified matched batches yield a clear loss, a near tie, and a

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.

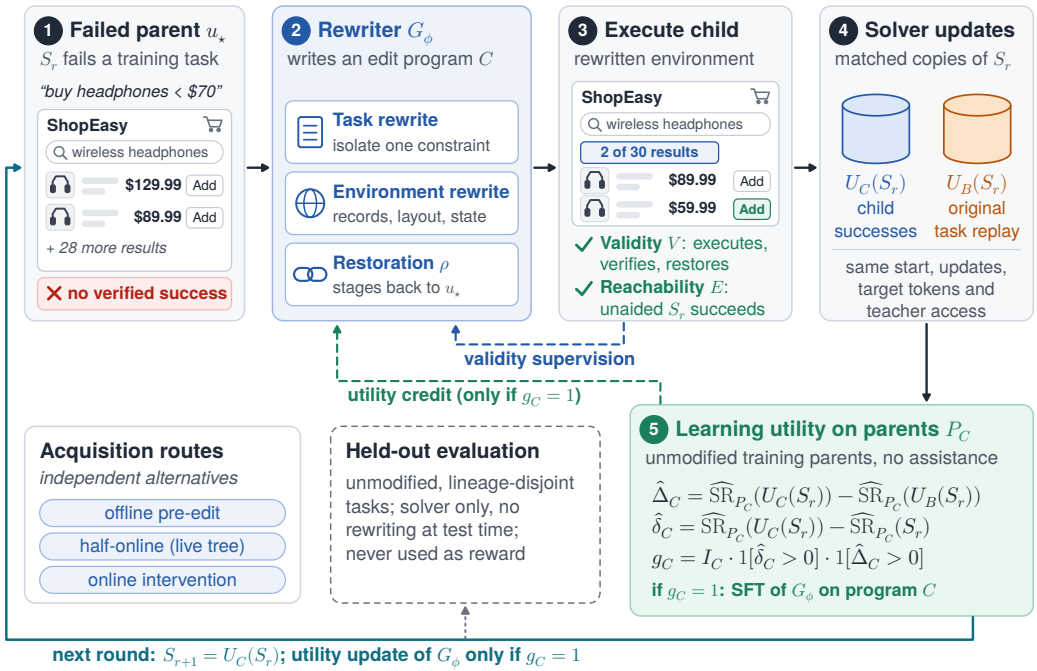


Figure 1: **biCurriculum.** A rewriter proposes an executable edit program for a failed parent. Native checks establish validity and unaided child success establishes reachability. Two solver copies receive matched updates from rewritten successes or original-task replay. Rewriter credit is based on the change on unmodified parents, not on child success itself.

clear gain relative to replay. A trajectory audit identifies failure-step coverage as one concrete source of this variation, and a targeted failure-step-preserving correction raises sampled native success on the same B1 training parents from 2/12 to 12/12.

2 Related Work

Automatic curricula and environment design. Curricula order training data by difficulty (Bengio et al., 2009), select tasks by learning progress (Graves et al., 2017; Matiisen et al., 2020), or generate goals of intermediate difficulty (Florensa et al., 2018; Sukhbaatar et al., 2018). Reverse and start-state curricula begin episodes near the goal and move the start state backward, so every episode still ends with the task's final steps (Florensa et al., 2017; Resnick et al., 2018; Salimans & Chen, 2018). Unsupervised environment design generates, replays or edits levels to maximize regret or learning potential (Wang et al., 2019; Dennis et al., 2020; Jiang et al., 2021; Parker-Holder et al., 2022; Chung et al., 2024), and LLM-based generators adapt environments to an agent's weaknesses (Zala et al., 2024; Yeo et al., 2026).

Task and environment generation for LLM agents. GenEnv co-trains a solver and a generator with a target success band (Guo et al., 2025); ACuRL, RLAnything, Environment Evolution, SEAL and SENTINEL adapt tasks or environments to observed agent performance and failures (Xue et al., 2026; Wang et al., 2026b; Fan et al., 2026; Hu et al., 2026; Wang et al., 2026c). VeriEnv and related work build verified synthetic web environments (Chae et al., 2026; Zhang et al., 2026a); InfiniteWeb, GUI-Genesis and AgentSynth synthesize environments or tasks at scale, and OMNI-EPIC writes environments as code (Zhang et al., 2026b; Cao et al., 2026; Xie et al., 2026; Faldor et al., 2025); Learn-by-Interact, proposer–evaluator and self-challenging pipelines let agents practice self-generated tasks (Su et al., 2025; Zhou et al., 2025b;a). Self-play for reasoning pairs a proposer with a solver and rewards learnable problems (Zhao et al., 2025; Huang et al., 2026b), and learnability-based selection prioritizes training prompts (Foster et al., 2025). EnvHarness is closest in mechanism: an LLM writes

components that reshape a static environment for a diagnosed weakness and validates them with fresh rollouts (Huang et al., 2026a). These methods admit or weight tasks by validity, solvability, difficulty, learnability or success under fresh rollouts, measured *before* a solver update. Our utility signal is measured *after* an actual solver update, on the unchanged parent, against a same-start replay update; the claimed distinction is this credit-assignment experiment, not environment generation by itself.

Learning from failures, own successes and data value. Hindsight relabeling converts failures into successes for substituted goals (Andrychowicz et al., 2017); WebRL generates new instructions from failed ones (Qi et al., 2025); and self-improvement methods retrain on filtered own successes (Oh et al., 2018; Zelikman et al., 2022; Dong et al., 2023; Singh et al., 2024). biCurriculum relabels at the instance level—goal, environment and verifier change jointly—and never counts a child success as parent success; replay of verified successes serves as its control, and progressive suffix replay and conversational feedback learning are further alternatives (Zhang & Yang, 2026; Wang et al., 2026a). Learning to teach, data valuation and teacher–student data generation credit data by the student's improvement after training (Fan et al., 2018; Ghorbani & Zou, 2019; Khan et al., 2025; Li et al., 2025); our $\widehat{\Delta}_C$ is a retraining-based analogue of influence and data valuation (Koh & Liang, 2017; Yoon et al., 2020; Xia et al., 2024), measured by native task success. Web agents are also trained by curriculum and multi-turn RL or on synthetic trajectories (Wei et al., 2025; Murty et al., 2024; Sun et al., 2025), or improved at inference time (Yang et al., 2025b); we follow the WebArena(-Lite) and ALFWorld protocols (Zhou et al., 2024; Liu et al., 2025; Shridhar et al., 2021).

3 Problem Setup

An interactive instance is $u = (x, e, v, i)$, comprising task semantics x, executable environment and data e, verifier v, and initial state i. For a failed training parent $u_{j,\star}$, rewriter G_ϕ emits a chain

$$C_j = (u_{j,0}, \ldots, u_{j,L_j}; \mathcal{E}_j, \rho_j), \qquad u_{j,L_j} = u_{j,\star}, \tag{1}$$

where $\mathcal{E}_j$ contains executable edits and ρ_j restores the parent. The present experiments use $L_j = 1$: one child followed by the unchanged parent.

We distinguish three quantities. **Validity** $V_{C,j}$ requires compilation against native interfaces, a native positive witness, registered negative controls, and restoration to the parent's identities. **Reachability** $E_{C,j}$ requires at least one unaided solver success on the child under the declared probe budget. **Learning utility** is a property of an update, not an instance. Starting from the same solver checkpoint S_r, we train a curriculum copy $U_C(S_r)$ on verified child successes and a replay copy $U_B(S_r)$ on verified successes from unmodified training tasks. On parent j,

$$\widehat{\Delta}_{C,j} = \widehat{\mathrm{SR}}_j(U_C(S_r)) - \widehat{\mathrm{SR}}_j(U_B(S_r)), \qquad \widehat{\delta}_{C,j} = \widehat{\mathrm{SR}}_j(U_C(S_r)) - \widehat{\mathrm{SR}}_j(S_r). \tag{2}$$

A positive rewriter target must satisfy native validity and reachability and have both gains positive. Held-out evaluation is never used as rewriter reward.

4 Method

Executable rewrite and verification. The rewriter receives the parent, a summary of the solver failure, the supported edit vocabulary, and a budget. A task edit may remove or isolate a requirement; an environment edit may change initial state, records, layout, or workflow. The compiler binds the program to native task interfaces and instantiates a child verifier consistent with the changed objective. Validity checks include the successful witness, no-op or skipped-step controls when applicable, child-witness replay on the unchanged parent, and restoration of the parent goal, state, verifier, and action mapping. Auxiliary model judgments are not substituted for native outcomes.

Solver update. Let D be supervised decisions from complete execution-verified trajectories plus shared retention anchors. Prompts and observations are masked; loss is applied only to executed action and termination tokens:

$$\mathcal{L}_S(\theta) = -\frac{1}{|D|} \sum_{d \in D} \frac{1}{|M_d|} \sum_{k \in M_d} \log S_\theta(y_{d,k} \mid h_d, y_{d,<k}). \tag{3}$$

Algorithm 1 One biCurriculum round (offline acquisition)

Require: solver S_r, rewriter G_ϕ, failed parents P_C, replay pool B, retention anchors A
1: propose edit chains $C \sim G_\phi(\cdot \mid P_C, \text{failures})$ and compile them natively
2: check validity $V_{C,j}$ and unaided child reachability $E_{C,j}$; collect verified child successes D_C
3: train matched copies $U_C(S_r) \leftarrow \textsc{Update}(S_r; D_C \cup A)$ and $U_B(S_r) \leftarrow \textsc{Update}(S_r; B \cup A)$
4: evaluate $S_r, U_C(S_r), U_B(S_r)$ on unmodified targets and compute Eq. 2
5: update G_ϕ only on chains with $g_{C,j} = 1$ in Eq. 4; log rejected chains and costs
6: **return** $S_{r+1} \leftarrow U_C(S_r)$ and updated G_ϕ

Curriculum and replay copies use the same starting checkpoint and update operator, and are matched on supervised decisions, target tokens, optimizer updates, and retention data. Episode length, input tokens, and acquisition cost are reported rather than silently treated as matched.

Utility-gated rewriter update. For a chain C_j, define $I_{C,j} = V_{C,j}E_{C,j}$ and

$$g_{C,j} = I_{C,j}\,\mathbf{1}[\widehat{\delta}_{C,j} > 0]\,\mathbf{1}[\widehat{\Delta}_{C,j} > 0]. \tag{4}$$

The intended rewriter update is token-normalized SFT only on chains with $g_{C,j} = 1$ (Algorithm 1). Invalid, unreachable, and non-positive chains remain in the log but receive no gradient. The current study evaluates the components and matched utility comparisons; it does not claim that a multi-round utility-gated rewriter has already produced a benchmark advantage.

5 Experimental Setup

Models and controlled ALFWorld environment. The controlled study uses ALFWorld (Shridhar et al., 2021; Côté et al., 2018), whose PDDL tasks permit exact goal and initial-state edits. Solver and rewriter are Qwen3.5-9B adapters (Qwen Team, 2026; Yang et al., 2025a); we use rank-16 LoRA for solver adaptation as a compute-efficient implementation rather than a methodological requirement, and all compared solver arms use the same adaptation rule. Evaluation is greedy with a 50-step cap. The main goal-rewrite vocabulary contains `retain_state_hold`, `pickup_only`, and `no_edit`; the first two remove the parent placement requirement, and `pickup_only` additionally removes the required state (Appendix Table 5).

Matched ALFWorld batches. All controlled comparisons start from S_0, the replay-trained adapter of an earlier comparison: teacher-imitation initialization followed by 44 updates on 175 decisions from the solver's own verified successes on original training tasks. S_0 solves 105/128 tasks on the frozen development query and 232 of the 274 historically exposed ALFWorld valid-seen/unseen tasks (the frozen base solves 91). B1–B3 train a replay copy R and a rewritten-data copy C from the same start with 39 shared retention decisions. For each arm, B1 uses 54 decisions, 395 target tokens, and 14 updates; B2 uses 52 decisions, 381 target tokens, and 13 updates; and B3 uses 52 decisions, 383 target tokens, and 13 updates. Within a batch, supervised decisions, target tokens, optimizer updates, and retention data are matched. Total input tokens, episode length, acquisition cost, and in B2 family composition are not exactly matched (Appendix B). B1–B2 use the frozen rewriter; B3 uses the validity-SFT rewriter. Each reported ALFWorld arm is one trained adapter, so task-level paired intervals do not capture training-order variability.

Complete WebArena-Lite evaluation. We evaluate the same Qwen3.5-9B solver family on all 165 WebArena-Lite tasks under the recorded project evaluation protocol. The starting solver scores 29/165. Replay-control and rewritten-data updates are available for three completed training runs, denoted Runs A–C; we report run labels rather than raw random-seed integers. Run A retains per-task paired outcomes, while Runs B–C are used as aggregate training-replication checks. Because project development previously inspected part of the Lite suite, we treat this as a protocol-complete benchmark evaluation with an exposure caveat rather than an untouched blind test. The present manuscript uses WebArena as benchmark-scale evidence only: unlike the ALFWorld B1–B3 study, it does not rely on a paper-level audit of every training-budget matching variable for mechanism attribution.

Statistics and evaluation scope. The 128-task ALFWorld query was frozen after the earlier 16-task screens and before outcomes on this query. It spans all six task families and is lineage-disjoint from the rows used to construct the compared updates, but it is a known-family development query rather than a blind benchmark. We report paired task outcomes, exact two-sided McNemar tests, and paired task-bootstrap intervals where the paired counts are available. For the three-run WebArena summary we report every run and the mean difference rather than a significance test over only three training replicates. Historical ALFWorld 274-task results remain explicitly labeled exposed because those tasks were inspected during development.

6 RESULTS

6.1 FULL WEBARENA-LITE: A REPLICATED GAIN FROM REWRITTEN TRAINING DATA

Table 1: **Complete WebArena-Lite165 full-task success.** The shared starting solver scores 29/165. In Run A, the rewritten-data arm has 13 task gains and 3 losses relative to replay (exact two-sided McNemar $p = .021$); the paired task-bootstrap 95% interval for C$-$R is $[+1.8, +10.9]$ pp. Run A is the paired task-level analysis; Runs B–C are used as training-replication checks and are not assigned additional paired p-values.

Run	Replay control	Rewritten-data arm	Δ tasks	Δ SR
A	34/165	**44/165**	+10	+6.1 pp
B	35/165	**43/165**	+8	+4.8 pp
C	33/165	**44/165**	+11	+6.7 pp
Mean	34.0	**43.7**	+9.7	+5.9 pp

The rewritten-data arm exceeds the replay control in all three training runs (Table 1). The mean difference is 9.7 tasks, or 5.9 percentage points, and the per-run difference ranges from +8 to +11 tasks. In the fully paired Run A, the 10-task net gain comes from 13 rewrite-only and 3 replay-only successes. The gain is not confined to one site: relative to replay, the rewritten-data arm gains two tasks on Reddit, four on Shopping, three on Admin, and one on GitLab, with no net change on Map or cross-site tasks. We do not infer a population confidence interval from three training runs, and prior project-level exposure of some Lite tasks remains a limitation. We therefore use WebArena as benchmark-scale evidence that rewritten training data can improve the solver, not as the controlled credit-assignment experiment. The latter is provided by the explicitly audited ALFWorld matched-copy study below; neither result by itself shows that multi-round utility-gated rewriter learning caused the gain (Section 7).

6.2 ALFWORLD MATCHED UTILITY IS HETEROGENEOUS RATHER THAN UNIFORMLY POSITIVE

The three registered batches give qualitatively different answers (Table 2, Figure 2). B1 favors replay: 107 versus 118, with 2 curriculum-only and 13 replay-only successes. B2 is nearly tied. B3 reverses the ordering: the curriculum arm solves 120/128 versus 109/128 for replay, with 13 curriculum-only and 2 replay-only successes (exact $p = .0074$). Multiplying the raw B3 value by four gives a conservative bound of .0296 over the four originally registered curriculum-versus-replay slots; the equally strong negative B1 result is retained rather than hidden. We therefore do not summarize ALFWorld as a universal win. B3 establishes that rewritten data can outperform the matched replay update under the controlled protocol, whereas B1 shows that valid and reachable rewritten data can also underperform replay. The subsequent audit identifies failure-step coverage as one concrete mechanism consistent with this variation.

B3 should also not be read as evidence that utility-gated rewriter learning itself caused the gain. Its rewriter was trained for validity, and the separate six-parent probe shows schema repair without a demonstrated semantic-choice change. The ALFWorld result is therefore evidence about the *training data produced by executable rewriting*; attribution to a multi-round utility-updated G remains open.

Table 2: **Matched ALFWorld comparisons on the frozen 128-task development query.** Paired task-bootstrap 95% intervals for C−R are $[-14.1, -3.1]$, $[-9.4, +6.3]$, and $[+3.1, +14.8]$ pp for B1–B3. B1–B2 use the frozen rewriter; B3 uses the validity-SFT rewriter. Each row contains one trained adapter for each arm, so intervals and exact tests condition on that trained pair.

Batch	Start S_0	Replay R	Curriculum C	C−R	C-only / R-only	Exact p
B1	105/128	**118/128**	107/128	−8.6 pp	2 / 13	.0074
B2	105/128	110/128	108/128	−1.6 pp	11 / 13	.84
B3	105/128	109/128	**120/128**	+8.6 pp	13 / 2	.0074

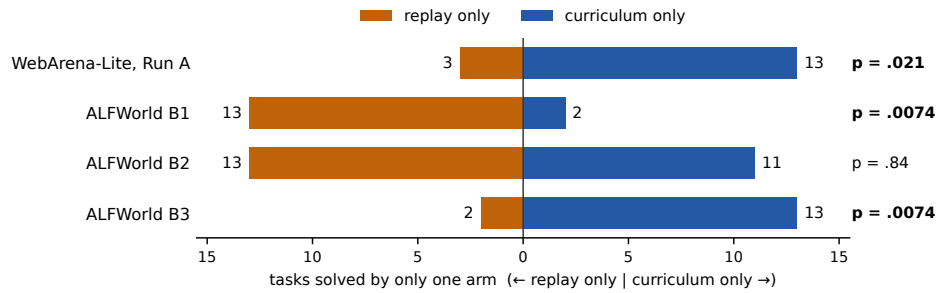


Figure 2: **Paired discordant tasks for the completed task-level comparisons.** Bars count tasks solved by only one arm: curriculum only (blue) or replay only (orange), colored as U_C and U_B in Figure 1; exact two-sided McNemar p-values are shown at right. WebArena Run A and ALFWorld B3 favor rewritten data, B1 favors replay, and B2 is near-balanced. Exact two-sided McNemar p-values use only the discordant paired tasks. This view makes the central empirical point visible: the downstream utility of rewritten training data is strongly heterogeneous across batches.

6.3 Failure relevance explains the B1 failure and motivates a targeted correction

On six frozen failed parents (Figure 3), schema-target training improves native program validity from 2/6 to 6/6, and the starting solver reaches 4/6 resulting children. Yet a fixed-rubric diagnostic of all 14 executed ALFWorld children identifies a sharper issue. Two independent model-based annotators agree on the learning-relevance label for all 14: only 5/14 children exercise the step at which the parent failed, and only 2/10 children actually used for solver training do so. Eight parent failures undo an already completed state change; the original `retain_state_hold` child terminates immediately after that state change and therefore removes the critical placement step. This diagnostic is not used as a benchmark metric or training label; it is used to explain the observed matched-update outcomes.

The same-backend follow-up in Table 3 is consistent with the diagnosis. The original step-removed curriculum succeeds in only 2/12 rollouts, compared with 5/12 for the start and 10/12 for replay. The failure-step-preserving C′ arm succeeds in all 12 rollouts. The correction therefore recovers ten successes relative to the original curriculum and two relative to replay on the same three parents. Because the evaluation is clustered within three training parents, and because the C′ child data were collected by sampling rather than greedily, the difference cannot be attributed to the rewrite operator alone; we treat it as mechanism evidence rather than a generalization result or a stand-alone significance claim. Its value is narrower and more useful: changing *what the child requires* fixes the failure mode exposed by B1, whereas simply making a child valid and reachable did not.

6.4 Additional context does not replace matched utility

Historical fixed rewrites further document why matched utility and replication matter: On 274 historically exposed ALFWorld tasks, a fixed teacher-designed environment rewrite scores 256/274 vs. 232/274 for replay, and an order repeat also favors rewriting (265 vs. 216), but a later, same-start

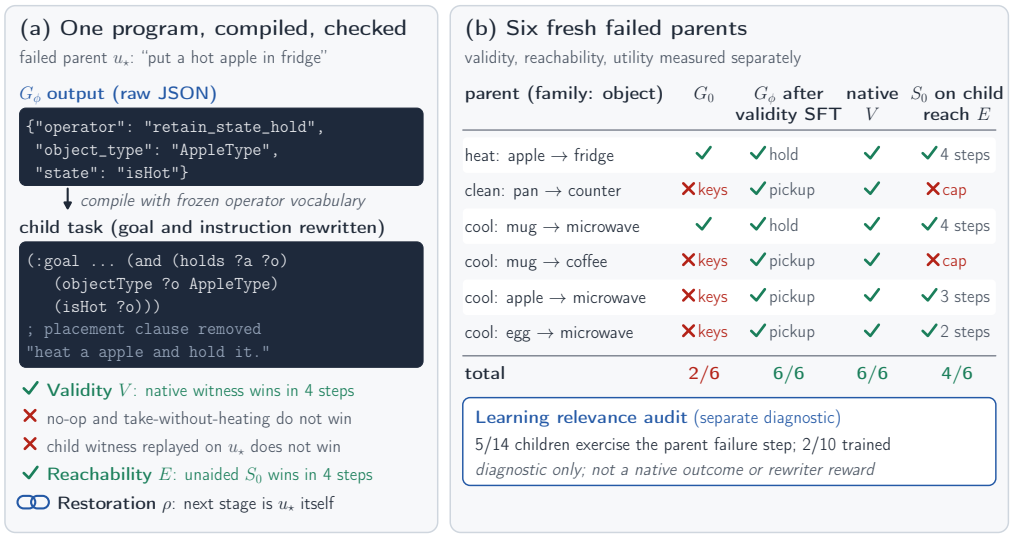


Figure 3: **Executed ALFWorld rewrites.** On six frozen failed parents, G_0 yields 2/6 valid programs because four outputs omit a required state field. After two schema-target updates, all six programs compile and pass native controls; the starting solver reaches 4/6 children in one greedy rollout. The semantic edit choices do not change, so this is evidence for interface compliance, not improved utility selection. The learning-relevance audit is shown separately from validity and reachability because a valid, solvable child need not rehearse the parent's failing step.

Table 3: **Failure-step-preserving follow-up on the three B1 training parents.** Each arm has four sampled rollouts per parent (12 total). C′ keeps the original failing placement step while removing the preceding heating burden; it contains children for Tomato→Fridge-18 and Potato→Fridge-19 only, so its Mug→CoffeeMachine-25 outcomes measure spillover. C′ child rows are sampled S_0 successes, whereas the original C rows are greedy successes, so the collection policy also differs. These 12 outcomes are repeated samples from only three training parents, not 12 independent tasks.

Same-backend arm	Success	vs. S_0	vs. replay
S_0	5/12	—	−5
Replay	**10/12**	+5	—
Original step-removed C	2/12	−3	−8
Failure-step-preserving C′	**12/12**	+7	+2

104-decision refresh reverses the ordering (251 vs. 253). A WebShop transfer check on the same ALFWorld-trained adapters shows score gain over replay but no apparent full-success gain. These measurements are retained in Appendix D and Table 6: They are context, not evidence for the learned rewriter.

7 Limitations and Conclusion

The evidence now consists of one complete web benchmark gain, one positive controlled ALFWorld batch, one negative controlled batch, and one targeted mechanism correction. This variety is why we are setting a conservative boundary for our claims. WebArena-Lite is protocol-complete, not a fresh, blind benchmark, since part of the suite had prior exposure at the project level. Only Run A has per-task paired outcomes in the current evidence package; Runs B-C replicate the aggregate direction and are not treated as second and third paired tests. We reserve detailed matching and causal mechanism claims for ALFWorld, where training auditing is explicit. The 128-task ALFWorld query is a known-family development query, and each B1-B3 row has one trained adapter per arm. Since batches differ in the particular rewritten trajectories they include, B1 versus B3 is diagnostic, not a randomized comparison of rewriter update rules.

The failure-step follow-up is also deliberately narrow: 12 rollouts cluster on only three B1 training parents. It shows that preserving the failing step repairs the bad original-curriculum behavior on those parents but does not establish held-out transfer. The learning-relevance audit is model-based and diagnostic and is not used to score benchmark success. Finally, current experiments validate the utility criterion through matched solver copies but have not yet isolated a long-horizon benchmark gain attributable to multi-round, utility-gated updates of the rewriter.

Within these limits, the empirical picture is more than a simple positive leaderboard claim. WebArena-Lite finds a consistent 8-11 task margin over replay in all three training runs. ALFWorld shows both a substantial loss and a substantial gain under matched updates: The mechanism question is observable, not masked by an average. The follow-up directly alters the rewrite to preserve the lost capability and regains performance on the affected parents. biCurriculum's main contribution is thus a quantifiable criterion for executable curricula: A child is only helpful when training on it improves the unchanged task compared to a strong replay intervention. The results show that the criterion does reveal an important difference missed by validity and child success alone.

Reproducibility Statement

The key comparisons are recalculated from stored per-task native outcomes with fixed task identities. Appendix A documents the validity controls; Appendix B documents the matching audit; Appendix C documents the failure-step diagnostic; Appendix D documents additional benchmark context. Training and evaluation records contain model and adapter identities, actions executed, outcomes, assistance flags, and budgets. Teacher-assisted trajectories are separately labeled from unaided solver rollouts; unexecuted trajectories are not counted as successes. The three WebArena training replications are named Runs A-C; Run A per-task outcomes and the matching evaluation records are the paired-analysis objects; Runs B-C are reported as aggregate replication checks. We plan to release the edit compiler, frozen task lists, training configurations, outcome records, and analysis scripts in an anonymized artifact.

AI-Use Statement

Generative AI tools have been sparingly used for exploratory ideation, drafting and editing, literature search, and implementation and data processing tasks. All decisions about research design and conception, verification of experimental records and analyses, interpretation of results, and preparation of the manuscript and accompanying materials are the authors' sole responsibility.

Ethics and Responsible Execution

Environment changes occur only on isolated authorized research instances; public production sites and service instances for other projects are not modified. No hidden evaluator data, credentials, or any private operational details enter the model inputs or the planned evidence release. Auxiliary judgments are kept separate from the native task outcomes, and incomplete attempts are preserved and not silently recoded as either failures or successes.

References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48. ACM, 2009. doi: 10.1145/1553374.1553380.

Yuan Cao, Dezhi Ran, Mengzhou Wu, et al. GUI-GENESIS: Automated synthesis of efficient environments with verifiable rewards for GUI agent post-training. *arXiv preprint arXiv:2602.14093*, 2026. URL https://arxiv.org/abs/2602.14093.

Hyungjoo Chae, Jungsoo Park, and Alan Ritter. Safe and scalable web agent learning via recreated websites. *arXiv preprint arXiv:2603.10505*, 2026. URL https://arxiv.org/abs/2603.10505.

Hojun Chung, Junseo Lee, Minsoo Kim, Dohyeong Kim, and Songhwai Oh. Adversarial environment design via regret-guided diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 63715–63746, 2024. doi: 10.52202/079017-2035. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/74953ef4abd9c436344e59d687ad34d3-Abstract-Conference.html.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, et al. TextWorld: A learning environment for text-based games. *arXiv preprint arXiv:1806.11532*, 2018. URL https://arxiv.org/abs/1806.11532. Computer Games Workshop at IJCAI 2018.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In *Advances in Neural Information Processing Systems*, volume 33, pp. 13049–13061, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/985e9a46e10005356bbaf194249f6856-Abstract.html.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward rAnked FineTuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2304.06767.

Maxence Faldor, Jenny Zhang, Antoine Cully, and Jeff Clune. OMNI-EPIC: Open-endedness via models of human notions of interestingness with environments programmed in code. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2405.15568.

Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Learning to teach. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1805.03643.

Zhiyuan Fan, Tinghao Yu, Yuanjun Cai, et al. Environment evolution for terminal agents. *arXiv preprint arXiv:2609.04128*, 2026. URL https://arxiv.org/abs/2609.04128.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on Robot Learning*, 2017. URL https://arxiv.org/abs/1707.05300.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1515–1528, 2018. URL https://proceedings.mlr.press/v80/florensa18a.html.

Thomas Foster, Anya Sims, Johannes Forkel, and Jakob Foerster. LILO: Learning to reason at the frontier of learnability. In *Advances in Neural Information Processing Systems*, volume 38, pp. 48941–48974, 2025. doi: 10.52202/085713-1634. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/45d5823950cab9a88e10c6e8dbff351c-Abstract-Conference.html.

Amirata Ghorbani and James Zou. Data Shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251, 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1311–1320, 2017. URL https://proceedings.mlr.press/v70/graves17a.html.

Jiacheng Guo, Ling Yang, Peter Chen, Qixin Xiao, Yinjie Wang, Xinzhe Juan, Jiahao Qiu, Ke Shen, and Mengdi Wang. GenEnv: Difficulty-aligned co-evolution between LLM agents and environment simulators. *arXiv preprint arXiv:2512.19682*, 2025. URL https://arxiv.org/abs/2512.19682v2. Version 2.

Yihao Hu, Zhihao Wen, Xiujin Liu, Pan Wang, Xin Zhang, and Wei Wu. SEAL: Synergistic co-evolution of agents and learning environments. *arXiv preprint arXiv:2605.24426*, 2026. URL https://arxiv.org/abs/2605.24426.

Chengsong Huang, Zifeng Wang, Rujun Han, Jun Yan, Yanfei Chen, Zoey CuiZhu, Ke Jiang, Peng Xia, Han Yu, Yufan Zhuang, Yifei Ming, Jiaqi Pan, Bhavana Dalvi Mishra, Jiaxin Huang, Burak Gokturk, Tomas Pfister, and Chen-Yu Lee. EnvHarness: Awakening static worlds for agent learning. *arXiv preprint arXiv:2608.19880*, 2026a. URL https://arxiv.org/abs/2608.19880v1. Version 1.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-Zero: Self-evolving reasoning LLM from zero data. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d49b9aacebda61051166335af6fd3061-Abstract-Conference.html.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4940–4950, 2021. URL https://proceedings.mlr.press/v139/jiang21b.html.

Zaid Khan, Elias Stengel-Eskin, Jaemin Cho, and Mohit Bansal. DataEnvGym: Data generation agents in teacher environments with student feedback. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.06215.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Xiaochuan Li, Zichun Yu, and Chenyan Xiong. Montessori-Instruct: Generate influential training data tailored for student learning. In *International Conference on Learning Representations*, pp. 74809–74834, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ba1d33849b963efc6b5d3082ad68f480-Abstract-Conference.html.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, et al. VisualAgentBench: Towards large multimodal models as visual foundation agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/eea71dc576381b88f2a0ca4dedc2140d-Abstract-Conference.html.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, 2020. doi: 10.1109/TNNLS.2019.2934906.

Shikhar Murty, Hao Zhu, Dzmitry Bahdanau, and Christopher D. Manning. NNetNav: Unsupervised learning of browser agents through environment interaction in the wild. *arXiv preprint arXiv:2410.02907*, 2024. URL https://arxiv.org/abs/2410.02907.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3878–3887, 2018. URL https://proceedings.mlr.press/v80/oh18b.html.

Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17473–17498, 2022. URL https://proceedings.mlr.press/v162/parker-holder22a.html.

Zehan Qi, Xiao Liu, Iat Long Iong, et al. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c66e1fcc9691aae706250638f36f681b-Abstract-Conference.html.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Cinjon Resnick, Roberta Raileanu, Sanyam Kapoor, Alexander Peysakhovich, Kyunghyun Cho, and Joan Bruna. Backplay: "Man muss immer umkehren". *arXiv preprint arXiv:1807.06919*, 2018. URL https://arxiv.org/abs/1807.06919.

Tim Salimans and Richard Chen. Learning Montezuma's Revenge from a single demonstration. *arXiv preprint arXiv:1812.03381*, 2018. URL https://arxiv.org/abs/1812.03381.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2312.06585.

Hongjin Su, Ruoxi Sun, Jinsung Yoon, Pengcheng Yin, Tao Yu, and Sercan Ö. Arık. Learn-by-interact: A data-centric framework for self-adaptive agents in realistic environments. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2501.10893.

Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1703.05407.

Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, et al. OS-Genesis: Automating GUI agent trajectory construction via reverse task synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5555–5579, 2025. doi: 10.18653/v1/2025.acl-long.277. URL https://aclanthology.org/2025.acl-long.277/.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired Open-Ended Trailblazer (POET): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019. URL https://arxiv.org/abs/1901.01753.

Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. OpenClaw-RL: Train any agent simply by talking. *arXiv preprint arXiv:2603.10165*, 2026a. URL https://arxiv.org/abs/2603.10165v2. Version 2.

Yinjie Wang, Tianbao Xie, Ke Shen, Mengdi Wang, and Ling Yang. RLAnything: Forge environment, policy, and reward model in completely dynamic RL system. In *International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=JLQV6NxS6W. arXiv:2602.02488.

Ziyi Wang, Yuxuan Lu, Yimeng Zhang, Qun Liu, Chen Luo, Jiri Gesi, Hanqing Lu, Yisi Sang, Manling Li, Jing Huang, and Dakuo Wang. SENTINEL: Failure-driven reinforcement learning for training tool-using language model agents. *arXiv preprint arXiv:2606.12908*, 2026c. URL https://arxiv.org/abs/2606.12908.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. WebAgent-R1: Training web agents via end-to-end multi-turn reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7909–7928, 2025. doi: 10.18653/v1/2025.emnlp-main.401. URL https://aclanthology.org/2025.emnlp-main.401/.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132, 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

Jingxu Xie, Dylan Xu, Xuandong Zhao, and Dawn Song. AgentSynth: Scalable task generation for generalist computer-use agents. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fb0f9005938c5a80b1114a42156e99e2-Abstract-Conference.html.

Tianci Xue, Zeyi Liao, Tianneng Shi, Zilu Wang, Kai Zhang, Dawn Song, Yu Su, and Huan Sun. Autonomous continual learning for environment adaptation of computer-use agents. *arXiv preprint arXiv:2602.10356*, 2026. URL https://arxiv.org/abs/2602.10356. Version 2, 11 May 2026.

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

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. AgentOccam: A simple yet strong baseline for LLM-based web agents. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f2c6e459b95694a24ac69c469a4ee746-Abstract-Conference.html.

Teresa Yeo, Dulaj Weerakoon, Dulanga Weerakoon, and Archan Misra. Towards adaptive environment generation for training embodied agents. *arXiv preprint arXiv:2602.06366*, 2026. URL https://arxiv.org/abs/2602.06366. AAAI-26 Bridge Program B10.

Jinsung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10842–10851, 2020. URL https://proceedings.mlr.press/v119/yoon20a.html.

Abhay Zala, Jaemin Cho, Han Lin, Jaehong Yoon, and Mohit Bansal. EnvGen: Generating and adapting environments via LLMs for training embodied agents. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2403.12014.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488, 2022. doi: 10.52202/068431-1126. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/639a9a172c044fbb64175b5fad42e9a5-Abstract-Conference.html.

Chenghao Zhang, Canran Xiao, SaiSai Hu, and Dan Roth. Training needs trustworthy worlds: Verified synthetic web environments for agent learning. *arXiv preprint arXiv:2608.21898*, 2026a. URL https://arxiv.org/abs/2608.21898.

Zijing Zhang and Xiajie Yang. CRPS: Curriculum replay via progressive suffixes from successful trajectories for long-horizon LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13891–13904. Association for Computational Linguistics, July 2026. doi: 10.18653/v1/2026.findings-acl.680. URL https://aclanthology.org/2026.findings-acl.680/.

Ziyun Zhang, Zezhou Wang, Xiaoyi Zhang, Zongyu Guo, Jiahao Li, Bin Li, and Yan Lu. InfiniteWeb: Scalable web environment synthesis for GUI agent training. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28465–28492, 2026b. doi: 10.18653/v1/2026.acl-long.1313. URL https://aclanthology.org/2026.acl-long.1313/.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute Zero: Reinforced self-play reasoning with zero data. In *Advances in Neural Information Processing Systems*, volume 38, pp. 105816–105879, 2025. doi: 10.52202/085713-3534. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/9837dc00ff67d176373268ed48042d49-Abstract-Conference.html.

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. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4410c0711e9154a7a2d26f9b3816d1ef-Abstract-Conference.html.

Yifei Zhou, Sergey Levine, Jason Weston, Xian Li, and Sainbayar Sukhbaatar. Self-challenging language model agents. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113959–113991, 2025a. doi: 10.52202/085713-3801. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a5a305fac88fb4ae40969cfec5eef48d-Abstract-Conference.html.

Yifei Zhou, Qianlan Yang, Kaixiang Lin, Min Bai, Xiong Zhou, Yu-Xiong Wang, Sergey Levine, and Li Erran Li. Proposer-Agent-Evaluator (PAE): Autonomous skill discovery for foundation model internet agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79490–79528, 2025b. URL https://proceedings.mlr.press/v267/zhou25ah.html.

A Protocol Details

Native validity. For an edit C_j, the compiler checks that the declared operator and object/state bind to the parent task. A positive witness must satisfy the child verifier. Registered negative controls include no-op, skipped-state or wrong-object trajectories where applicable, and replay of the child witness on the unchanged parent. Restoration must return the goal, environment state, verifier revision, initial snapshot, and action mapping to the parent's identities. These checks establish syntax and environment consistency, not learning relevance.

Reachability and assistance. Reachability requires unaided solver success on the child and at least one executed solver action. Teacher proposals may initialize the rewriter or provide separate solver demonstrations, but teacher actions are tagged and excluded from unaided child-success and evaluation denominators. The native verifier, rather than an auxiliary language-model judge, determines task success.

Utility credit. Curriculum and replay copies start from the same solver checkpoint and use the same update operator. Missing data are not filled by unlimited repetition. For per-parent credit, Eq. 2 is evaluated on unchanged parents with paired reset-start rollouts; a chain is eligible for rewriter SFT only when it is valid, reachable, improves over the start, and improves over replay. This admission rule is a noisy training signal, not a statistical significance test. Figure 4 summarizes which signals are measured and which supervise each model.

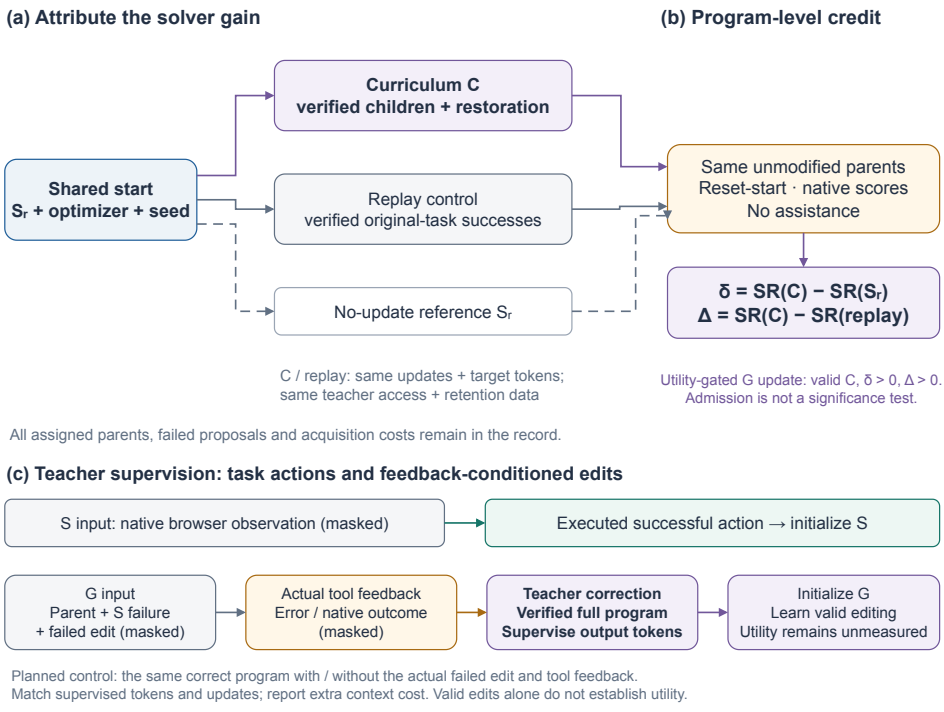


Figure 4: **Credit and supervision interfaces.** Curriculum, replay, and no-update copies are evaluated on the same unmodified parents; teacher action supervision and rewriter-program supervision remain separate. The diagram clarifies which signals are measured and which are never used as task success.

Replay sampling contract. Verified original-task demonstrations used for replay are frozen before candidate generation, and both branches share the same separate retention-anchor pool. Replay is sampled under the same finite presentation, supervised-target-token, and optimizer-update caps as the curriculum branch. Missing parents are reported rather than filled with unexecuted targets or unlimited repeats; current child trajectories never enter the original-parent replay pool. This

comparison therefore measures the value of the rewritten-data intervention relative to the specified replay distribution, not ordering alone. Collection and input-context costs are reported separately when not matched.

B Matching Audit and Controlled Evidence

Table 4: Matching audit for B1–B3 from the saved training files (one trained adapter for each arm). Supervised decisions, target tokens, updates, and shared retention data are matched within each batch; input tokens, episode structure, family composition and acquisition cost are not. A dash means the variable is not itemized for B3 in this audit, not that it is matched.

Variable	B1 R / C	B2 R / C	B3 R / C
Supervised decisions	54 / 54	52 / 52	52 / 52
Target tokens	395 / 395	381 / 381	383 / 383
Optimizer updates	14 / 14	13 / 13	13 / 13
Shared retention decisions	39 / 39	39 / 39	39 / 39
Input tokens	128,444 / 116,704	116,845 / 109,934	113,761 / 107,329
Non-shared episodes	2 original / 3 child	2 original / 3 child	—
Decisions per non-shared episode	7.5 / 5.0	6.5 / 4.3	—
Prompt characters in non-shared data	88,416 / 59,919	59,809 / 41,588	—
Family of non-shared data	heat / heat	heat / 1 heat, 2 cool	—
Acquisition cost	replay pool / rewriter, compiler and probes (not equalized)		

In B1, the non-shared training data in both R and C come from the heat family: R contains two original-task episodes, whereas C contains three shorter child episodes. In B2, R's non-shared training data come from the heat family, whereas two of C's three child episodes come from the cool family. B3 retains the same matched update counts shown above; its positive 120/128 result should therefore be interpreted together with, not substituted for, the B1/B2 results. Acquisition cost and episode-level input context are not equalized by the target-token match.

Rewriting-free controls. Two further arms start from S_0 and train on fresh S_0 rollouts of original training tasks without a matched child arm: self-imitation SFT on all admissible successes (10 updates) and one-pass fixed-rollout policy gradient on all four sampled episodes per task group (3 updates). Each leaves S_0's total unchanged at 105/128 (10/10 and 3/3 discordant; exact $p = 1$), a null result rather than evidence of equivalence; their update counts are not matched to R or C.

Table 5: Goal-predicate differences of the executed ALFWorld operators. The compiler accepts an edit only if the named object type T and state s bind exactly one goal variable of the parent; instructions are generated from fixed templates.

Rewrite	Child goal	Removed from parent goal
Original parent (heat / clean / cool)	$\exists o, r : \mathrm{objectType}(o, T) \wedge s(o) \wedge \mathrm{receptacleType}(r, R) \wedge \mathrm{inReceptacle}(o, r)$	None
`retain_state_hold`	$\exists a, o : \mathrm{holds}(a, o) \wedge \mathrm{objectType}(o, T) \wedge s(o)$	Placement predicates: `receptacleType`, `inReceptacle`
`pickup_only`	$\exists a, o : \mathrm{holds}(a, o) \wedge \mathrm{objectType}(o, T)$	Placement predicates and state s: `isHot`, `isClean`, `isCool`
`no_edit`	Parent goal	None (no child)

Statistical reporting. For paired task-level comparisons we use exact two-sided McNemar tests and task bootstrap intervals over the fixed evaluation list. The B1–B4 curriculum-versus-replay family was pre-registered as four slots; B1 and B3 each have raw $p = .0074$, so a conservative four-way

Bonferroni bound is below .030 for either contrast. We do not pool B1–B3 because their rewritten data differ, and the intervals condition on one trained adapter for each arm. For the three WebArena training runs we report each run separately and summarize the mean task difference; three runs are too few to justify a precise population-level variance estimate. The E3 follow-up has only three parent clusters, so its 12 rollout slots are descriptive repeated samples, not 12 independent tasks.

Per-parent B1 utility before the corrected rewrite. The earlier sampled diagnostic on the three B1 training parents used a different backend and stopped before a fully balanced comparison. On the first rollout round complete for every arm, replay solves each parent while the original curriculum solves none. We retain this historical diagnostic only as motivation. The main failure-step comparison now uses the same-backend aggregate counts in Table 3.

Rewriter construction pool. On the six-parent frozen pool, G_0 has 2/6 schema-valid proposals. Two validity-SFT updates yield 6/6 valid programs, but the operator/object choices remain unchanged. All six children pass native construction controls and the starting solver solves four in one greedy rollout. The later utility-surrogate rewriter makes the same six semantic choices and therefore provides no evidence that utility feedback itself changed selection on this probe.

C Learning-Relevance and Failure-Step Follow-up

Two independent model-based annotators applied a fixed rubric to all 14 executed ALFWorld children and agreed on every learning-relevance label. All 14 are labeled semantically valid, but only 5/14 are learning-relevant; among the 10 children used for training, 2/10 are relevant. Eight parent failures undo a completed state change, three never take the correct object type, two skip the state change, and one makes a wrong placement action. For 7/14 parents, the failing transition lies outside the short failure-history excerpt shown to the rewriter. These labels are used only as a diagnostic of the training data, never as the native task outcome or a rewriter reward.

For Tomato→Fridge-18 and Potato→Fridge-19, state-given children keep the original parent goal and verifier but initialize goal-type objects outside the fridge in the required hot state, removing the heating prefix while preserving the failing placement step. Before training, the starting solver reaches these children in 2/4 sampled rollouts, versus 4/4 for the corresponding step-removed children; every failed state-given rollout repeats the parent cooling error.

The failure-step-preserving C′ training arm uses 56 decisions (17 child + 39 retention) from the same S_0 start and 14 LoRA updates. Its teacher-forced development exact match changes from 33/39 to 34/39 while NLL rises from 0.162 to 0.304, so the recorded-state diagnostics are mixed rather than uniformly favorable. The native same-backend comparison on the three B1 training parents is S_0 5/12, replay 10/12, original step-removed C 2/12, and C′ 12/12. Each denominator is four sampled rollouts for each of the same three parents. Because the unit set contains only three parent clusters, the manuscript reports the counts descriptively and does not turn the 12 rollout slots into an independent-task significance test.

D Additional Robustness Details

WebArena-Lite site distribution. For the fully paired Run A, the rewritten-data-minus-replay net task differences are Reddit $+2$, Shopping $+4$, Admin $+3$, GitLab $+1$, Map 0, and cross-site 0, summing to the overall +10. The gain is therefore distributed across four single-site categories rather than generated by one site, while no net gain is observed on Map or cross-site tasks.

Historical ALFWorld rewrite sensitivity. A separate fixed, teacher-designed environment rewrite provides a useful stress test of training-order sensitivity. On 274 historically exposed tasks, the original comparison is 256/274 for rewriting versus 232/274 for replay, and an order repeat gives 265/274 versus 216/274. A later same-start 104-decision refresh from the 232/274 checkpoint reverses the ordering: 251/274 rewriting versus 253/274 replay. These are not independent seeds under one frozen protocol and are not pooled; they motivate reporting the B1–B3 controlled batches separately rather than selecting one favorable historical order.

Table 6: Completed context results. The fixed rewrite is teacher-designed, not learned G; same-domain WebShop cSFT is a training-pipeline diagnostic. None of these rows is used as evidence that utility-gated rewriter learning caused the main benchmark result.

Context experiment	Base/start	Comparator	Rewritten / trained arm
WebShop transfer of ALFWorld adapters (score / SR)	42.50 / 20.8	38.92 / 19.4 replay	42.07 / 21.2 fixed rewrite
Historical ALFWorld274 (success)	91/274 base	232/274 replay	256/274 fixed rewrite
Same-start ALFWorld refresh (success)	232/274 start	253/274 replay	251/274 fixed rewrite
WebShop same-domain teacher cSFT (score / SR)	42.50 / 20.8	—	71.68 / 44.4

E Reproducibility Identities and Reporting Boundaries

The solver and rewriter use the same pinned Qwen3.5-9B model family in the controlled ALFWorld study, with checkpoint identities retained in the project records. We use rank-16 LoRA for solver adaptation as a compute-efficient implementation rather than a methodological requirement; the controlled B1–B3 comparisons use the same adaptation rule and the per-batch decision, target-token, and update counts in Table 4. Native ALFWorld evaluation uses the declared 50-step cap and greedy decoding. The controlled B1–B3 results, the narrow failure-step follow-up, the historical fixed-rewrite diagnostics, and the WebArena-Lite runs are kept as separate experimental families rather than merged into a single causal table.

For WebArena, Runs A–C are anonymous labels for three completed training runs. The anonymous artifact should retain the frozen task order, checkpoint identities, evaluator configuration, and available per-task outcomes; raw random-seed integers are not needed in the paper. Run A has per-task discordance counts (13 rewrite-only, 3 replay-only); the manuscript does not infer discordant counts or p-values for Runs B–C from aggregate totals alone. Because a complete paper-level training-budget audit is not reconstructed for these runs here, WebArena is not used for the matched-copy mechanism claim.

Reported task-bootstrap intervals condition on the trained policies and do not include training-order uncertainty. A dash in any context table means no complete comparable result, not zero. Repeated rollouts on the same parent are not relabeled as independent tasks, partial evaluations retain their actual denominator, and execution-process success is never substituted for task success.